Cisco IOS Software Configuration Guide

Release 12.2(33)SXH and Later Releases

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-13013-06


CONTENTS

Preface

This preface describes who should read the *Cisco IOS Software Configuration Guide,* Release 12.2SX , and its document conventions.

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining the switches supported in Cisco IOS Release 12.2SX.

Related Documentation

The following publications are available for Cisco IOS Release 12.2SX:

- *Catalyst 6500 Series Switch Installation Guide*
- *Catalyst 6500 Series Switch Module Installation Guide*
- Cisco IOS Master Command List
- *Catalyst 6500 Series Switch Cisco IOS System Message Guide*, Release 12.2SX
- *Release Notes for Cisco IOS Release 12.2SX*
- Cisco IOS Configuration Guides and Command References:

  http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html
- For information about MIBs, go to this URL:

  http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Conventions

This document uses the following conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in **boldface**.
italic font	Arguments for which you supply values are in *italics*.

Convention	Description
[ ]	Elements in square brackets are optional.
{ x | y | z }	Alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`screen` font	Terminal sessions and information the system displays are in `screen` font.
`boldface screen` font	Information you must enter is in **`boldface screen`** font.
`italic screen` font	Arguments for which you supply values are in *`italic screen`* font.
→	`This pointer highlights an important line of text in an example.`
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords are in angle brackets.

Notes use the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Cautions use the following conventions:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly *What's New* in Cisco Product Documentation, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Product Overview

This chapter consists of these sections:

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Supervisor Engine Memory Devices and Ports

For complete information about the chassis, modules, and software features supported by Cisco IOS Release 12.2SX, see the *Release Notes for Cisco IOS Release 12.2SX*:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html

These sections describe the ports and flash memory devices on the supervisor engines:

Understanding Supervisor Engine 720-10GE Memory Devices and Ports

These sections describe the Supervisor Engine 720-10GE memory devices and ports:

Supervisor Engine 720-10GE Flash Memory Devices

The Supervisor Engine 720-10GE has these flash memory devices:

- **disk0:** (active) and **slavedisk0:** (standby):
  - External CompactFlash Type II slots
  - For CompactFlash Type II flash PC cards sold by Cisco Systems, Inc.
- **sup-bootdisk:** (active) and **slavesup-bootdisk:** (standby):
  - Switch processor (SP) 1-GB internal CompactFlash flash memory
  - From SP ROMMON, it is **bootdisk:**
  - Not accessible from route processor (RP) ROMMON
- **bootflash:** (active) and **slave-bootflash:** (standby):
  - RP 64-MB internal flash memory
  - Not accessible from SP ROMMON

Supervisor Engine 720-10GE Ports

The Supervisor Engine 720-10GE has these ports:

- Console port—EIA/TIA-232 (RS-232) port
- Ports 1 and 2
  - Gigabit Ethernet SFP (fiber or 10/100/1000 Mbps RJ-45)
  - Fast Ethernet SFP
- Port 3—10/100/1000 Mbps RJ-45
- Ports 4 and 5—10-Gigabit Ethernet X2

Note The 1-Gigabit Ethernet ports and the 10-Gigabit Ethernet ports have the same QoS port architecture (2q4t/1p3q4t) unless you disable the 1-Gigabit Ethernet ports with the **mls qos 10g-only** global configuration command. With the 1-Gigabit Ethernet ports disabled, the QoS port architecture of the 10-Gigabit Ethernet ports is 8q4t/1p7q4t.

See the "Configuring Optional Interface Features" section on page 8-6 for information about configuring the ports.

Understanding Supervisor Engine 720 Memory Devices and Ports

These sections describe the Supervisor Engine 720 memory devices and ports:

Supervisor Engine 720 Flash Memory Devices

The Supervisor Engine 720 has these flash memory devices:

- **disk0:** and **disk1:** (active) and **slavedisk0:** and **slavedisk1:** (standby):
  - External CompactFlash Type II slots
  - For CompactFlash Type II flash PC cards sold by Cisco Systems, Inc.
- **sup-bootflash:** (active) and **slavesup-bootflash:** (standby):
  - Switch processor (SP) 64-MB internal flash memory
  - From SP ROMMON, it is **bootflash:**
  - Not accessible from route processor (RP) ROMMON
- With WS-CF-UPG=, **sup-bootdisk:** (active) and **slavesup-bootflash:** (standby):
  - SP 512-MB internal CompactFlash flash memory
  - From SP ROMMON, it is **bootdisk:**
  - Not accessible from RP ROMMON
  - See this publication for more information:

    http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/hardware/Config_Notes/78_17277.html
- **bootflash:** (active) and **slave-bootflash:** (standby):
  - RP 64-MB internal flash memory
  - Not accessible from SP ROMMON

Configuring Supervisor Engine 720 Ports

The Supervisor Engine 720 has these ports:

- Port 1—Small form-factor pluggable (SFP); no unique configuration options.
- Port 2— RJ-45 connector and an SFP connector (default). To use the RJ-45 connector, you must change the configuration.

  To configure port 2 on a Supervisor Engine 720 to use either the RJ-45 connector or the SFP connector, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface gigabitethernet`** *`slot`*`/2`	Selects the Ethernet port to be configured.
Step 2	`Router(config-if)#` **`media-type`** `{`**`rj45`** `|` **`sfp`**`}`	Selects the connector to use.

This example shows how to configure port 2 on a Supervisor Engine 720 in slot 5 to use the RJ-45 connector:

```
Router(config)# interface gigabitethernet 5/2
Router(config-if)# media-type rj45
```

See the “Configuring Optional Interface Features” section on page 8-6 for more information about configuring the ports.

Understanding Supervisor Engine 32 Memory Devices and Ports

These sections describe the Supervisor Engine 32 memory devices and ports:

- Supervisor Engine 32 Flash Memory Devices, page 1-4
- Supervisor Engine 32 Ports, page 1-4

Note Supervisor Engine 32 does not support switch fabric connectivity.

Supervisor Engine 32 Flash Memory Devices

The Supervisor Engine 32 has these flash memory devices:

- **disk0:** (active) and **slavedisk0:** (standby):
  - External CompactFlash Type II slots
  - For CompactFlash Type II flash PC cards sold by Cisco Systems, Inc.
- **sup-bootdisk:** (active) and **slavesup-bootflash:** (standby):
  - Switch processor (SP) 256-MB internal CompactFlash flash memory
  - From SP ROMMON, it is **bootdisk:**
  - Not accessible from route processor (RP) ROMMON
- **bootflash:** (active) and **slave-bootflash:** (standby):
  - RP 64-MB internal flash memory
  - Not accessible from SP ROMMON

Supervisor Engine 32 Ports

The Supervisor Engine 32 has these ports:

- Console port—EIA/TIA-232 (RS-232) port
- Two Universal Serial Bus (USB) 2.0 ports—Not currently enabled
- WS-SUP32-GE-3B:
  - Ports 1 through 8—Small form-factor pluggable (SFP)
  - Port 9—10/100/1000 Mbps RJ-45
- WS-SUP32-10GE:
  - Ports 1 and 2—10-Gigabit Ethernet XENPAK
  - Port 3—10/100/1000 Mbps RJ-45

See the "Configuring Optional Interface Features" section on page 8-6 for information about configuring the ports.

Understanding ME6500 Flash Memory Devices and Ports

These sections describe the Cisco ME6500 series Ethernet switch memory devices and ports:

- ME6500 Flash Memory Devices, page 1-5
- ME6500 Ports, page 1-5

ME6500 Flash Memory Devices

The ME6500 has these flash memory devices:

- **disk0:**
  - One external CompactFlash Type II slot
  - Supports CompactFlash Type II flash PC cards
- **sup-bootflash:**
  - Switch processor (SP) 128 MB internal CompactFlash flash memory
  - From SP ROMMON, it is **bootflash:**
  - Not accessible from route processor (RP) ROMMON
- **bootflash:**
  - RP 64-MB internal flash memory
  - Not accessible from SP ROMMON

ME6500 Ports

The ME6500 has these ports:

- ME-C6524GS-8S and ME-C6524GT-8S
  - Ports 25–32: Gigabit Ethernet SFP
  - Requires Gigabit Ethernet SFPs
- ME-C6524GS-8S
  - Ports 1–24: Gigabit Ethernet SFP
  - Requires Gigabit Ethernet SFPs
- ME-C6524GT-8S—Ports 1–24: 10/100/1000 Mbps RJ-45 Ethernet ports

User Interfaces

Release 12.2SX supports configuration using the following interfaces:

- CLI—See Chapter 2, "Command-Line Interfaces."
- SNMP—See the Release 12.2 IOS *Configuration Fundamentals Configuration Guide* and *Command Reference* at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/ffun_c.html

- Cisco IOS web browser interface—See "Using the Cisco Web Browser" in the IOS *Configuration Fundamentals Configuration Guide* at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf005.html

Module Status Monitoring

The supervisor engine polls the installed modules with Switch Communication Protocol (SCP) messages to monitor module status.

The SCP sends a message every two seconds to each module. Module nonresponse after 3 messages (6 seconds) is classified as a failure. CPU_MONITOR system messages are sent every 30 seconds. After 25 sequential failures (150 seconds), the supervisor engine power cycles the module and sends a CPU_MONITOR TIMED_OUT system message and OIR PWRCYCLE system messages.

Software Features Supported in Hardware by the PFC and DFC

The PFC3 and DFC3 provide hardware support for these Cisco IOS software features:

- Access Control Lists (ACLs) for Layer 3 ports and VLAN interfaces:
  - Permit and deny actions of input and output standard and extended ACLs

    Note Flows that require ACL logging are processed in software on the route processor (RP).

  - Except on MPLS interfaces, reflexive ACL flows after the first packet in a session is processed in software on the RP
  - Dynamic ACL flows

    Note Idle timeout is processed in software on the RP.

  For more information about PFC and DFC support for ACLs, see Chapter 49, "Understanding Cisco IOS ACL Support."

  For complete information about configuring ACLs, see the Cisco IOS Security Configuration Guide, Release 12.2, "Traffic Filtering and Firewalls," at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfacls.html

- Bidirectional Protocol Independent Multicast (PIM) in hardware—See "Understanding IPv4 Bidirectional PIM" section on page 37-9.
- IPv4 Multicast over point-to-point generic route encapsulation (GRE) Tunnels—See the publication at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/interface/configuration/guide/icflogin.html

- Multiple-path Unicast Reverse Path Forwarding (RPF) Check—To configure Unicast RPF Check, see the "Configuring Unicast Reverse Path Forwarding Check" section on page 47-2.
- Except on MPLS interfaces, Network Address Translation (NAT) for IPv4 unicast and multicast traffic.

  Note the following information about hardware-assisted NAT:

   - NAT of UDP traffic is not supported in PFC3A mode.
   - The PFC3 does not support NAT of multicast traffic.
   - The PFC3 does not support NAT configured with a route-map that specifies length.
   - When you configure NAT and NDE on an interface, the PFC3 sends all traffic in fragmented packets to the RP to be processed in software. (CSCdz51590)

  To configure NAT, see the *Cisco IOS IP Configuration Guide*, Release 12.2, "IP Addressing and Services," "Configuring IP Addressing," "Configuring Network Address Translation," at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfipadr.html

  To prevent a significant volume of NAT traffic from being sent to the RP, due to either a DoS attack or a misconfiguration, enter the **mls rate-limit unicast acl** {**ingress** | **egress**} command described at this URL:

  http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_m2.html#mls_rate-limit_unicast_acl

  (CSCea23296)

- NetFlow Aggregation—See this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/netflow.html#NetFlow_Aggregation

- Policy-based routing (PBR) for route-map sequences that use the **match ip address**, **set ip next-hop**, and **ip default next-hop** PBR keywords.

  To configure PBR, see the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2, "Classification," "Configuring Policy-Based Routing," at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfpbr_ps1835_TSD_Products_Configuration_Guide_Chapter.html

  Note If the RP address falls within the range of a PBR ACL, traffic addressed to the RP is policy routed in hardware instead of being forwarded to the RP. To prevent policy routing of traffic addressed to the RP, configure PBR ACLs to deny traffic addressed to the RP.

- Except on MPLS interfaces, TCP intercept—To configure TCP intercept, see the "Configuring TCP Intercept" section on page 47-2.

Note The PFC3 does not provide hardware acceleration for tunnels configured with the **tunnel key** command.

- GRE Tunneling and IP in IP Tunneling—The PFC3 and DFC3s support the following **tunnel** commands:
   - **tunnel destination**
   - **tunnel mode gre**
   - **tunnel mode ipip**
   - **tunnel source**
   - **tunnel ttl**
   - **tunnel tos**

  Other supported types of tunneling run in software on the RP.

The **tunnel ttl** command (default 255) sets the TTL of encapsulated packets.

The **tunnel tos** command, if present, sets the ToS byte of a packet when it is encapsulated. If the **tunnel tos** command is not present and QoS is not enabled, the ToS byte of a packet sets the ToS byte of the packet when it is encapsulated. If the **tunnel tos** command is not present and QoS is enabled, the ToS byte of a packet as modified by PFC QoS sets the ToS byte of the packet when it is encapsulated.

To configure GRE Tunneling and IP in IP Tunneling, see these publications:

http://www.cisco.com/en/US/docs/ios/12_2/interface/configuration/guide/icflogin.html

http://www.cisco.com/en/US/docs/ios/12_2/interface/command/reference/irfshoip.html

To configure the **tunnel tos** and **tunnel ttl** commands, see this publication for more information:

http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_tos.html

Note the following information about tunnels:

- Each hardware-assisted tunnel must have a unique source. Hardware-assisted tunnels cannot share a source even if the destinations are different. Use secondary addresses on loopback interfaces or create multiple loopback interfaces. Failure to use unique source addresses may result in control plane failures when software path congestion occurs. (CSCdy72539)
- Each tunnel interface uses one internal VLAN.
- Each tunnel interface uses one additional router MAC address entry per router MAC address.
- The PFC3A does not support any PFC QoS features on tunnel interfaces.
- Other PFC versions support PFC QoS features on tunnel interfaces.
- The RP supports tunnels configured with egress features on the tunnel interface. Examples of egress features are output Cisco IOS ACLs, NAT (for inside to outside translation), TCP intercept, and encryption.

• VLAN ACLs (VACLs)—To configure VACLs, see Chapter 51, "Configuring Port ACLs and VLAN ACLs."

PART 1

Configuration Fundamentals

Command-Line Interfaces

This chapter describes the command-line interfaces (CLIs) you use to configure the switches supported by Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Accessing the CLI

These sections describe accessing the CLI:

- Accessing the CLI through the EIA/TIA-232 Console Interface, page 2-2
- Accessing the CLI through Telnet, page 2-2

Accessing the CLI through the EIA/TIA-232 Console Interface

Note EIA/TIA-232 was known as recommended standard 232 (RS-232) before its acceptance as a standard by the Electronic Industries Alliance (EIA) and Telecommunications Industry Association (TIA).

Perform initial configuration over a connection to the EIA/TIA-232 console interface. See the *Catalyst 6500 Series Switch Module Installation Guide* for console interface cable connection procedures.

To make a console connection, perform this task:

	Command	Purpose
Step 1	Press Return.	Brings up the prompt.
Step 2	`Router>` **`enable`**	Initiates enable mode enable.
Step 3	`Password:` *`password`* `Router#`	Completes enable mode enable.
Step 4	`Router#` **`quit`**	Exits the session when finished.

After making a console connection, you see this display:

```
Press Return for Console prompt

Router> enable
Password:
Router#
```

Accessing the CLI through Telnet

Note Before you can make a Telnet connection to the switch, you must configure an IP address (see the "Configuring IPv4 Routing and Addresses" section on page 30-4).

The switch supports up to eight simultaneous Telnet sessions. Telnet sessions disconnect automatically after remaining idle for the period specified with the **exec-timeout** command.

To make a Telnet connection to the switch, perform this task:

	Command	Purpose
Step 1	**telnet** {*hostname* | *ip_addr*}	Makes a Telnet connection from the remote host to the switch you want to access.
Step 2	`Password:` *password* `Router#`	Initiates authentication. **Note** If no password has been configured, press Return.
Step 3	`Router>` **enable**	Initiates enable mode enable.
Step 4	`Password:` *password* `Router#`	Completes enable mode enable.
Step 5	`Router#` **quit**	Exits the session when finished.

This example shows how to open a Telnet session to the switch:

```
unix_host% telnet Router_1
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.

User Access Verification

Password:
Router_1> enable
Password:
Router_1#
```

Performing Command Line Processing

Commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters. You can scroll through the last 20 commands stored in the history buffer, and enter or edit the command at the prompt. Table 2-1 lists the keyboard shortcuts for entering and editing commands.

Table 2-1 Keyboard Shortcuts

Keystrokes	Purpose
Press **Ctrl-B** or press the left arrow key[1]	Moves the cursor back one character.
Press **Ctrl-F** or press the right arrow key[1]	Moves the cursor forward one character.
Press **Ctrl-A**	Moves the cursor to the beginning of the command line.
Press **Ctrl-E**	Moves the cursor to the end of the command line.
Press **Esc B**	Moves the cursor back one word.
Press **Esc F**	Moves the cursor forward one word.

1. The arrow keys function only on ANSI-compatible terminals such as VT100s.

Performing History Substitution

The history buffer stores the last 20 commands you entered. History substitution allows you to access these commands without retyping them, by using special abbreviated commands. Table 2-2 lists the history substitution commands.

Table 2-2 History Substitution Commands

Command	Purpose
Ctrl-P or the up arrow key.[1]	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the down arrow key.[1]	Returns to more recent commands in the history buffer after recalling commands with Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.
`Router# show history`	While in EXEC mode, lists the last several commands you have just entered.

1. The arrow keys function only on ANSI-compatible terminals such as VT100s.

Cisco IOS Command Modes

Note For complete information about Cisco IOS command modes, see the Cisco IOS *Configuration Fundamentals Configuration Guide* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/ffun_c.html

The Cisco IOS user interface is divided into many different modes. The commands available to you depend on which mode you are currently in. To get a list of the commands in a given mode, type a question mark (?) at the system prompt. See the "Displaying a List of Cisco IOS Commands and Syntax" section on page 2-5.

When you start a session on the switch, you begin in user mode, often called user EXEC mode. Only a limited subset of the commands are available in EXEC mode. To have access to all commands, you must enter privileged EXEC mode. Normally, you must type in a password to access privileged EXEC mode. From privileged EXEC mode, you can type in any EXEC command or access global configuration mode.

The configuration modes allow you to make changes to the running configuration. If you later save the configuration, these commands are stored across reboots. You must start at global configuration mode. From global configuration mode, you can enter interface configuration mode, subinterface configuration mode, and a variety of protocol-specific modes.

Note With Release 12.1(11b)E and later, when you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

ROM-monitor mode is a separate mode used when the switch cannot boot properly. For example, the switch might enter ROM-monitor mode if it does not find a valid system image when it is booting, or if its configuration file is corrupted at startup. See the "ROM-Monitor Command-Line Interface" section on page 2-7.

Table 2-3 lists and describes frequently used Cisco IOS modes.

Table 2-3 Frequently Used Cisco IOS Command Modes

Mode	Description of Use	How to Access	Prompt
User EXEC	Connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and display system information.	Log in.	`Router>`
Privileged EXEC (enable)	Set operating parameters. The privileged command set includes the commands in user EXEC mode, as well as the **configure** command. Use this command to access the other command modes.	From the user EXEC mode, enter the **enable** command and the enable password.	`Router#`
Global configuration	Configure features that affect the system as a whole.	From the privileged EXEC mode, enter the **configure terminal** command.	`Router(config)#`
Interface configuration	Many features are enabled for a particular interface. Interface commands enable or modify the operation of an interface.	From global configuration mode, enter the **interface** *type slot/port* command.	`Router(config-if)#`
Console configuration	From the directly connected console or the virtual terminal used with Telnet, use this configuration mode to configure the console interface.	From global configuration mode, enter the **line console 0** command.	`Router(config-line)#`

The Cisco IOS command interpreter, called the EXEC, interprets and executes the commands you enter. You can abbreviate commands and keywords by entering just enough characters to make the command unique from other commands. For example, you can abbreviate the **show** command to **sh** and the **configure terminal** command to **config t**.

When you type **exit**, the switch backs out one level. To exit configuration mode completely and return to privileged EXEC mode, press **Ctrl-Z**.

Displaying a List of Cisco IOS Commands and Syntax

In any command mode, you can display a list of available commands by entering a question mark (?).

```
Router> ?
```

To display a list of commands that begin with a particular character sequence, type in those characters followed by the question mark (?). Do not include a space. This form of help is called word help because it completes a word for you.

```
Router# co?
```

```
collect  configure  connect  copy
```

To display keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark. This form of help is called command syntax help because it reminds you which keywords or arguments are applicable based on the command, keywords, and arguments you have already entered.

For example:

```
Router# configure ?
  memory             Configure from NV memory
  network            Configure from a TFTP network host
  overwrite-network  Overwrite NV memory from TFTP network host
  terminal           Configure from the terminal
  <cr>
```

To redisplay a command you previously entered, press the up arrow key or **Ctrl-P**. You can continue to press the up arrow key to see the last 20 commands you entered.

Tip If you are having trouble entering a command, check the system prompt, and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

Enter **exit** to return to the previous mode. Press **Ctrl-Z** or enter the **end** command in any mode to immediately return to privileged EXEC mode.

Securing the CLI

Securing access to the CLI prevents unauthorized users from viewing configuration settings or making configuration changes that can disrupt the stability of your network or compromise your network security. You can create a strong and flexible security scheme for your switch by configuring one or more of these security features:

- Protecting access to privileged EXEC commands

  At a minimum, you should configure separate passwords for the user EXEC and privileged EXEC (enable) IOS command modes. You can further increase the level of security by configuring username and password pairs to limit access to CLI sessions to specific users. For more information, see "Configuring Security with Passwords, Privilege Levels, and Login Usernames for CLI Sessions on Networking Devices" at this URL:

  http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_cfg_sec_4cli.html

- Controlling switch access with RADIUS, TACACS+, or Kerberos

  For a centralized and scalable security scheme, you can require users to be authenticated and authorized by an external security server running either Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access-Control System Plus (TACACS+), or Kerberos.

  For more information about RADIUS, see "Configuring RADIUS" at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfrad.html

  For more information about TACACS+, see "Configuring TACACS+" at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scftplus.html

  For more information about Kerberos, see "Configuring Kerberos" at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfkerb.html

- Configuring a secure connection with SSH or HTTPS

  To prevent eavesdropping of your configuration session, you can use a Secure Shell (SSH) client or a browser that supports HTTP over Secure Socket Layer (HTTPS) to make an encrypted connection to the switch.

  For more information about SSH, see "Configuring Secure Shell" at this URL:

  http://www.cisco.com/en/US/docs/ios-xml/ios/sec_usr_ssh/configuration/12-2sx/sec-secure-copy.html

  For more information about HTTPS, see "HTTPS - HTTP Server and Client with SSL 3.0" at this URL:

  http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_cfg_sec_4cli.html

- Copying configuration files securely with SCP

  To prevent eavesdropping when copying configuration files or image files to or from the switch, you can use the Secure Copy Protocol (SCP) to perform an encrypted file transfer. For more information about SCP, see "Secure Copy" at this URL:

  http://www.cisco.com/en/US/docs/ios-xml/ios/sec_usr_ssh/configuration/12-2sy/sec-usr-ssh-sec-copy.html

For additional information about securing the CLI, see "Cisco IOS Security Configuration Guide: Securing User Services, Release 12.2SX" at this URL:

http://www.cisco.com/en/US/docs/ios-xml/ios/security/config_library/12-2sx/secuser-12-2sx-library.html

ROM-Monitor Command-Line Interface

The ROM-monitor is a ROM-based program that executes upon platform power-up, reset, or when a fatal exception occurs. The switch enters ROM-monitor mode if it does not find a valid software image, if the NVRAM configuration is corrupted, or if the configuration register is set to enter ROM-monitor mode. From the ROM-monitor mode, you can load a software image manually from flash memory, from a network server file, or from bootflash.

You can also enter ROM-monitor mode by restarting and pressing the **Break** key during the first 60 seconds of startup.

Note The **Break** key is always enabled for 60 seconds after rebooting, regardless of whether the **Break** key is configured to be off by configuration register settings.

To access the ROM-monitor mode through a terminal server, you can escape to the Telnet prompt and enter the **send break** command for your terminal emulation program to break into ROM-monitor mode.

Once you are in ROM-monitor mode, the prompt changes to rommon 1>. Enter a question mark (**?**) to see the available ROM-monitor commands.

For more information about the ROM-monitor commands, see the Cisco IOS Master Command List.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Smart Port Macros

This chapter describes how to configure and apply smart port macros. Release 12.2(33)SXH and later releases support smart port macros.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Smart Port Macros

These sections describe smart port macros:

Understanding Cisco-Provided Smart Port Macros

There are Cisco-provided smart port macros embedded in the switch software (see Table 3-1). You can display these macros and the commands they contain by using the **show parser macro** user EXEC command.

Table 3-1 Cisco-Provided Smart Port Macros

Macro Name	Description
cisco-global	Use this global configuration macro to enable load balancing across VLANs, provide rapid convergence of spanning-tree instances and to enable port error recovery.
cisco-desktop	Use this interface configuration macro for increased network security and reliability when connecting a desktop device, such as a PC, to a switch port.
cisco-phone	Use this interface configuration macro when connecting a desktop device such as a PC with a Cisco IP phone to a switch port. This macro is an extension of the **cisco-desktop** macro and provides the same security and resiliency features, but with the addition of dedicated voice VLANs to ensure proper treatment of delay-sensitive voice traffic.
cisco-switch	Use this interface configuration macro for Layer 2 connections between devices like switches and routers.
cisco-router	Use this interface configuration macro for Layer 3 connections between devices like switches and routers.

Cisco also provides a collection of pretested, Cisco-recommended baseline configuration templates for Catalyst switches. The online reference guide templates provide the CLI commands that you can use to create smart port macros based on the usage of the port. You can use the configuration templates to create smart port macros to build and deploy Cisco-recommended network designs and configurations.

Understanding User-Created Smart Port Macros

Smart port macros provide a convenient way to save and share common configurations. You can use smart port macros to enable features and settings based on the location of a switch in the network and for mass configuration deployments across the network.

Each smart port macro is a user-defined set of Cisco IOS CLI commands. When you apply a smart port macro on an interface, the CLI commands within the macro are configured on the interface. When the macro is applied to an interface, the existing interface configurations are not lost. The new commands are added to the interface and are saved in the running configuration file.

Configuring Smart Port Macros

- Smart Port Macro Default Configuration, page 3-2
- Smart Port Macro Configuration Guidelines, page 3-3
- Applying the Cisco-Provided Smart Port Macros, page 3-4
- Configuring User-Created Smart Port Macros, page 3-13

Smart Port Macro Default Configuration

This example shows how to list the Cisco-provided smart port macros:

```
Router# show parser macro brief | include default
    default global    : cisco-global
```

```
default interface: cisco-desktop
default interface: cisco-phone
default interface: cisco-switch
default interface: cisco-router
```

There are no smart port macros applied to any interfaces.

Smart Port Macro Configuration Guidelines

These section describe the smart port macro configuration guidelines:

- You can display all of the macros on the switch by using the **show parser macro** user EXEC command. Display the contents of a specific macro by using the **show parser macro name** *macro-name* user EXEC command.
- A macro cannot be edited. If the name following the **macro name** command is an existing macro's name, that macro is replaced by the new macro.
- If a description already exists for a macro, the **macro description** command appends any description that you enter to the existing description; it does not replace it. The entered descriptions are separated by the pipe ("|") character.
- The maximum macro description length is 256 characters. When the description string becomes longer than 256 characters, the oldest descriptions are deleted to make room for new ones.
- User-created recursive macros are not supported. You cannot define a macro that calls another macro.
- Each user-created macro can have up to three keyword-value pairs.
- A macro definition can contain up to 3,000 characters. Line endings count as two characters.
- When creating a macro, do not use the **exit** or **end** commands or change the command mode by using **interface** *interface-id*. This could cause commands that follow **exit**, **end**, or **interface** *interface-id* to execute in a different command mode. When creating a macro, all CLI commands should be in the same configuration mode.
- When creating a macro that requires the assignment of unique values, use the **parameter** *value* keywords to designate values specific to the interface. Keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. Any full match of a keyword, even if it is part of a larger string, is considered a match and is replaced by the corresponding value.
- Macro names are case sensitive. For example, the commands **macro name Sample-Macro** and **macro name sample-macro** will result in two separate macros.
- Some macros might contain keywords that require a parameter value. You can use the **macro global apply** *macro-name* **?** global configuration command or the **macro apply** *macro-name* **?** interface configuration command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.
- When a macro is applied globally to a switch or to a switch interface, the existing configuration on the interface is retained. This is helpful when applying an incremental configuration.
- If you modify a macro definition by adding or deleting commands, the changes are not reflected on the interface where the original macro was applied. You need to reapply the updated macro on the interface to apply the new or changed commands.

- You can use the **macro global trace** *macro-name* global configuration command or the **macro trace** *macro-name* interface configuration command to apply and debug a macro to find any syntax or configuration errors. If a command fails because of a syntax error or a configuration error, the macro continues to apply the remaining commands.
- Some CLI commands are specific to certain interface types. If a macro is applied to an interface that does not accept the configuration, the macro will fail the syntax check or the configuration check, and the switch will return an error message.
- Applying a macro to an interface range is the same as applying a macro to a single interface. When you use an interface range, the macro is applied sequentially to each interface within the range. If a macro command fails on one interface, it is still applied to the remaining interfaces.
- When you apply a macro to a switch or a switch interface, the macro name is automatically added to the switch or interface. You can display the applied commands and macro names by using the **show running-config** user EXEC command.

Applying the Cisco-Provided Smart Port Macros

These sections describe how to apply the Cisco-provided smart port macros:

- Using the cisco-global Smart Port Macro, page 3-4
- Using the cisco-desktop Smart Port Macro, page 3-5
- Using the cisco-phone Smart Port Macro, page 3-7
- Using the cisco-switch Smart Port Macro, page 3-9
- Using the cisco-router Smart Port Macro, page 3-11

Using the cisco-global Smart Port Macro

These sections describe how to apply the cisco-global smart port macro:

- Displaying the Contents of the cisco-global Smart Port Macro, page 3-4
- Applying the cisco-global Smart Port Macro, page 3-5

Displaying the Contents of the cisco-global Smart Port Macro

This example shows how to display the contents of the cisco-global smart port macro:

```
Router# show parser macro name cisco-global
Macro name : cisco-global
Macro type : default global
# Enable dynamic port error recovery for link state
# failures
errdisable recovery cause link-flap
errdisable recovery interval 60

# VTP requires Transparent mode for future 802.1x Guest VLAN
# and current Best Practice
vtp domain [smartports]
vtp mode transparent

# Config Cos to DSCP mappings
mls qos map cos-dscp 0 8 16 26 32 46 48 56

# Enable aggressive mode UDLD on all fiber uplinks
udld aggressive
```

```
# Enable Rapid PVST+ and Loopguard
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
```

Applying the cisco-global Smart Port Macro

To apply the cisco-global smart port macro, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **macro global apply cisco-global**	Applies the cisco-global smart port macro.
Step 3	Router(config)# **end**	Returns to privileged EXEC mode.
Step 4	Router# **show parser macro description**	Displays the macros that have been applied.

This example shows how to apply the cisco-global smart port macro and display the name of the applied macro:

```
Router# configure terminal
Router(config)# macro global apply cisco-global
Changing VTP domain name from previous_domain_name to [smartports]
Setting device to VTP TRANSPARENT mode.
Router(config)# end
Router# show parser macro description
Global Macro(s): cisco-global

Interface    Macro Description(s)
--------------------------------------------------------------
--------------------------------------------------------------
Router#
```

Using the cisco-desktop Smart Port Macro

These sections describe how to apply the cisco-desktop smart port macro:

- Displaying the Contents of the cisco-desktop Smart Port Macro, page 3-5
- Applying the cisco-desktop Smart Port Macro, page 3-6

Displaying the Contents of the cisco-desktop Smart Port Macro

This example shows how to display the contents of the cisco-desktop smart port macro:

```
Router# show parser macro name cisco-desktop
Macro name : cisco-desktop
Macro type : default interface
# macro keywords $AVID
# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport
switchport access vlan $AVID
switchport mode access

# Enable port security limiting port to a single
# MAC address -- that of desktop
switchport port-security
```

```
switchport port-security maximum 1

# Ensure port-security age is greater than one minute
# and use inactivity timer
switchport port-security violation restrict
switchport port-security aging time 2
# Configure port as an edge network port
spanning-tree portfast
spanning-tree portfast edge bpduguard default
```

Applying the cisco-desktop Smart Port Macro

To apply the cisco-desktop smart port macro, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface** *type*[1] *slot/port*	Selects the interface to configure.
Step 3	Router(config-if)# **macro apply cisco-desktop $AVID** *access_vlan_ID*	Applies the cisco-desktop smart port macro. The recommended range for *access_vlan_ID* is 2–4094.
Step 4	Router(config-if)# **end**	Returns to privileged EXEC mode.
Step 5	Router# **show parser macro description interface** *type*[1] *slot/port*	Displays the macros that have been applied.
Step 6	Router# **show running-config interface** *type*[1] *slot/port*	Displays all of the commands configured on the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to apply the cisco-desktop smart port macro to Gigabit Ethernet port 1/1 with VLAN 2 specified as the access VLAN and how to verify the result:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/1
Router(config-if)# macro apply cisco-desktop $AVID 2
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on GigabitEthernet1/1 but will only
 have effect when the interface is in a non-trunking mode.
Router(config)# end
Router# show parser macro description interface gigabitethernet 1/1
Global Macro(s): cisco-global

Interface    Macro Description(s)
--------------------------------------------------------------
Gi1/1        cisco-desktop
--------------------------------------------------------------
Router# show running-config interface gigabitethernet 1/1
Building configuration...

Current configuration : 307 bytes
!
interface GigabitEthernet1/1
 switchport
 switchport access vlan 2
 switchport mode access
```

```
 switchport port-security
 switchport port-security aging time 2
 switchport port-security violation restrict
 shutdown
 macro description cisco-desktop
 spanning-tree portfast
 spanning-tree bpduguard enable
end

Router#
```

Using the cisco-phone Smart Port Macro

These sections describe how to apply the cisco-phone smart port macro:

- Displaying the Contents of the cisco-phone Smart Port Macro, page 3-7
- Applying the cisco-phone Smart Port Macro, page 3-8

Displaying the Contents of the cisco-phone Smart Port Macro

This example shows how to display the contents of the cisco-phone smart port macro:

```
Router# show parser macro name cisco-phone
Macro name : cisco-phone
Macro type : default interface
# macro keywords $AVID $VVID
# VoIP enabled interface - Enable data VLAN
# and voice VLAN (VVID)
# Recommended value for access vlan (AVID) should not be 1
switchport
switchport access vlan $AVID
switchport mode access

# Update the Voice VLAN (VVID) value which should be
# different from data VLAN
# Recommended value for voice vlan (VVID) should not be 1
switchport voice vlan $VVID

# Enable port security limiting port to a 3 MAC
# addressess -- One for desktop and two for phone
switchport port-security
switchport port-security maximum 3

# Ensure port-security age is greater than one minute
# and use inactivity timer
switchport port-security violation restrict
switchport port-security aging time 2
# Enable auto-qos to extend trust to attached Cisco phone
auto qos voip cisco-phone

# Configure port as an edge network port
spanning-tree portfast
spanning-tree portfast edge bpduguard default
```

Applying the cisco-phone Smart Port Macro

To apply the cisco-phone smart port macro, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface type[1] slot/port`	Selects the interface to configure.
Step 3	`Router(config-if)# macro apply cisco-phone $AVID access_vlan_ID $VVID voice_vlan_ID`	Applies the cisco-phone smart port macro. The recommended range for *access_vlan_ID* is 2–4094. The recommended range for *voice_vlan_ID* is 2–4094.
Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Router# show parser macro description interface type[1] slot/port`	Displays the macros that have been applied.
Step 6	`Router# show running-config interface type[1] slot/port`	Displays all of the commands configured on the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When applying the cisco-phone smart port macro, note the following information:

- Some of the generated commands are in the category of PFC QoS commands that are applied to all ports controlled by a port ASIC. When one of these generated commands is applied, PFC QoS displays the messages caused by application of the command to all the ports controlled by the port ASIC. Depending on the module, these commands are applied to as many as 48 ports. See the “Number of port groups” and “Port ranges per port group” listed for each module in the *Release Notes for Cisco IOS Release 12.2SX.*
- You might see messages that instruct you to configure other ports to trust CoS. You must do so to enable the generated QoS commands.
- You might not be able to apply the cisco-phone smart port macro and other macros on ports that are controlled by the same port ASIC because of conflicting port trust state requirements.

This example shows how to apply the cisco-phone smart port macro to Gigabit Ethernet port 2/2 with VLAN 2 specified as the access VLAN and how to verify the result:

```
Router# configure terminal
Router(config)# interface gigabitethernet 2/2
Router(config-if)# macro apply cisco-phone $AVID 2 $VVID 3
Hardware QoS is enabled
Propagating cos-map to inband port
Propagating cos-map configuration to: [port list not shown]
```

[Output for other ports controlled by the same port ASIC omitted]

```
Warning: rcv cosmap will not be applied in hardware.
  To modify rcv cosmap in hardware, all of the interfaces below
  must be put into 'trust cos' state:
    [port list not shown]
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on GigabitEthernet1/2 but will only
 have effect when the interface is in a non-trunking mode.
Router(config)# end
```

```
Router# show parser macro description interface gigabitethernet 2/2
Global Macro(s): cisco-global

Interface    Macro Description(s)
--------------------------------------------------------------
Gi2/2        cisco-phone
--------------------------------------------------------------

Router# show running-config interface gigabitethernet 2/2
Building configuration...

Building configuration...

Current configuration : 307 bytes
!
interface GigabitEthernet1/2
Building configuration...

Current configuration : 1336 bytes
!
interface GigabitEthernet2/2
 switchport
 switchport access vlan 2
 switchport mode access
 switchport voice vlan 3
 switchport port-security
 switchport port-security maximum 3
 switchport port-security aging time 2
 switchport port-security violation restrict
 shutdown
```

[QoS queuing commands omitted: these vary according to port type]

```
 mls qos trust cos
 auto qos voip cisco-phone
 macro description cisco-phone
 spanning-tree portfast
 spanning-tree bpduguard enable
end

Router#
```

Using the cisco-switch Smart Port Macro

These sections describe how to apply the cisco-switch smart port macro:

- Displaying the Contents of the cisco-switch Smart Port Macro, page 3-9
- Applying the cisco-switch Smart Port Macro, page 3-10

Displaying the Contents of the cisco-switch Smart Port Macro

This example shows how to display the contents of the cisco-switch smart port macro:

```
Router# show parser macro name cisco-switch
Macro name : cisco-switch
Macro type : default interface
# macro keywords $NVID
# Do not apply to EtherChannel/Port Group
# Access Uplink to Distribution

# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
```

```
switchport
switchport trunk native vlan $NVID

# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan VRANGE

# Hardcode trunk and disable negotiation to
# speed up convergence
switchport trunk encapsulation dot1q
switchport mode trunk
switchport nonegotiate

# 802.1w defines the link as pt-pt for rapid convergence
spanning-tree link-type point-to-point

Router#
```

Applying the cisco-switch Smart Port Macro

To apply the cisco-switch smart port macro, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface** *type*[1] *slot/port*	Selects the interface to configure.
Step 3	Router(config-if)# **macro apply cisco-switch $NVID** *native_vlan_ID*	Applies the cisco-switch smart port macro. The recommended range for *native_vlan_ID* is 2–4094.
Step 4	Router(config-if)# **end**	Returns to privileged EXEC mode.
Step 5	Router# **show parser macro description interface** *type*[1] *slot/port*	Displays the macros that have been applied.
Step 6	Router# **show running-config interface** *type*[1] *slot/port*	Displays all of the commands configured on the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to apply the cisco-switch smart port macro to Gigabit Ethernet port 1/4 with VLAN 4 specified as the native VLAN and how to verify the result:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/4
Router(config-if)# macro apply cisco-switch $NVID 4
Router(config-if)# end
Router# show parser macro description interface gigabitethernet 1/4
Interface    Macro Description(s)
--------------------------------------------------------------
Gi1/4        cisco-switch
--------------------------------------------------------------
Router# show running-config interface gigabitethernet 1/4
Building configuration...

Current configuration : 247 bytes
!
interface GigabitEthernet1/4
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 4
 switchport mode trunk
```

```
 switchport nonegotiate
 shutdown
 macro description cisco-switch
 spanning-tree link-type point-to-point
end

Router#
```

Using the cisco-router Smart Port Macro

These sections describe how to apply the cisco-router smart port macro:

- Displaying the Contents of the cisco-router Smart Port Macro, page 3-11
- Applying the cisco-router Smart Port Macro, page 3-12

Displaying the Contents of the cisco-router Smart Port Macro

This example shows how to display the contents of the cisco-router smart port macro:

```
Router# show parser macro name cisco-router
Macro name : cisco-router
Macro type : default interface
# macro keywords $NVID
# Do not apply to EtherChannel/Port Group
# Access Uplink to Distribution
switchport

# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID

# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan VRANGE

# Hardcode trunk and disable negotiation to
# speed up convergence
switchport trunk encapsulation dot1q
switchport mode trunk
switchport nonegotiate

# Configure qos to trust this interface
auto qos voip trust
mls qos trust dscp

# Ensure fast access to the network when enabling the interface.
# Ensure that switch devices cannot become active on the interface.
spanning-tree portfast
spanning-tree portfast edge bpduguard default

Router#
```

Applying the cisco-router Smart Port Macro

To apply the cisco-router smart port macro, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the interface to configure.
Step 3	`Router(config-if)#` **`macro apply cisco-router $NVID`** *`native_vlan_ID`*	Applies the cisco-router smart port macro. The recommended range for *native_vlan_ID* is 2–4094.
Step 4	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Router#` **`show parser macro description interface`** *`type`*[1] *`slot/port`*	Displays the macros that have been applied.
Step 6	`Router#` **`show running-config interface`** *`type`*[1] *`slot/port`*	Displays all of the commands configured on the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note The cisco-router smart port macro includes the **auto qos voip trust** command. When entered on a port configured with the **switchport** command, the **auto qos voip trust** command generates and applies the **mls qos trust cos** command to the port, but the cisco-router smart port macro changes the port trust state to trust DSCP with the **mls qos trust dscp** command. When you apply the cisco-router smart port macro, ignore messages that instruct you to enter the **mls qos trust cos** command on other ports controlled by the port ASIC.

This example shows how to apply the cisco-router smart port macro to Gigabit Ethernet port 1/5 and how to verify the result:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/5
Router(config-if)# macro apply cisco-router $NVID 5
Hardware QoS is enabled
Propagating cos-map to inband port
Propagating cos-map configuration to: [port list not shown]
```

[Output for other ports controlled by the same port ASIC omitted]

[Output from temporarily applied trust CoS command omitted]

```
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on GigabitEthernet1/5 but will only
 have effect when the interface is in a non-trunking mode.
Router(config-if)# end
Router# show parser macro description interface gigabitethernet 1/5
Interface    Macro Description(s)
--------------------------------------------------------------
Gi1/5        cisco-router
--------------------------------------------------------------
Router# show running-config interface gigabitethernet 1/5
Building configuration...

Current configuration : 1228 bytes
```

```
!
interface GigabitEthernet1/5
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 5
 switchport mode trunk
 switchport nonegotiate
 shutdown
 wrr-queue bandwidth 20 100 200
```

[QoS queuing commands omitted: these vary according to port type]

```
 mls qos trust dscp
 auto qos voip trust
 macro description cisco-router
 spanning-tree portfast
 spanning-tree bpduguard enable
end

Router#
```

Configuring User-Created Smart Port Macros

These sections describe how to configure user-created smart port macros:

- Creating Smart Port Macros, page 3-13
- Applying Smart Port Macros, page 3-14

Creating Smart Port Macros

To create a smart port macro, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`macro name`** *`macro-name`*	Creates a macro. Macro names are case sensitive. For example, the commands **macro name Sample-Macro** and **macro name sample-macro** will result in two separate macros. A macro definition can contain up to 3,000 characters. Line endings count as two characters. There is no prompt displayed in macro creation mode. Enter the macro commands on separate lines. Use the **#** character at the beginning of a line to enter a comment within the macro. Use the @ character to end the macro. Do not use the **exit** or **end** commands or change the command mode with the **interface** *interface-id* in a macro. This could cause any commands following **exit**, **end**, or **interface** *interface-id* to execute in a different command mode. For best results, all commands in a macro should be in the same configuration mode. Each user-created macro can have up to three keyword-value pairs.
Step 3	`#` **`macro keywords`** *`keyword1 keyword2 keyword3`*	(Optional) You can create a help string to describe the keywords that you define in the macro. You can enter up to three help string comments in a macro.
Step 4	**`end`**	Returns to privileged EXEC mode.
Step 5	**`show parser macro name`** *`macro-name`*	Verifies that the macro was created.

Note The **no** form of the **macro name** global configuration command only deletes the macro definition. It does not affect the configuration of those interfaces on which the macro is already applied.

This example shows how to create a macro that defines the Layer 2 access VLAN and the number of secure MAC addresses and also includes two help string keywords by using **# macro keywords**:

```
Router(config)# macro name test
#macro keywords $VLANID $MAX
switchport access vlan $VLANID
switchport port-security maximum $MAX
@
```

Applying Smart Port Macros

To apply a smart port macro, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# default interface interface-id`	(Optional) Clears all configuration from the specified interface.
Step 3	`Router(config)# interface interface_id`	(Required for interface macros.) Specifies the interface on which to apply the macro and enters interface configuration mode.

Step 4	`Router(config)# macro global {apply | trace} macro-name [keyword value] [keyword value] [keyword value]`
	Or
	`Router(config-if)# macro {apply | trace} macro-name [keyword value] [keyword value] [keyword value]`
		Applies or traces and applies each individual command defined in the macro.
		For global macros:
		• To find any syntax or configuration errors, enter the **macro global trace** *macro-name* command to apply and debug the macro.
		• To display a list of any keyword-value pairs defined in the macro, enter the **macro global apply macro-name ?** command.
		For interface macros:
		• To find any syntax or configuration errors, enter the **macro trace** *macro-name* command to apply and debug the macro.
		• To display a list of any keyword-value pairs defined in the macro, enter the **macro apply macro-name ?** command.
		To successfully apply the macro, you must enter any required keyword-value pairs.
		Keyword matching is case sensitive.
		In the commands that the macro applies, all matching occurrences of keywords are replaced with the corresponding values.

Step 5	`Router(config)# end`	Returns to privileged EXEC mode.
Step 6	`Router# show parser macro description [interface interface_id]`	Verifies that the macro is applied to the interface.

You can delete a global macro-applied configuration on a switch only by entering the **no** version of each command that is in the macro. You can delete all configurations on an interface by entering the **default interface** *interface_id* interface configuration command.

This example shows how to apply the user-created macro called snmp, to set the host name address to **test-server** and to set the IP precedence value to 7:

```
Router(config)# macro global apply snmp ADDRESS test-server VALUE 7
```

This example shows how to debug the user-created macro called snmp by using the **macro global trace** global configuration command to find any syntax or configuration errors in the macro when it is applied to the switch:

```
Router(config)# macro global trace snmp VALUE 7
Applying command...'snmp-server enable traps port-security'
```

```
Applying command...'snmp-server enable traps linkup'
Applying command...'snmp-server enable traps linkdown'
Applying command...'snmp-server host'
%Error Unknown error.
Applying command...'snmp-server ip precedence 7'
```

This example shows how to apply the user-created macro called desktop-config and to verify the configuration:

```
Router(config)# interface fastethernet1/2
Router(config-if)# macro apply desktop-config
Router(config-if)# end
Router# show parser macro description
Interface    Macro Description
--------------------------------------------------------------
Fa1/2      desktop-config
--------------------------------------------------------------
```

This example shows how to apply the user-created macro called desktop-config and to replace all occurrences of *vlan* with VLAN ID 25:

```
Router(config-if)# macro apply desktop-config vlan 25
```

Displaying Smart Port Macros

To display the smart port macros, use one or more of the privileged EXEC commands in Table 3-2.

Table 3-2 Commands for Displaying Smartports Macros

Command	Purpose
show parser macro	Displays all configured macros.
show parser macro name *macro-name*	Displays a specific macro.
show parser macro brief	Displays the configured macro names.
show parser macro description [**interface** *interface-id*]	Displays the macro description for all interfaces or for a specified interface.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Virtual Switching Systems

This chapter describes how to configure a virtual switching system (VSS) for the Catalyst 6500 series switch. Cisco IOS Release 12.2(33)SXH1 and later releases support VSS.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The *Cisco IOS Virtual Switch Command Reference* at this URL:

  http://www.cisco.com/en/US/docs/ios/vswitch/command/reference/vs_book.html

- The Cisco IOS Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

---

Tip For additional information about Cisco Catalyst 6500 series switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

---

This chapter consists of these sections:

Understanding Virtual Switching Systems

These sections describe a VSS:

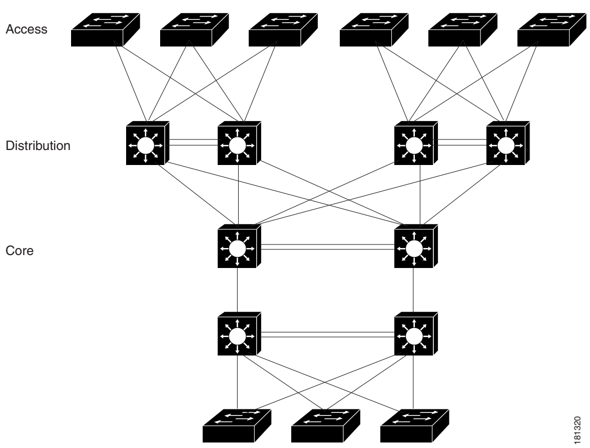
VSS Overview

Network operators increase network reliability by configuring redundant pairs of network devices and links. Figure 4-1 shows a typical switch network configuration. Redundant network elements and redundant links can add complexity to network design and operation. Virtual switching simplifies the network by reducing the number of network elements and hiding the complexity of managing redundant switches and links.

A VSS combines a pair of Catalyst 6500 series switches into a single network element. The VSS manages the redundant links, which externally act as a single port channel.

The VSS simplifies network configuration and operation by reducing the number of Layer 3 routing neighbors and by providing a loop-free Layer 2 topology.

Figure 4-1 Typical Switch Network Design



The following sections present an overview of the VSS. These topics are covered in detail in subsequent chapters:

Key Concepts

The VSS incorporates the following key concepts:

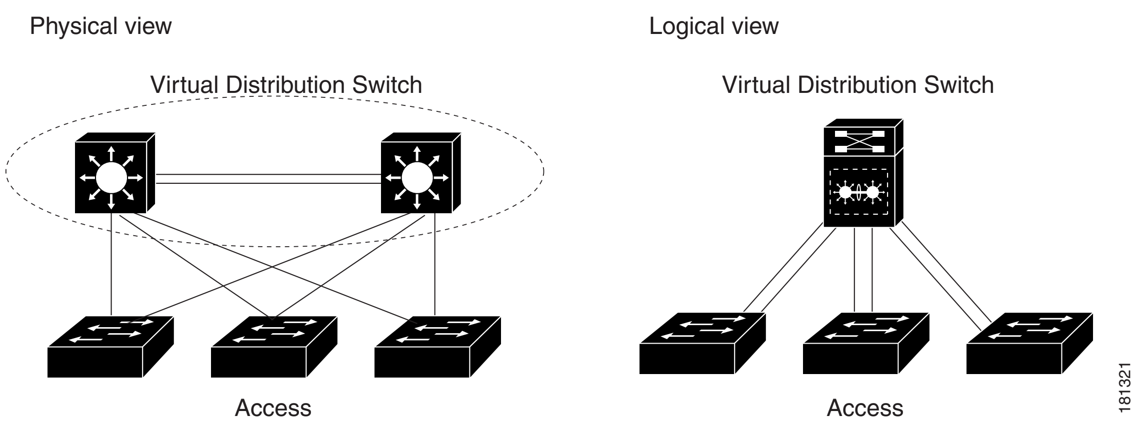
- Virtual Switching System, page 4-3
- VSS Active and VSS Standby Chassis, page 4-3
- Virtual Switch Link, page 4-4
- Multichassis EtherChannel, page 4-5

Virtual Switching System

A VSS combines a pair of switches into a single network element. For example, a VSS in the distribution layer of the network interacts with the access and core networks as if it were a single switch. See Figure 4-2.

An access switch connects to both chassis of the VSS using one logical port channel. The VSS manages redundancy and load balancing on the port channel. This capability enables a loop-free Layer 2 network topology. The VSS also simplifies the Layer 3 network topology because the VSS reduces the number of routing peers in the network.

Figure 4-2 VSS in the Distribution Network



VSS Active and VSS Standby Chassis

When you create or restart a VSS, the peer chassis negotiate their roles. One chassis becomes the VSS active chassis, and the other chassis becomes the VSS standby.

The VSS active chassis controls the VSS. It runs the Layer 2 and Layer 3 control protocols for the switching modules on both chassis. The VSS active chassis also provides management functions for the VSS, such as module online insertion and removal (OIR) and the console interface.

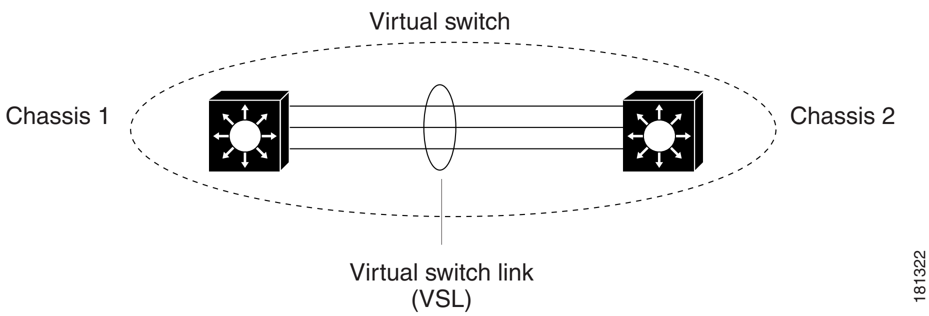
The VSS active and VSS standby chassis perform packet forwarding for ingress data traffic on their locally hosted interfaces. However, the VSS standby chassis sends all control traffic to the VSS active chassis for processing.

Virtual Switch Link

For the two chassis of the VSS to act as one network element, they need to share control information and data traffic.

The virtual switch link (VSL) is a special link that carries control and data traffic between the two chassis of a VSS, as shown in Figure 4-3. The VSL is implemented as an EtherChannel with up to eight links. The VSL gives control traffic higher priority than data traffic so that control messages are never discarded. Data traffic is load balanced among the VSL links by the EtherChannel load-balancing algorithm.

Figure 4-3 Virtual Switch Link



When you configure VSL all existing configurations are removed from the interface except for specific allowed commands. When you configure VSL, the system puts the interface into a restricted mode. When an interface is in restricted mode, only specific configuration commands can be configured on the interface.

The following VSL configuration commands are not removed from the interface when it becomes restricted:

- **mls qos trust cos**
- **mls qos channel-consistency**
- **description**
- **logging event**
- **load-interval**
- **vslp**
- **port-channel port**

When in VSL restricted mode, only these configuration commands are available:

- **channel-group**
- **default**
- **description**
- **exit**
- **load-interval**
- **logging**
- **mls**
- **mls ip**

- **mls ipx**
- **mls netflow**
- **mls rp**
- **mls switching**
- **no**
- **shutdown**

Note The **mls qos** command is not available when a port is in VSL restricted mode.

Multichassis EtherChannel

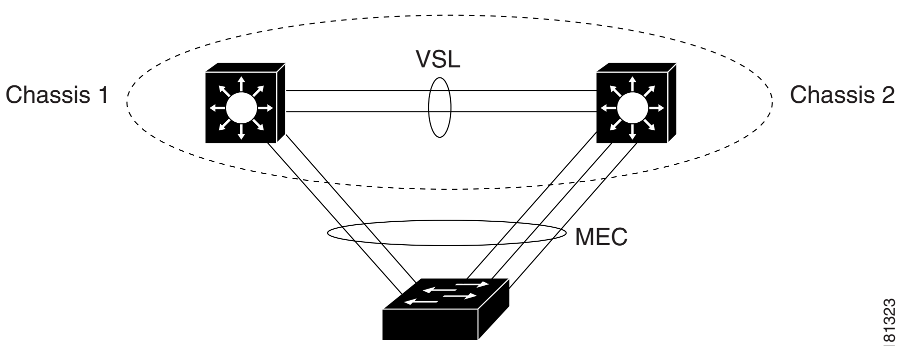
An EtherChannel (also known as a port channel) is a collection of two or more physical links that combine to form one logical link. Layer 2 protocols operate on the EtherChannel as a single logical entity.

A multichassis EtherChannel (MEC) is a port channel that spans the two chassis of a VSS. The access switch views the MEC as a standard port channel. See Figure 4-4.

The VSS supports a maximum of 512 EtherChannels. This limit applies to the combined total of regular EtherChannels and MECs. Because VSL requires two EtherChannel numbers (one for each chassis), there are 510 user-configurable EtherChannels. If an installed service module uses an internal EtherChannel, that EtherChannel will be included in the total.

Note For releases earlier than Cisco IOS Release 12.2(33)SXI, the maximum number of EtherChannels is 128, allowing 126 user-configurable EtherChannels.

Figure 4-4 VSS with MEC



VSS Functionality

The following sections describe the main functionality of a VSS:

- Redundancy and High Availability, page 4-6
- Packet Handling, page 4-6
- System Management, page 4-6
- VSS Quad-Sup Uplink Forwarding, page 4-7
- Interface Naming Convention, page 4-8
- Software Features, page 4-8

Redundancy and High Availability

In a VSS, supervisor engine redundancy operates between the VSS active and VSS standby chassis, using stateful switchover (SSO) and nonstop forwarding (NSF). The peer chassis exchange configuration and state information across the VSL and the VSS standby supervisor engine runs in hot VSS standby mode.

The VSS standby chassis monitors the VSS active chassis using the VSL. If it detects failure, the VSS standby chassis initiates a switchover and takes on the VSS active role. When the failed chassis recovers, it takes on the VSS standby role.

If the VSL fails completely, the VSS standby chassis assumes that the VSS active chassis has failed, and initiates a switchover. After the switchover, if both chassis are VSS active, the dual-active detection feature detects this condition and initiates recovery action. For additional information about dual-active detection, see the "Dual-Active Detection" section on page 4-22.

Packet Handling

The VSS active supervisor engine runs the Layer 2 and Layer 3 protocols and features for the VSS and manages the DFC modules for both chassis.

The VSS uses VSL to communicate protocol and system information between the peer chassis and to carry data traffic between the chassis when required.

Both chassis perform packet forwarding for ingress traffic on their interfaces. If possible, ingress traffic is forwarded to an outgoing interface on the same chassis to minimize data traffic that must traverse the VSL.

Because the VSS standby chassis is actively forwarding traffic, the VSS active supervisor engine distributes updates to the VSS standby supervisor engine PFC and all VSS standby chassis DFCs.

System Management

The VSS active supervisor engine acts as a single point of control for the VSS. For example, the VSS active supervisor engine handles OIR of switching modules on both chassis. The VSS active supervisor engine uses VSL to send messages to and from local ports on the VSS standby chassis.

The command console on the VSS active supervisor engine is used to control both chassis. In virtual switch mode, the command console on the VSS standby supervisor engine blocks attempts to enter configuration mode.

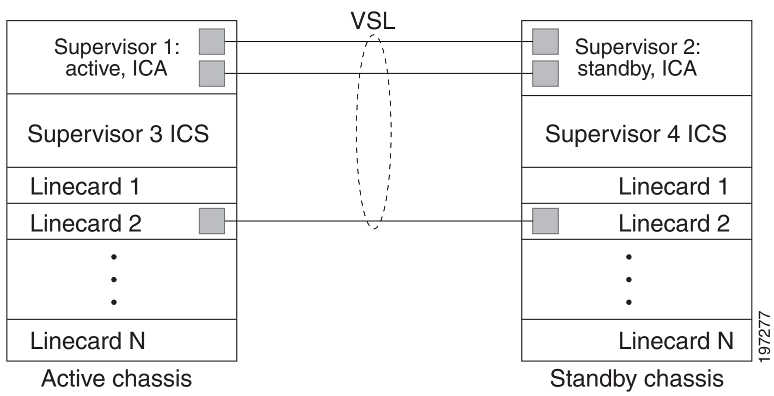
The VSS standby chassis runs a subset of system management tasks. For example, the VSS standby chassis handles its own power management.

VSS Quad-Sup Uplink Forwarding

When you use VSS quad-supervisor uplink forwarding, the in-chassis standby (ICS) supervisor engine acts as a DFC line card. Only one processor, the SP processor, acts as the DFC line card; the RP processor is reset to ROMMON. During the bootup, once the chassis level role is resolved, the ICS downloads the image from the in-chassis active (ICA) supervisor engine. Once the supervisor engine is booted with the image, it will function in the same way as a DFC line card. All applications running in virtual switch (VS) view the in-chassis standby as a DFC line card.

See Figure 4-6 for the various roles that supervisor engines can assume within a quad-supervisor VSS system.

Figure 4-5 Typical VSS Quad-Supervisor Configuration



If your supervisor engine is:

- in-chassis active, it can be VSS active or VSS standby.
- in-chassis standby, it can only be an ICS.
- VSS active, it can only be ICA.
- VSS standby, it can only be ICA.

Quad-supervisor uplink forwarding provides these key features:

- eFSU upgrades— You can upgrade or downgrade your VSS system using ISSU. See "Upgrading a VSS" section on page 4-54 for more information about eFSU upgrades.
- Image version mismatch—Before the bootup, the ICS completes a version check. If there is a version mismatch, the ICS is set to ROMMON. If you want to boot different images on the ICS and ICA. You need to configure the **no switch virtual in-chassis standby bootup version mismatch-check** command. This command is only valid once all four supervisors are running software that supports Quad-supervisor uplink forwarding. If one supervisor is running software that does not support Quad-supervisor uplink forwarding the command will have no effect.
- EARL mode mismatch—If the supervisor engine EARL modes do not match then the supervisor engine is reset to ROMMON. It is recommended that all four supervisor engines run the same EARL Lite or EARL Heavy version.
- VSS RPR switchover—On RPR switchover the ICS will be reset. For more information regarding RPR see "RPR and SSO Redundancy" section on page 4-12.

- In-chassis RPR switchover—ICS supervisor engines in the supervisor engine 1 and supervisor engine 2 positions boot up as RPR-Warm. RPR-Warm is when a supervisor engine acts as a DFC. When a VSS stateful switchover occurs, the supervisor engine is reset to ROMMON and boot ups with the supervisor engine image. You can verify the switchover mode of the supervisor engines by entering the **show module** command.
- VSS stateful switchover—When the in-chassis active supervisor engine crashes, a switchover occurs and the whole chassis reloads (including the ICS) during which the standby supervisor engine takes over as the in-chassis active supervisor engine. A z-switchover operates exactly like a switchover except that the ICS supervisor engine takes priority and is assigned the in-chassis standby supervisor engine. You can initiate a z-switchover by entering the **redundancy force switchover** command on the in-chassis active supervisor engine. You can verify the switchover mode of the supervisor engines by entering the **show module** command.

If you insert a supervisor engine from another system (VS or standalone) in the supervisor engine 1 or supervisor engine 2 position of your existing two supervisor engine VSS system, the supervisor engine does a reset to update the supervisor engine number, and then reboots before going online as a DFC.

Interface Naming Convention

In VSS mode, interfaces are specified using the switch number (in addition to slot and port), because the same slot numbers are used on both chassis. For example, the **interface 1/5/4** command specifies port 4 of the switching module in slot 5 of switch 1. The **interface 2/5/4** command specifies port 4 on the switching module in slot 5 of switch 2.

Software Features

With some exceptions, the VSS has feature parity with the standalone Catalyst 6500 series switch. Major exceptions include:

- In software releases earlier than Cisco IOS Release 12.2(33)SXI2, the VSS does not support IPv6 unicast or MPLS.
- In software releases earlier than Cisco IOS Release 12.2(33)SXI, port-based QoS and port ACLs (PACLs) are supported only on Layer 2 single-chassis or multichassis EtherChannel (MEC) links. Beginning with Cisco IOS Release 12.2(33)SXI, port-based QoS and PACLs can be applied to any physical port in the VSS, excluding ports in the VSL. PACLs can be applied to no more than 2046 ports in the VSS.
- In software releases earlier than Cisco IOS Release 12.2(33)SXI4, the VSS does not support supervisor engine redundancy within a chassis.
- Starting in Cisco IOS Release 12.2(33)SXI4, the VSS does support supervisor engine redundancy within a chassis.
- In releases earlier than Release 12.2(33) SXH2, the VSS feature and the lawful intercept feature cannot be configured together. (CSCsl77715)

Hardware Requirements

The following sections describe the hardware requirements of a VSS:

- Chassis and Modules, page 4-9
- VSL Hardware Requirements, page 4-9
- PFC, DFC, and CFC Requirements, page 4-10

Chassis and Modules

Table 4-1 describes the hardware requirements for the VSS chassis and modules.

Table 4-1 VSS Hardware Requirements

Hardware	Count	Requirements
Chassis	2	The VSS is available on chassis that support VS-S720-10G supervisor engines and WS-X6708-10G switching modules. **Note** The two chassis need not be identical.
Supervisor Engines	2	The VSS requires Supervisor Engine 720 with 10-Gigabit Ethernet ports. You must use either two VS-S720-10G-3C or two VS-S720-10G-3CXL supervisor engine modules. The two supervisor engines must match exactly.
Switching Modules	2+	The VSS requires 67xx series switching modules. The VSS does not support classic, CEF256, or dCEF256 switching modules. In virtual switch mode, unsupported switching modules remain powered off.

VSL Hardware Requirements

The VSL EtherChannel supports only 10-Gigabit Ethernet ports. The 10-Gigabit Ethernet port can be located on the supervisor engine module or on one of the following switching modules:

- WS-X6708-10G-3C or WS-X6708-10G-3CXL
- WS-X6716-10G-3C or WS-X6716-10G-3CXL
- WS-X6716-10T-3C or WS-X6716-10T-3CXL

Note

- Using the 10-Gigabit Ethernet ports on a WS-X6716-10G switching module in the VSL EtherChannel requires Cisco IOS Release 12.2(33)SXI or a later release.
- Using the 10-Gigabit Ethernet ports on a WS-X6716-10T switching module in the VSL EtherChannel requires Cisco IOS Release 12.2(33)SXI4 or a later release.

We recommend that you use both of the 10-Gigabit Ethernet ports on the supervisor engines to create the VSL between the two chassis.

You can add additional physical links to the VSL EtherChannel by using the 10-Gigabit Ethernet ports on WS-X6708-10G, WS-X6716-10G, or WS-X6716-10T switching modules.

Note

- When using the 10-Gigabit Ethernet ports on the WS-X6716-10G or WS-X6716-10T switching module as VSL links, you must operate the ports in performance, not oversubscription, mode. If you enter the **no hw-module switch x slot y oversubscription** command to configure non-oversubscription mode (performance mode), then only ports 1, 5, 9, and 13 are configurable; the other ports on the module are disabled.
- Port-groups are independent of each other and one, or more, port-groups can operate in non-oversubscribed (1:1) mode (e.g. for VSL) with the 3 unused ports administratively shutdown, while the others can still operate in oversubscribed (4:1) mode.

PFC, DFC, and CFC Requirements

The VSS supports any 67xx series switching module with CFC hardware.

The VSS supports DFC3C or DFC3CXL hardware and does not support DFC3A/3B/3BXL hardware.

If any switching module in the VSS is provisioned with DFC3C, the whole VSS must operate in PFC3C mode. If a 67xx series switching module with a DFC3A/3B/3BXL is inserted in the chassis of a VSS, the module will remain unpowered, because VSS supports only DFC3C and DFC3CXL.

If the supervisor engines are provisioned with PFC3C, the VSS will automatically operate in 3C mode, even if some of the modules are 3CXL. However, if the supervisor engines are provisioned with PFC3CXL, but some of the modules are 3C, you need to configure the VSS to operate in 3C mode. The **platform hardware vsl pfc mode pfc3c** configuration command sets the system to operate in 3C mode after the next restart. See the "SSO Dependencies" section on page 4-25 for further details about this command.

Multichassis EtherChannel Requirements

Physical links from any 67xx series switching module can be used to implement a Multichassis EtherChannel (MEC).

Service Module Support

VSS mode supports these service modules:

- Network Analysis Modules (NAM):
  - WS-SVC-NAM-1
  - WS-SVC-NAM-2
- Application Control Engines (ACE):
  - ACE10-6500-K9
  - ACE20-MOD-K9
- Intrusion Detection System Services Module (IDSM): WS-SVC-IDSM2-K9
- Wireless Services Module (WiSM): WS-SVC-WISM-1-K9
- Firewall Services Module (FWSM): WS-SVC-FWM-1-K9

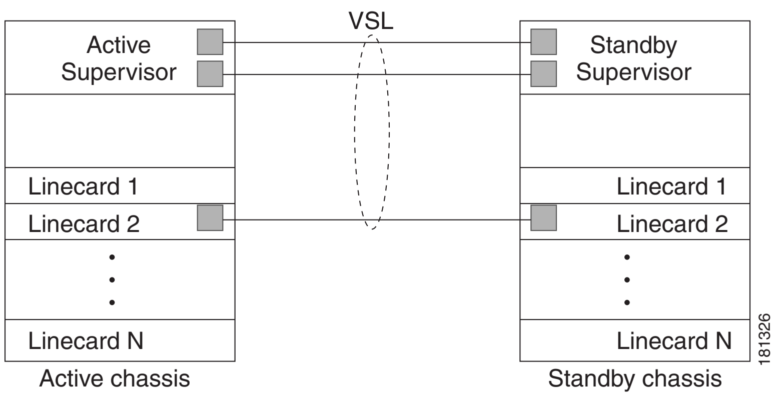
Note Before deploying a service module in VSS mode, check the service module release notes and if necessary, upgrade the service module software.

Understanding VSL Topology

A VSS contains two chassis that communicate using the VSL, which is a special port group.

We recommend that you configure both of the 10-Gigabit Ethernet ports on the supervisor engines as VSL ports. Optionally, you can also configure the VSL port group to contain switching module 10-Gigabit Ethernet ports. This configuration provides additional VSL capacity. See Figure 4-6 for an example topology.

Figure 4-6 VSL Topology Example



VSS Redundancy

The following sections describe how redundancy in a VSS supports network high availability:

- Overview, page 4-11
- RPR and SSO Redundancy, page 4-12
- Failed Chassis Recovery, page 4-13
- VSL Failure, page 4-13
- User Actions, page 4-14

Overview

A VSS operates stateful switchover (SSO) between the VSS active and VSS standby supervisor engines. Compared to standalone mode, a VSS has the following important differences in its redundancy model:

- The VSS active and VSS standby supervisor engines are hosted in separate chassis and use the VSL to exchange information.
- The VSS active supervisor engine controls both chassis of the VSS. The VSS active supervisor engine runs the Layer 2 and Layer 3 control protocols and manages the switching modules on both chassis.
- The VSS active and VSS standby chassis both perform data traffic forwarding.

If the VSS active supervisor engine fails, the VSS standby supervisor engine initiates a switchover and assumes the VSS active role.

RPR and SSO Redundancy

A VSS operates with stateful switchover (SSO) redundancy if it meets the following requirements:

- Both supervisor engines must be running the same software version.
- VSL-related configuration in the two chassis must match.
- PFC mode must match.
- SSO and nonstop forwarding (NSF) must be configured on each chassis.

See the “SSO Dependencies” section on page 4-25 for additional details about the requirements for SSO redundancy on a VSS. See Chapter 6, “Configuring NSF with SSO Supervisor Engine Redundancy” for information about configuring SSO and NSF.

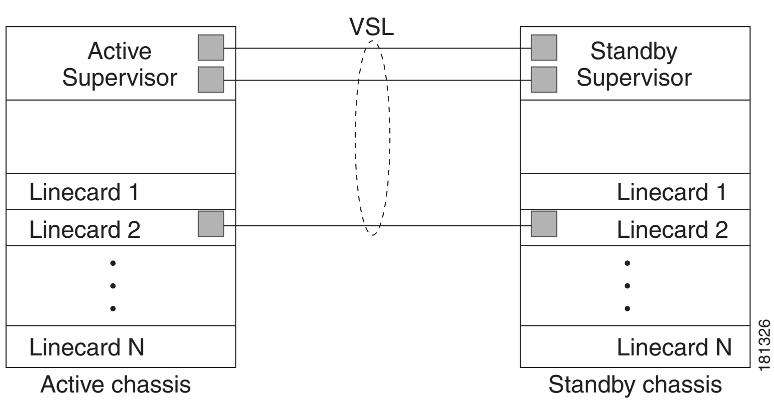
With SSO redundancy, the VSS standby supervisor engine is always ready to assume control following a fault on the VSS active supervisor engine. Configuration, forwarding, and state information are synchronized from the VSS active supervisor engine to the redundant supervisor engine at startup and whenever changes to the VSS active supervisor engine configuration occur. If a switchover occurs, traffic disruption is minimized.

If a VSS does not meet the requirements for SSO redundancy, the VSS will use route processor redundancy (RPR). In RPR mode, the VSS active supervisor engine does not synchronize configuration changes or state information with the VSS standby. The VSS standby supervisor engine is only partially initialized and the switching modules on the VSS standby supervisor are not powered up. If a switchover occurs, the VSS standby supervisor engine completes its initialization and powers up the switching modules. Traffic is disrupted for the normal reboot time of the chassis.

The VSS normally runs stateful switchover (SSO) between the VSS active and VSS standby supervisor engines (see Figure 4-7). The VSS determines the role of each supervisor engine during initialization.

The supervisor engine in the VSS standby chassis runs in hot standby state. The VSS uses the VSL link to synchronize configuration data from the VSS active to the VSS standby supervisor engine. Also, protocols and features that support high availability synchronize their events and state information to the VSS standby supervisor engine.

Figure 4-7 Chassis Roles in a VSS



Failed Chassis Recovery

If the VSS active chassis or supervisor engine fails, the VSS initiates a stateful switchover (SSO) and the former VSS standby supervisor engine assumes the VSS active role. The failed chassis performs recovery action by reloading the supervisor engine.

If the VSS standby chassis or supervisor engine fails, no switchover is required. The failed chassis performs recovery action by reloading the supervisor engine.

The VSL links are unavailable while the failed chassis recovers. After the chassis reloads, it becomes the new VSS standby chassis and the VSS reinitializes the VSL links between the two chassis.

The switching modules on the failed chassis are unavailable during recovery, so the VSS operates only with the MEC links that terminate on the VSS active chassis. The bandwidth of the VSS is reduced until the failed chassis has completed its recovery and become operational again. Any devices that are connected only to the failed chassis experience an outage.

Note The VSS may experience a brief data path disruption when the switching modules in the VSS standby chassis become operational after the SSO.

After the SSO, much of the processing power of the VSS active supervisor engine is consumed in bringing up a large number of ports simultaneously in the VSS standby chassis. As a result, some links might be brought up before the supervisor engine has configured forwarding for the links, causing traffic to those links to be lost until the configuration is complete. This condition is especially disruptive if the link is an MEC link. Two methods are available to reduce data disruption following an SSO:

- Beginning in Cisco IOS Release 12.2(33)SXH2, you can configure the VSS to activate non-VSL ports in smaller groups over a period of time rather than all ports simultaneously. For information about deferring activation of the ports, see the "Configuring Deferred Port Activation During VSS Standby Recovery" section on page 4-44.
- You can defer the load sharing of the peer switch's MEC member ports during reestablishment of the port connections. See the "Failed Chassis MEC Recovery" section on page 4-16 for details about load share deferral.

VSL Failure

To ensure fast recovery from VSL failures, fast link notification is enabled in virtual switch mode on all port channel members (including VSL ports) whose hardware supports fast link notification.

Note Fast link notification is not compatible with link debounce mechanisms. In virtual switch mode, link debounce is disabled on all port channel members.

If a single VSL physical link goes down, the VSS adjusts the port group so that the failed link is not selected.

If the VSS standby chassis detects complete VSL link failure, it initiates a stateful switchover (SSO). If the VSS active chassis has failed (causing the VSL links to go down), the scenario is chassis failure, as described in the previous section.

If only the VSL has failed and the VSS active chassis is still operational, this is a dual-active scenario. The VSS detects that both chassis are operating in VSS active mode and performs recovery action. See the "Dual-Active Detection" section on page 4-22 for additional details about the dual-active scenario.

User Actions

From the VSS active chassis command console, you can initiate a VSS switchover or a reload.

If you enter the **reload** command from the command console, the entire VSS performs a reload.

To reload only the VSS standby chassis, use **redundancy reload peer** command.

To force a switchover from the VSS active to the VSS standby supervisor engine, use the **redundancy force-switchover** command.

To reset the VSS standby supervisor engine or to reset both the VSS active and VSS standby supervisor engines, use the **redundancy reload shelf** command.

Multichassis EtherChannels

These sections describe multichassis EtherChannels (MECs):

- Overview, page 4-14
- MEC Failure Scenarios, page 4-15

Overview

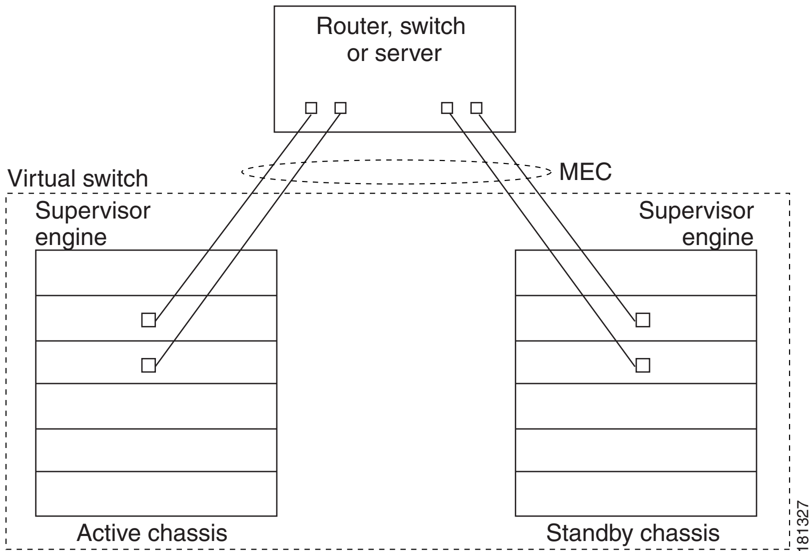
A multichassis EtherChannel is an EtherChannel with ports that terminate on both chassis of the VSS (see Figure 4-8). A VSS MEC can connect to any network element that supports EtherChannel (such as a host, server, router, or switch).

At the VSS, an MEC is an EtherChannel with additional capability: the VSS balances the load across ports in each chassis independently. For example, if traffic enters the VSS active chassis, the VSS will select an MEC link from the VSS active chassis. This MEC capability ensures that data traffic does not unnecessarily traverse the VSL.

Each MEC can optionally be configured to support either PAgP or LACP. These protocols run only on the VSS active chassis. PAgP or LACP control packets destined for an MEC link on the VSS standby chassis are sent across VSL.

An MEC can support up to eight VSS active physical links, which can be distributed in any proportion between the VSS active and VSS standby chassis.

Figure 4-8 MEC Topology



MEC Failure Scenarios

We recommend that you configure the MEC with at least one link to each chassis. This configuration conserves VSL bandwidth (traffic egress link is on the same chassis as the ingress link), and increases network reliability (if one VSS supervisor engine fails, the MEC is still operational).

The following sections describe possible failures and the resulting impacts:

- Single MEC Link Failure, page 4-15
- All MEC Links to the VSS Active Chassis Fail, page 4-15
- All MEC Links to the VSS Standby Chassis Fail, page 4-16
- All MEC Links Fail, page 4-16
- VSS Standby Chassis Failure, page 4-16
- VSS Active Chassis Failure, page 4-16
- Failed Chassis MEC Recovery, page 4-16

Single MEC Link Failure

If a link within the MEC fails (and other links in the MEC are still operational), the MEC redistributes the load among the operational links, as in a regular port.

All MEC Links to the VSS Active Chassis Fail

If all links to the VSS active chassis fail, the MEC becomes a regular EtherChannel with operational links to the VSS standby chassis.

Data traffic terminating on the VSS active chassis reaches the MEC by crossing the VSL to the VSS standby chassis. Control protocols continue to run in the VSS active chassis. Protocol messages reach the MEC by crossing the VSL.

All MEC Links to the VSS Standby Chassis Fail

If all links fail to the VSS standby chassis, the MEC becomes a regular EtherChannel with operational links to the VSS active chassis.

Control protocols continue to run in the VSS active chassis. All control and data traffic from the VSS standby chassis reaches the MEC by crossing the VSL to the VSS active chassis.

All MEC Links Fail

If all links in an MEC fail, the logical interface for the EtherChannel is set to unavailable. Layer 2 control protocols perform the same corrective action as for a link-down event on a regular EtherChannel.

On adjacent switches, routing protocols and Spanning Tree Protocol (STP) perform the same corrective action as for a regular EtherChannel.

VSS Standby Chassis Failure

If the VSS standby chassis fails, the MEC becomes a regular EtherChannel with operational links on the VSS active chassis. Connected peer switches detect the link failures, and adjust their load-balancing algorithms to use only the links to the VSS active chassis.

VSS Active Chassis Failure

VSS active chassis failure results in a stateful switchover (SSO). See the "VSS Redundancy" section on page 4-11 for details about SSO on a VSS. After the switchover, the MEC is operational on the new VSS active chassis. Connected peer switches detect the link failures (to the failed chassis), and adjust their load-balancing algorithms to use only the links to the new VSS active chassis.

Failed Chassis MEC Recovery

When a failed chassis returns to service as the new VSS standby chassis, protocol messages reestablish the MEC links between the recovered chassis and connected peer switches.

Although the recovered chassis' MEC links are immediately ready to receive unicast traffic from the peer switch, received multicast traffic may be lost for a period of several seconds to several minutes. To reduce this loss, you can configure the port load share deferral feature on MEC port channels of the peer switch. When load share deferral is configured, the peer's deferred MEC port channels will establish with an initial load share of 0. During the configured deferral interval, the peer's deferred port channels are capable of receiving data and control traffic, and of sending control traffic, but are unable to forward data traffic to the VSS. See the "Configuring Port Load Share Deferral on the Peer Switch" section on page 4-45 for details about configuring port load share deferral.

Packet Handling

In a VSS, the VSS active supervisor engine runs the Layer 2 and Layer 3 protocols and features for the VSS and manages the DFC modules for both chassis.

The VSS uses the VSL to communicate system and protocol information between the peer chassis and to carry data traffic between the two chassis.

Both chassis perform packet forwarding for ingress traffic on their local interfaces. The VSS minimizes the amount of data traffic that must traverse the VSL.

The following sections describe packet handling in a VSS:

- Traffic on the VSL, page 4-17
- Layer 2 Protocols, page 4-17
- Layer 3 Protocols, page 4-18
- SPAN, page 4-20

Traffic on the VSL

The VSL carries data traffic and in-band control traffic between the two chassis. All frames forwarded over the VSL link are encapsulated with a special 32-byte header, which provides information for the VSS to forward the packet on the peer chassis.

The VSL transports control messages between the two chassis. Messages include protocol messages that are processed by the VSS active supervisor engine, but received or transmitted by interfaces on the VSS standby chassis. Control traffic also includes module programming between the VSS active supervisor engine and switching modules on the VSS standby chassis.

The VSS needs to transmit data traffic over the VSL under the following circumstances:

- Layer 2 traffic flooded over a VLAN (even for dual-homed links).
- Packets processed by software on the VSS active supervisor engine where the ingress interface is on the VSS standby chassis.
- The packet destination is on the peer chassis, such as the following examples:
  - Traffic within a VLAN where the known destination interface is on the peer chassis.
  - Traffic that is replicated for a multicast group and the multicast receivers are on the peer chassis.
  - The known unicast destination MAC address is on the peer chassis.
  - The packet is a MAC notification frame destined for a port on the peer chassis.

VSL also transports system data, such as NetFlow export data and SNMP data, from the VSS standby chassis to the VSS active supervisor engine.

To preserve the VSL bandwidth for critical functions, the VSS uses strategies to minimize user data traffic that must traverse the VSL. For example, if an access switch is dual-homed (attached with an MEC terminating on both VSS chassis), the VSS transmits packets to the access switch using a link on the same chassis as the ingress link.

Traffic on the VSL is load-balanced with the same global hashing algorithms available for EtherChannels (the default algorithm is source-destination IP).

Layer 2 Protocols

The VSS active supervisor engine runs the Layer 2 protocols (such as STP and VTP) for the switching modules on both chassis. Protocol messages that are transmitted and received on the VSS standby chassis switching modules must traverse the VSL to reach the VSS active supervisor engine.

The following sections describe Layer 2 protocols for a VSS:

- Spanning Tree Protocol, page 4-18
- Virtual Trunk Protocol, page 4-18
- EtherChannel Control Protocols, page 4-18
- Multicast Protocols, page 4-18

Spanning Tree Protocol

The VSS active chassis runs Spanning Tree Protocol (STP). The VSS standby chassis redirects STP BPDUs across the VSL to the VSS active chassis.

The STP bridge ID is commonly derived from the chassis MAC address. To ensure that the bridge ID does not change after a switchover, the VSS continues to use the original chassis MAC address for the STP Bridge ID.

Virtual Trunk Protocol

Virtual Trunk Protocol (VTP) uses the IP address of the switch and local current time for version control in advertisements. After a switchover, VTP uses the IP address of the newly VSS active chassis.

EtherChannel Control Protocols

Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP) packets contain a device identifier. The VSS defines a common device identifier for both chassis to use.

A new PAgP enhancement has been defined for assisting with dual-active scenario detection. For additional information, see the "Dual-Active Detection" section on page 4-22.

Multicast Protocols

In Release 12.2(33)SXI4 and later releases, fast-redirect optimization makes multicast traffic redirection between inter-chassis or intra-chassis line cards faster for Layer 2 trunk multichassis EtherChannel or distributed EtherChannel in case of member port link failure and recovery. This operation occurs mainly when a member port link goes down (port leaves the EtherChannel) and when the member port link goes up (port joins or rejoins the EtherChannel). Fast-redirect does not take effect when you add or remove a member port due to a configuration change or during system boot up.

Layer 3 Protocols

The MSFC on the VSS active supervisor engine runs the Layer 3 protocols and features for the VSS. Both chassis perform packet forwarding for ingress traffic on their interfaces. If possible, ingress traffic is forwarded to an outgoing interface on the same chassis, to minimize data traffic that must traverse the VSL.

Because the VSS standby chassis is actively forwarding traffic, the VSS active supervisor engine distributes updates to the VSS standby supervisor engine PFC and all VSS standby chassis DFCs.

The following sections describe Layer 3 protocols for a VSS:

IPv4

The supervisor engine on the VSS active chassis runs the IPv4 routing protocols and performs any required software forwarding.

Routing updates received on the VSS standby chassis are redirected to the VSS active chassis across the VSL.

Hardware forwarding is distributed across all DFCs on the VSS. The supervisor engine on the VSS active chassis sends FIB updates to all local DFCs, remote DFCs, and the VSS standby supervisor engine PFC.

All hardware routing uses the router MAC address assigned by the VSS active supervisor engine. After a switchover, the original MAC address is still used.

The supervisor engine on the VSS active chassis performs all software forwarding (for protocols such as IPX) and feature processing (such as fragmentation and TTL exceed). If a switchover occurs, software forwarding is disrupted until the new VSS active supervisor engine obtains the latest CEF and other forwarding information.

In virtual switch mode, the requirements to support non-stop forwarding (NSF) are the same as in standalone mode. For additional information about NSF requirements, refer to the *Catalyst 6500 Series Switch Cisco IOS Configuration Guide*, Release 12.2SX.

From a routing peer perspective, EtherChannels remain operational during a switchover (only the links to the failed chassis are down, so the routing adjacencies remains valid).

The VSS implements path filtering by storing only local paths (paths that do not traverse the VSL) in the FIB entries. Therefore, IP forwarding performs load sharing among the local paths. If no local paths to a given destination are available, the VSS updates the FIB entry to include remote paths (reachable by traversing the VSL).

IPv6 and MPLS

In Cisco IOS Release 12.2(33)SXI2 and later releases, the VSS supports IPv6 unicast and MPLS.

IPv4 Multicast

The IPv4 multicast protocols run on the VSS active supervisor engine. Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM) protocol packets received on the VSS standby supervisor engine are transmitted across VSL to the VSS active chassis.

The VSS active supervisor engine sends IGMP and PIM protocol packets to the VSS standby supervisor engine in order to maintain Layer 2 information for stateful switchover (SSO).

The VSS active supervisor engine distributes multicast FIB and adjacency table updates to the VSS standby supervisor engine and switching module DFCs.

For Layer 3 multicast in the VSS, learned multicast routes are stored in hardware in the VSS standby supervisor engine. After a switchover, multicast forwarding continues, using the existing hardware entries.

Note To avoid multicast route changes as a result of the switchover, we recommend that all links carrying multicast traffic be configured as MEC rather than Equal Cost Multipath (ECMP).

In virtual switch mode, the VSS active chassis does not program the multicast expansion table (MET) on the VSS standby chassis. The VSS standby supervisor engine programs the outgoing interface hardware entries for all local multicast receivers

If all switching modules on the VSS active chassis and VSS standby chassis are egress capable, the multicast replication mode is set to egress mode; otherwise, the mode is set to ingress mode.

In egress mode, replication is distributed to DFCs that have ports in outgoing VLANs for a particular flow. In ingress mode, replication for all outgoing VLANs is done on the ingress DFC.

For packets traversing VSL, all Layer 3 multicast replication occurs on the ingress chassis. If there are multiple receivers on the egress chassis, replicated packets are forwarded over the VSL.

Software Features

Software features run only on the VSS active supervisor engine. Incoming packets to the VSS standby chassis that require software processing are sent across the VSL.

For features supported in hardware, the ACL configuration is sent to the TCAM manager on the VSS active supervisor engine, the VSS standby supervisor engine, and all DFCs.

SPAN

The VSS supports all SPAN features for non-VSL interfaces. The VSS supports SPAN features on VSL interfaces with the following limitations:

- If the VSL is configured as a local SPAN source, the SPAN destination interface must be on the same chassis as the source interface.
- VSL cannot be configured as a SPAN destination.
- VSL cannot be configured as a traffic source of RSPAN, ERSPAN, or egress-only SPAN.

The number of SPAN sessions available to a VSS is the same as for a single chassis running in standalone mode.

System Monitoring

The following sections describe system monitoring and system management for a VSS:

- Power Management, page 4-20
- Environmental Monitoring, page 4-20
- File System Access, page 4-21
- VSL Diagnostics, page 4-21
- Service Modules, page 4-21
- Network Management, page 4-22

Power Management

From the VSS active chassis, you can control power-related functions for the VSS standby chassis. For example, use the **power enable switch** command to control power to the modules and slots on the VSS standby chassis. Use the **show power switch** command to see the current power settings and status.

Environmental Monitoring

Environmental monitoring runs on both supervisor engines. The VSS standby chassis reports notifications to the VSS active supervisor engine. The VSS active chassis gathers log messages for both chassis. The VSS active chassis synchronizes the calendar and system clock to the VSS standby chassis.

File System Access

You can access file systems of both chassis from the VSS active chassis. Prefix the device name with the switch number and slot number to access directories on the VSS standby chassis. For example, the command **dir sw2-slot6-disk0:** lists the contents of disk0 on the VSS standby chassis (assuming switch 2 is the VSS standby chassis). You can access the VSS standby chassis file system only when VSL is operational.

VSL Diagnostics

You can use the **diagnostic schedule** and **diagnostic start** commands on a VSS. In virtual switch mode, these commands require an additional parameter, which specifies the chassis to apply the command.

When you configure a VSL port on a switching module or a supervisor engine module, the diagnostics suite incorporates additional tests for the VSL ports.

Use the **show diagnostic content** command to display the diagnostics test suite for a module.

The following VSL-specific diagnostics tests are available on WS-X6708-10G switching modules with VSL ports. These tests are disruptive:

- TestVslBridgeLink
- TestVslLocalLoopback

The following VSL-specific diagnostics tests are available on a Supervisor Engine 720-10GE with VSL ports. These tests are disruptive:

- TestVSActiveToStandbyLoopback
- TestVslBridgeLink
- TestVslLocalLoopback

The following VSL-specific diagnostics test is available for VSL ports on the switching module or the supervisor engine. This test is not disruptive:

- TestVslStatus

See the "ViSN Tests" section on page B-47.

Service Modules

The following system monitoring and system management guidelines apply to service modules supported by the VSS:

- The supervisor engine in the same chassis as the service module controls the powering up of the service module. After the service module is online, you initiate a session from the VSS active supervisor engine to configure and maintain the service module.
- Use the **session** command to connect to the service module. If the service module is in the VSS standby chassis, the session runs over the VSL.
- The VSS active chassis performs the graceful shutdown of the service module, even if the service module is in the VSS standby chassis.

Network Management

The following sections describe network management for a VSS:

- Telnet over SSH Sessions and the Web Browser User Interface, page 4-22
- SNMP, page 4-22
- Command Console, page 4-22

Telnet over SSH Sessions and the Web Browser User Interface

A VSS supports remote access using Telnet over SSH sessions and the Cisco web browser user interface.

All remote access is directed to the VSS active supervisor engine, which manages the whole VSS.

If the VSS performs a switchover, Telnet over SSH sessions and web browser sessions are disconnected.

SNMP

The SNMP agent runs on the VSS active supervisor engine.

CISCO-VIRTUAL-SWITCH-MIB is a new MIB for virtual switch mode and contains the following main components:

- cvsGlobalObjects — Domain #, Switch #, Switch Mode
- cvsCoreSwitchConfig — Switch Priority
- cvsChassisTable — Chassis Role and Uptime
- cvsVSLConnectionTable — VSL Port Count, Operational State
- cvsVSLStatsTable — Total Packets, Total Error Packets
- cvsVSLPortStatsTable — TX/RX Good, Bad, Bi-dir and Uni-dir Packets

Command Console

Connect console cables to both supervisor engine console ports. You can only use configuration mode in the console for the VSS active supervisor engine.

The console on the VSS standby chassis will indicate that chassis is operating in VSS standby mode by adding the characters "-stdby" to the command line prompt. You cannot enter configuration mode on the VSS standby chassis console.

The following example shows the prompt on the VSS standby console:

```
Router-stdby> show switch virtual

Switch mode                  : Virtual Switch
Virtual switch domain number : 100
Local switch number          : 1
Local switch operational role: Virtual Switch Standby
Peer switch number           : 2
Peer switch operational role : Virtual Switch Active
```

Dual-Active Detection

If the VSL fails, the VSS standby chassis cannot determine the state of the VSS active chassis. To ensure that switchover occurs without delay, the VSS standby chassis assumes the VSS active chassis has failed and initiates switchover to take over the VSS active role.

If the original VSS active chassis is still operational, both chassis are now VSS active. This situation is called a *dual-active scenario*. A dual-active scenario can have adverse affects on network stability, because both chassis use the same IP addresses, SSH keys, and STP bridge ID. The VSS must detect a dual-active scenario and take recovery action.

The VSS supports these three methods for detecting a dual-active scenario:

- Enhanced PAgP—Uses PAgP messaging over the MEC links to communicate between the two chassis through a neighbor switch. Enhanced PAgP is faster than IP BFD, but requires a neighbor switch that supports the PAgP enhancements.
- IP Bidirectional Forwarding Detection (BFD)—Uses BFD messaging over a backup Ethernet connection. IP BFD uses a direct connection between the two chassis and does not require support from a neighbor switch.
- dual-active fast-hello—Uses special hello messages over a backup Ethernet connection. Dual-active fast-hello is faster than IP BFD and does not require support from a neighbor switch. This method is available only in Cisco IOS Release 12.2(33)SXI and later releases,

You can configure all three detection methods to be VSS active at the same time.

For line redundancy, we recommend dedicating at least two ports per switch for dual-active detection. For module redundancy, the two ports can be on different switching modules in each chassis, and should be on different modules than the VSL links, if feasible.

The dual-active detection and recovery methods are described in the following sections:

- Dual-Active Detection Using Enhanced PAgP, page 4-23
- Dual-Active Detection Using IP BFD, page 4-24
- Dual-Active Detection Using Dual-Active Fast Hello Packets, page 4-24
- Recovery Actions, page 4-24

Dual-Active Detection Using Enhanced PAgP

If a VSS MEC terminates on a Cisco switch, you can run the port aggregation protocol (PAgP) on the MEC. If enhanced PAgP is running on an MEC between the VSS and another switch running Release 12.2(33)SXH1 or a later release, the VSS can use enhanced PAgP to detect a dual-active scenario.

The MEC must have at least one port on each chassis of the VSS. In VSS mode, PAgP messages include a new type length value (TLV) that contains the ID of the VSS active switch. Only switches in VSS mode send the new TLV.

When the VSS standby chassis detects VSL failure, it initiates SSO and becomes VSS active. Subsequent PAgP messages to the connected switch from the newly VSS active chassis contain the new VSS active ID. The connected switch sends PAgP messages with the new VSS active ID to both VSS chassis.

If the formerly VSS active chassis is still operational, it detects the dual-active scenario because the VSS active ID in the PAgP messages changes. This chassis initiates recovery actions as described in the "Recovery Actions" section on page 4-24.

Dual-Active Detection Using IP BFD

To use the IP BFD detection method, you must provision a direct Ethernet connection between the two switches. Regular Layer 3 ping will not function correctly on this connection, as both chassis have the same IP address. The VSS instead uses the Bidirectional Forwarding Detection (BFD) protocol.

If the VSL fails, both chassis create BFD neighbors, and try to establish adjacency. If the original VSS active chassis receives an adjacency message, it realizes that this is a dual-active scenario, and initiates recovery actions as described in the "Recovery Actions" section on page 4-24.

Note If Flex Links are configured on the VSS, we recommend using the PAgP detection method. Do not configure Flex Links and BFD dual-active detection on the same VSS.

Dual-Active Detection Using Dual-Active Fast Hello Packets

Cisco IOS Release 12.2(33)SXI and later releases support the dual-active fast hello method.

To use the dual-active fast hello packet detection method, you must provision a direct Ethernet connection between the two VSS chassis. You can dedicate up to four non-VSL links for this purpose.

The two chassis periodically exchange special Layer 2 dual-active hello messages containing information about the switch state. If the VSL fails and a dual-active scenario occurs, each switch recognizes from the peer's messages that there is a dual-active scenario and initiates recovery actions as described in the "Recovery Actions" section on page 4-24. If a switch does not receive an expected dual-active fast hello message from the peer before the timer expires, the switch assumes that the link is no longer capable of dual-active detection.

Recovery Actions

An VSS active chassis that detects a dual-active condition shuts down all of its non-VSL interfaces (except interfaces configured to be excluded from shutdown) to remove itself from the network, and waits in recovery mode until the VSL links have recovered. You might need to physically repair the VSL failure. When the shut down chassis detects that VSL is operational again, the chassis reloads and returns to service as the VSS standby chassis.

Loopback interfaces are also shut down in recovery mode. Do not configure loopback interfaces while in recovery mode, because any new loopback interfaces configured in recovery mode will not be shut down.

Note If the running configuration of the chassis in recovery mode has been changed without saving, the chassis will not automatically reload. In this situation, you must save the running configuration and then reload manually.

VSS Initialization

A VSS is formed when the two chassis and the VSL link between them become operational. The peer chassis communicate over the VSL to negotiate the chassis roles.

If only one chassis becomes operational, it assumes the VSS active role. The VSS forms when the second chassis becomes operational and both chassis bring up their VSL interfaces.

VSS initialization is described in the following sections:

- Virtual Switch Link Protocol, page 4-25
- SSO Dependencies, page 4-25
- Initialization Procedure, page 4-26

Virtual Switch Link Protocol

The Virtual Switch Link Protocol (VSLP) consists of several protocols that contribute to virtual switch initialization. The VSLP includes the following protocols:

- Role Resolution Protocol—The peer chassis use Role Resolution Protocol (RRP) to negotiate the role (VSS active or VSS standby) for each chassis.
- Link Management Protocol—The Link Management Protocol (LMP) runs on all VSL links, and exchanges information required to establish communication between the two chassis. LMP identifies and rejects any unidirectional links. If LMP flags a unidirectional link, the chassis that detects the condition brings the link down and up to restart the VSLP negotiation. VSL moves the control traffic to another port if necessary.

SSO Dependencies

For the VSS to operate with SSO redundancy, the VSS must meet the following conditions:

- Identical software versions—Both supervisor engine modules on the VSS must be running the identical software version.
- VSL configuration consistency—During the startup sequence, the VSS standby chassis sends virtual switch information from the startup-config file to the VSS active chassis. The VSS active chassis ensures that the following information matches correctly on both chassis:
  - Switch virtual domain
  - Switch virtual node
  - Switch priority
  - VSL port channel: switch virtual link identifier
  - VSL ports: channel-group number, shutdown, total number of VSL ports
  - Power redundancy-mode
  - Power enable on VSL modules

  If the VSS detects a mismatch, it prints out an error message on the VSS active chassis console and the VSS standby chassis comes up in RPR mode.

  After you correct the configuration file, save the file by entering the **copy running-config startup-config** command on the VSS active chassis, and then restart the VSS standby chassis.
- PFC mode check—If both supervisor engines are provisioned with PFC3C, the VSS will automatically operate in PFC3C mode, even if some of the switching modules are equipped with DFC3CXLs.

  However, if the supervisor engines are provisioned with PFC3CXL and there is a mixture of DFC3C and DFC3CXL switching modules, the system PFC mode will depend on how the 3C and 3CXL switching modules are distributed between the two chassis.

Each chassis in the VSS determines its system PFC mode. If the supervisor engine of a given chassis is provisioned with PFC3CXL and all the switching modules in the chassis are provisioned with DFC3CXL, the PFC mode for the chassis is PFC3CXL. However, if any of the switching modules is provisioned with DFC3C, the chassis PFC mode will be set to PFC3C. If there is a mismatch between the PFC modes of two chassis, the VSS will come up in RPR mode instead of SSO mode. You can prevent this situation by using the **platform hardware vsl pfc mode pfc3c** command to force the VSS to operate in PFC3C mode after the next reload.

- SSO and NSF enabled

  SSO and NSF must be configured and enabled on both chassis. For detailed information on configuring and verifying SSO and NSF, see Chapter 6, "Configuring NSF with SSO Supervisor Engine Redundancy."

If these conditions are not met, the VSS operates in RPR redundancy mode. For a description of SSO and RPR, see the "VSS Redundancy" section on page 4-11.

Initialization Procedure

The following sections describe the VSS initialization procedure:

VSL Initialization

A VSS is formed when the two chassis and the VSL link between them become operational. Because both chassis need to be assigned their role (VSS active or VSS standby) before completing initialization, VSL is brought online before the rest of the system is initialized. The initialization sequence is as follows:

1. The VSS initializes all cards with VSL ports, and then initializes the VSL ports.
2. The two chassis communicate over VSL to negotiate their roles (VSS active or VSS standby).
3. The VSS active chassis completes the boot sequence, including the consistency check described in the "SSO Dependencies" section on page 4-25.
4. If the consistency check completed successfully, the VSS standby chassis comes up in SSO VSS standby mode. If the consistency check failed, the VSS standby chassis comes up in RPR mode.
5. The VSS active chassis synchronizes configuration and application data to the VSS standby chassis.

System Initialization

If you boot both chassis simultaneously, the VSL ports become VSS active, and the chassis will come up as VSS active and VSS standby. If priority is configured, the higher priority switch becomes active.

If you boot up only one chassis, the VSL ports remain inactive, and the chassis comes up as VSS active. When you subsequently boot up the other chassis, the VSL links become active, and the new chassis comes up as VSS standby.

VSL Down

If the VSL is down when both chassis try to boot up, the situation is similar to a dual-active scenario.

One of the chassis becomes VSS active and the other chassis initiates recovery from the dual-active scenario. For further information, see the "Configuring Dual-Active Detection" section on page 4-45.

VSS Configuration Guidelines and Restrictions

The following sections describe restrictions and guidelines for VSS configuration:

General VSS Restrictions and Guidelines

When configuring the VSS, note the following guidelines and restrictions:

- The VSS configurations in the startup-config file must match on both chassis.
- If you configure a new value for switch priority, the change takes effect only after you save the configuration file and perform a restart.
- Enable the out-of-band MAC address table synchronization among DFC-equipped switching modules by entering the **mac-address-table synchronize** command.

VSL Restrictions and Guidelines

When configuring the VSL, note the following guidelines and restrictions:

- For line redundancy, we recommend configuring at least two ports per switch for the VSL. For module redundancy, the two ports can be on different switching modules in each chassis.
- The **no mls qos channel-consistency** command is automatically applied when you configure the VSL. Do not remove this command.
- VSL ports cannot be Mini Protocol Analyzer sources (the **monitor ... capture** command). Monitor capture sessions cannot be started if a source is the VSL on the port channel of the standby switch. The following message is displayed when a remote VSL port channel on the standby switch is specified and you attempt to start the monitor capture:

  ```
  % remote VSL port is not allowed as capture source
  ```

  The following message is displayed when a scheduled monitor capture start fails because a source is a remote VSL port channel:

  ```
  Packet capture session 1 failed to start. A source port is a remote VSL.
  ```

Multichassis EtherChannel Restrictions and Guidelines

When configuring MECs, note the following guidelines and restrictions:

- All links in an MEC must terminate locally on the VSS active or VSS standby chassis of the same virtual domain.
- For an MEC using the LACP control protocol, the *minlinks* command argument defines the minimum number of physical links in each chassis for the MEC to be operational.
- For an MEC using the LACP control protocol, the *maxbundle* command argument defines the maximum number of links in the MEC across the whole VSS.
- MEC supports LACP 1:1 redundancy. For additional information about LACP 1:1 redundancy, refer to the "Understanding LACP 1:1 Redundancy" section on page 19-5.
- An MEC can be connected to another MEC in a different VSS domain.

Dual-Active Detection Restrictions and Guidelines

When configuring dual-active detection, note the following guidelines and restrictions:

- If Flex Links are configured on the VSS, use PAgP dual-active detection.
- Do not configure Flex Links and BFD dual-active detection on the same VSS.
- For dual-active detection link redundancy, configure at least two ports per switch for dual-active detection. For module redundancy, the two ports can be on different switching modules in each chassis, and should be on different modules than the VSL, if feasible.
- When you configure dual-active fast hello mode, all existing configurations are removed automatically from the interface except for these commands:
  - **description**
  - **logging event**
  - **load-interval**
  - **rcv-queue cos-map**
  - **rcv-queue queue-limit**
  - **rcv-queue random-detect**
  - **rcv-queue threshold**
  - **wrr-queue bandwidth**
  - **wrr-queue cos-map**
  - **wrr-queue queue-limit**
  - **wrr-queue random-detect**
  - **wrr-queue threshold**
  - **priority-queue cos-map**
- Only these configuration commands are available on dual-active detection fast hello ports:
  - **default**
  - **description**
  - **dual-active**

  - **exit**
  - **load-interval**
  - **logging**
  - **no**
  - **shutdown**
- ASIC-specific QoS commands are not configurable on dual-active detection fast hello ports directly, but are allowed to remain on the fast hello port if the commands were configured on another non-fast hello port in that same ASIC group. For a list of these commands, see the "PFC QoS Configuration Guidelines and Restrictions" section on page 43-52.

Service Module Restrictions and Guidelines

When configuring service modules in a VSS, note the following guidelines and restrictions:

- When configuring and attaching VLAN groups to a service module interface in a VSS, use the **switch** {**1** | **2**} command keyword. For example, the **firewall vlan-group** command becomes the **firewall switch** *num* **slot** *slot* **vlan-group** command.
- When upgrading the software image of a service module in a VSS, use the **switch** {**1** | **2**} command keyword.
- EtherChannel load balancing (ECLB) is not supported between an IDSM-2 in the VSS active chassis and an IDSM-2 in the VSS standby chassis.
- A switchover between two service modules in separate chassis of a VSS is considered an intrachassis switchover.

Note For detailed instructions, restrictions, and guidelines for a service module in a VSS, see the configuration guide and command reference for the service module.

Configuring a VSS

These sections describe how to configure a VSS:

Converting to a VSS

By default, the Catalyst 6500 series switch is configured to operate in standalone mode (the switch is a single chassis). The VSS combines two standalone switches into one virtual switch, operating in virtual switch mode.

Note When you convert two standalone switches into one VSS, all non-VSL configuration settings on the VSS standby chassis will revert to the default configuration.

To convert two standalone chassis into a VSS, you perform the following major activities:

- Save the standalone configuration files.
- Configure SSO and NSF on each chassis.
- Configure each chassis as a VSS.
- Convert to a VSS.
- Configure the peer VSL information.

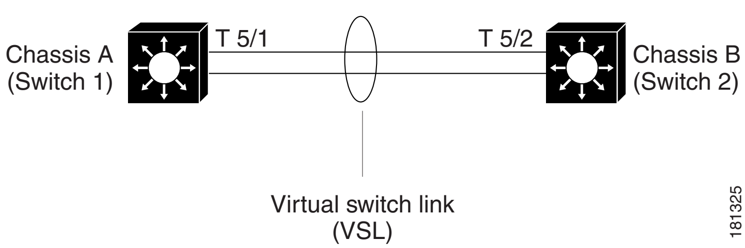
In virtual switch mode, both chassis use the same configuration file. When you make configuration changes on the VSS active chassis, these changes are automatically propagated to the VSS standby chassis.

The tasks required to convert the standalone chassis to a VSS are detailed in the following sections:

- Backing Up the Standalone Configuration, page 4-31
- Configuring SSO and NSF, page 4-31
- Assigning Virtual Switch Domain and Switch Numbers, page 4-32
- Configuring VSL Port Channel and Ports, page 4-33
- Converting the Chassis to Virtual Switch Mode, page 4-34
- (Optional) Configuring VSS Standby Chassis Modules, page 4-35

In the procedures that follow, the example commands assume the configuration shown in Figure 4-9.

Figure 4-9 Example VSS



Two chassis, A and B, are converted into a VSS with virtual switch domain 100. Interface 10-Gigabit Ethernet 5/1 on Switch 1 is connected to interface 10-Gigabit Ethernet 5/2 on Switch 2 to form the VSL.

Backing Up the Standalone Configuration

Save the configuration files for both chassis operating in standalone mode. You need these files to revert to standalone mode from virtual switch mode. On Switch 1, perform this task:

	Command	Purpose
Step 1	`Switch-1#` **`copy running-config startup-config`**	(Optional) Saves the running configuration to startup configuration.
Step 2	`Switch-1#` **`copy startup-config disk0:old-startup-config`**	Copies the startup configuration to a backup file.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	`Switch-2#` **`copy running-config startup-config`**	(Optional) Saves the running configuration to the startup configuration file.
Step 2	`Switch-2#` **`copy startup-config disk0:old-startup-config`**	Copies the startup configuration to a backup file.

Configuring SSO and NSF

SSO and NSF must be configured and enabled on both chassis. On Switch 1, perform this task:

	Command	Purpose
Step 1	`Switch-1(config)#` **`redundancy`**	Enters redundancy configuration mode.
Step 2	`Switch-1(config-red)#` **`mode sso`**	Configures SSO. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO mode.
Step 3	`Switch-1(config-red)#` **`exit`**	Exits redundancy configuration mode.
Step 4	`Switch-1(config)#` **`router`** *`routing_protocol processID`*	Enables routing, which places the router in router configuration mode.
Step 5	`Switch-1(config-router)#` **`nsf`**	Enables NSF operations for the routing protocol.
Step 6	`Switch-1(config-router)#` **`end`**	Exits to privileged EXEC mode.
Step 7	`Switch-1#` **`show running-config`**	Verifies that SSO and NSF are configured and enabled.
Step 8	`Switch-1#` **`show redundancy states`**	Displays the operating redundancy mode.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	`Switch-2(config)#` **`redundancy`**	Enters redundancy configuration mode.
Step 2	`Switch-2(config-red)#` **`mode sso`**	Configures SSO. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO mode.

	Command	Purpose
Step 3	`Switch-2(config-red)#` **`exit`**	Exits redundancy configuration mode.
Step 4	`Switch-2(config)#` **`router`** *`routing_protocol processID`*	Enables routing, which places the router in router configuration mode.
Step 5	`Switch-2(config-router)#` **`nsf`**	Enables NSF operations for the routing protocol.
Step 6	`Switch-2(config-router)#` **`end`**	Exits to privileged EXEC mode.
Step 7	`Switch-2#` **`show running-config`**	Verifies that SSO and NSF are configured and enabled.
Step 8	`Switch-2#` **`show redundancy states`**	Displays the operating redundancy mode.

For detailed information on configuring and verifying SSO and NSF, see Chapter 6, "Configuring NSF with SSO Supervisor Engine Redundancy."

Assigning Virtual Switch Domain and Switch Numbers

You must configure the same virtual switch domain number on both chassis of the VSS. The virtual switch domain is a number between 1 and 255, and must be unique for each VSS in your network (the domain number is incorporated into various identifiers to ensure that these identifiers are unique across the network).

Within the VSS, you must configure one chassis to be switch number 1 and the other chassis to be switch number 2.

To configure the virtual switch domain and switch number on both chassis, perform this task on Switch 1:

	Command	Purpose
Step 1	`Switch-1(config)#` **`switch virtual domain 100`**	Configures the virtual switch domain on Chassis A.
Step 2	`Switch-1(config-vs-domain)#` **`switch 1`**	Configures Chassis A as virtual switch number 1.
Step 3	`Switch-1(config-vs-domain)#` **`exit`**	Exits config-vs-domain.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	`Switch-2(config)#` **`switch virtual domain 100`**	Configures the virtual switch domain on Chassis B.
Step 2	`Switch-2(config-vs-domain)#` **`switch 2`**	Configures Chassis B as virtual switch number 2.
Step 3	`Switch-2(config-vs-domain)#` **`exit`**	Exits config-vs-domain.

Note The switch number is not stored in the startup or running configuration, because both chassis use the same configuration file (but must not have the same switch number).

Configuring VSL Port Channel and Ports

The VSL is configured with a unique port channel on each chassis. During the conversion, the VSS configures both port channels on the VSS active chassis. If the VSS standby chassis VSL port channel number has been configured for another use, the VSS comes up in RPR mode. To avoid this situation, check that both port channel numbers are available on both of the chassis.

Check the port channel number by using the **show running-config interface port-channel** command. The command displays an error message if the port channel is available for VSL. For example, the following command shows that port channel 20 is available on Switch 1:

```
Switch-1 # show running-config interface port-channel 20
% Invalid input detected at '^' marker.
```

To configure the VSL port channels, perform this task on Switch 1:

	Command	Purpose
Step 1	`Switch-1(config)# interface port-channel 10`	Configures port channel 10 on Switch 1.
Step 2	`Switch-1(config-if)# switch virtual link 1`	Associates Switch 1 as owner of port channel 10.
Step 3	`Switch-1(config-if)# no shutdown`	Activates the port channel.
Step 4	`Switch-1(config-if)# exit`	Exits interface configuration.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	`Switch-2(config)# interface port-channel 20`	Configures port channel 20 on Switch 2.
Step 2	`Switch-2(config-if)# switch virtual link 2`	Associates Switch 2 as owner of port channel 20.
Step 3	`Switch-2(config-if)# no shutdown`	Activates the port channel.
Step 4	`Switch-2(config-if)# exit`	Exits interface configuration mode.

You must add the VSL physical ports to the port channel. In the following example, interfaces 10-Gigabit Ethernet 3/1 and 3/2 on Switch 1 are connected to interfaces 10-Gigabit Ethernet 5/2 and 5/3 on Switch 2.

Tip For line redundancy, we recommend configuring at least two ports per switch for the VSL. For module redundancy, the two ports can be on different switching modules in each chassis.

To configure the VSL ports, perform this task on Switch 1:

	Command	Purpose
Step 1	`Switch-1(config)# interface range tengigabitethernet 3/1-2`	Enters configuration mode for interface range tengigabitethernet 3/1-2 on Switch 1.
Step 2	`Switch-1(config-if)# channel-group 10 mode on`	Adds this interface to channel group 10.
Step 3	`witch-1(config-if)# no shutdown`	Activates the port.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	`Switch-2(config)#` **`interface range tengigabitethernet 5/2-3`**	Enters configuration mode for interface range tengigabitethernet 5/2-3 on Switch 2.
Step 2	`Switch-2(config-if)#` **`channel-group 20 mode on`**	Adds this interface to channel group 20.
Step 3	`Switch-2(config-if)#` **`no shutdown`**	Activates the port.

Converting the Chassis to Virtual Switch Mode

Conversion to virtual switch mode requires a restart for both chassis. After the reboot, commands that specify interfaces with module/port now include the switch number. For example, a port on a switching module is specified by switch/module/port.

Prior to the restart, the VSS converts the startup configuration to use the switch/module/port convention. A backup copy of the startup configuration file is saved on the RP. This file is assigned a default name, but you are also prompted to override the default name if you want to change it.

Prior to the conversion, ensure that the PFC operating mode matches on both chassis. If they do not match, VSS comes up in RPR redundancy mode. Enter the **show platform hardware pfc mode** command on each chassis to display the current PFC mode. If only one of the chassis is in PFC3CXL mode, you can configure it to use PFC3C mode with the **platform hardware vsl pfc mode pfc3c** command.

To verify the PFC operating mode, perform this task:

	Command	Purpose
Step 1	`Switch-1#` **`show platform hardware pfc mode`**	Ensures that the PFC operating mode matches on both chassis, to ensure that the VSS comes up in SSO redundancy mode.
Step 2	`Switch-2#` **`show platform hardware pfc mode`**	Ensures that the PFC operating mode matches on both chassis, to ensure that the VSS comes up in SSO redundancy mode.
Step 3	`Switch-1(config)#` **`platform hardware vsl pfc mode pfc3c`**	(Optional) Sets the PFC operating mode to PFC3C on Chassis A.
Step 4	`Switch-2(config)#` **`platform hardware vsl pfc mode pfc3c`**	(Optional) Sets the PFC operating mode to PFC3C on Chassis B.

To convert Chassis 1 to virtual switch mode, perform this task:

Command	Purpose
`Switch-1#` **`switch convert mode virtual`**	Converts Switch 1 to virtual switch mode. After you enter the command, you are prompted to confirm the action. Enter **yes**. The system creates a converted configuration file, and saves the file to the RP bootflash.

To convert Chassis 2 to virtual switch mode, perform this task on Switch 2:

Command	Purpose
`Switch-2# switch convert mode virtual`	Converts Switch 2 to virtual switch mode. After you enter the command, you are prompted to confirm the action. Enter **yes**. The system creates a converted configuration file, and saves the file to the RP bootflash.

Note After you confirm the command (by entering **yes** at the prompt), the running configuration is automatically saved as the startup configuration and the chassis reboots. After the reboot, the chassis is in virtual switch mode, so you must specify interfaces with three identifiers (switch/module/port).

(Optional) Configuring VSS Standby Chassis Modules

After the reboot, each chassis contains the module provisioning for its own slots. In addition, the modules from the VSS standby chassis have been automatically provisioned on the VSS active chassis with default configuration.

Configurations for the VSS standby chassis modules are restored to their default settings (for example, no IP addresses).

You can view the module provisioning information in the configuration file, by entering the **show startup-config** command (after you have saved the configuration).

Note Do not delete or modify this section of the configuration file. In Cisco IOS Release 12.2(33)SXI and later releases, you can no longer add module provisioning entries using the **module provision** CLI command. When a module is not present, the provisioning entry for that module can be cleared using the **no slot** command with the **module provision** CLI command. Note that the VSS setup does not support the **module clear-config** command.

The following example shows the module provisioning information from a configuration file:

```
module provision switch 1
 slot 1 slot-type 148 port-type 60 number 4  virtual-slot 17
 slot 2 slot-type 137 port-type 31 number 16  virtual-slot 18
 slot 3 slot-type 227 port-type 60 number 8  virtual-slot 19
 slot 4 slot-type 225 port-type 61 number 48  virtual-slot 20
 slot 5 slot-type 82 port-type 31 number 2  virtual-slot 21
module provision switch 2
 slot 1 slot-type 148 port-type 60 number 4  virtual-slot 33
 slot 2 slot-type 227 port-type 60 number 8  virtual-slot 34
 slot 3 slot-type 137 port-type 31 number 16  virtual-slot 35
 slot 4 slot-type 225 port-type 61 number 48  virtual-slot 36
 slot 5 slot-type 82 port-type 31 number 2  virtual-slot 37
```

Displaying VSS Information

To display basic information about the VSS, perform one of these tasks:

Command	Purpose
`Router# show switch virtual`	Displays the virtual switch domain number, and the switch number and role for each of the chassis.
`Router# show switch virtual role`	Displays the role, switch number, and priority for each of the chassis in the VSS.
`Router# show switch virtual link`	Displays the status of the VSL.

The following example shows the information output from these commands:

```
Router# show switch virtual
Switch mode                  : Virtual Switch
Virtual switch domain number : 100
Local switch number          : 1
Local switch operational role: Virtual Switch Active
Peer switch number           : 2
Peer switch operational role : Virtual Switch Standby



Router# show switch virtual role
Switch  Switch Status  Preempt    Priority  Role     Session ID
        Number         Oper(Conf) Oper(Conf)         Local  Remote
------------------------------------------------------------------
LOCAL    1      UP     FALSE(N)   100(100)  ACTIVE   0      0
REMOTE   2      UP     FALSE(N)   100(100)  STANDBY  8158   1991

In dual-active recovery mode: No

Router# show switch virtual link
VSL Status: UP
VSL Uptime: 4 hours, 26 minutes
VSL SCP Ping: Pass OK
VSL ICC (Ping): Pass
VSL Control Link: Te 1/5/1
```

Converting a VSS to Standalone Chassis

To convert a VSS into two standalone chassis, you perform the following major steps:

- Copying the VSS Configuration to a Backup File, page 4-37
- Converting the VSS Active Chassis to Standalone, page 4-37
- Converting the Peer Chassis to Standalone, page 4-37

Copying the VSS Configuration to a Backup File

Save the configuration file from the VSS active chassis. You may need this file if you convert to virtual switch mode again. You only need to save the file from the VSS active chassis, because the configuration file on the VSS standby chassis is identical to the file on the VSS active chassis.

	Command	Purpose
Step 1	`Switch-1#` **`copy running-config startup-config`**	(Optional) Saves the running configuration to startup configuration. This step is only required if there are unsaved changes in the running configuration that you want to preserve.
Step 2	`Switch-1#` **`copy startup-config disk0:vs-startup-config`**	Copies the startup configuration to a backup file.

Converting the VSS Active Chassis to Standalone

When you convert the VSS active chassis to standalone mode, the VSS active chassis removes the provisioning and configuration information related to VSL links and the peer chassis modules, saves the configuration file, and performs a reload. The chassis comes up in standalone mode with only the provisioning and configuration data relevant to the standalone system.

The VSS standby chassis of the VSS becomes VSS active. VSL links on this chassis are down because the peer is no longer available.

To convert the VSS active chassis to standalone mode, perform this task on the VSS active chassis:

Command	Purpose
`Switch-1#` **`switch convert mode stand-alone`**	Converts Switch 1 to standalone mode. After you enter the command, you are prompted to confirm the action. Enter **yes**.

Converting the Peer Chassis to Standalone

When you convert the new VSS active chassis to standalone mode, the chassis removes the provisioning and configuration information related to VSL links and the peer chassis modules, saves the configuration file and performs a reload. The chassis comes up in standalone mode with only its own provisioning and configuration data.

To convert the peer chassis to standalone, perform this task on the VSS standby chassis:

Command	Purpose
`Switch-2#` **`switch convert mode stand-alone`**	Converts Switch 2 to standalone mode. After you enter the command, you are prompted to confirm the action. Enter **yes**.

Configuring VSS Parameters

These sections describe how to configure VSS parameters:

Configuring VSL Switch Priority

To configure the switch priority, perform this task:

	Command	Purpose
Step 1	Router(config)# **switch virtual domain 100**	Enters configuration mode for the virtual switch domain.
Step 2	Router(config-vs-domain)# **switch** [**1** | **2**] **priority** [*priority_num*]	Configures the priority for the chassis. The switch with the higher priority assumes the VSS active role. The range is 1 (lowest priority) to 255 (highest priority); the default is 100. **Note** • The new priority value only takes effect after you save the configuration and perform a reload of the VSS. • If the higher priority switch is currently in VSS standby state, you can make it the VSS active switch by initiating a switchover. Enter the **redundancy force-switchover** command. • The **show switch virtual role** command displays the operating priority and the configured priority for each switch in the VSS. • The **no** form of the command resets the priority value to the default priority value of 100. The new value takes effect after you save the configuration and perform a reload.
Step 3	Router# **show switch virtual role**	Displays the current priority.

Note If you make configuration changes to the switch priority, the changes only take effect after you save the running configuration to the startup configuration file and perform a reload. The **show switch virtual role** command shows the operating and priority values. You can manually set the VSS standby switch to VSS active using the **redundancy force-switchover** command.

This example shows how to configure virtual switch priority:

```
Router(config)# switch virtual domain 100
Router(config-vs-domain)# switch 1 priority 200
Router(config-vs-domain)# exit
```

This example shows how to display priority information for the VSS:

```
Router# show switch virtual role
Switch   Switch Status  Preempt    Priority  Role     Session ID
         Number         Oper(Conf) Oper(Conf)         Local  Remote
------------------------------------------------------------------
LOCAL     1     UP      FALSE(N)   100(200)  ACTIVE   0      0
REMOTE    2     UP      FALSE(N)   100(100)  STANDBY  8158   1991

In dual-active recovery mode: No
```

Configuring PFC Mode

If you have a mixture of DFC3C and DFC3CXL switching modules in the VSS, set the PFC mode by performing this task:

Command	Purpose
Router(config)# **platform hardware vsl pfc mode pfc3c**	Sets the PFC configuration mode for the VSS to PFC3C. **Note** This command requires a system reload before it takes effect.
Router# **show platform hardware pfc mode**	Displays the current settings for the PFC mode.

This example shows how to set the PFC configuration mode for the VSS to PFC3C. You can wait until the next maintenance window to perform the **reload** command.

```
Router(config)# platform hardware vsl pfc mode pfc3c
Router(config)# end
Router# reload
```

If all the supervisor engines and switching modules in the VSS are 3CXL, the following warning is displayed if you set the PFC mode to PFC3C:

```
Router(config)# platform hardware vsl pfc mode pfc3c
PFC Preferred Mode: PFC3CXL. The discrepancy between Operating Mode and
Preferred Mode could be due to PFC mode config. Your System has all PFC3XL modules.
Remove ' platform hardware vsl pfc mode pfc3c ' from global config.
```

This example shows how to display the operating and configured PFC modes:

```
Router# show platform hardware pfc mode
PFC operating mode : PFC3C
Configured PFC operating mode : PFC3C
```

Configuring a VSL

To configure a port channel to be a VSL, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface port-channel channel_num`	Enters configuration mode for the specified port channel.
Step 2	`Router(config-if)# switch virtual link switch_num`	Assigns the port channel to the virtual link for the specified switch.

Note We recommend that you configure the VSL prior to converting the chassis into a VSS.

This example shows how to configure the VSL:

```
Switch-1(config)# interface port-channel 10
Switch-1(config-if)# switch virtual link 1
Switch-1(config-if)# no shutdown
Switch-1(config)# interface tenGigabitEthernet 5/1
Switch-1(config-if)# channel-group 10 mode on
Switch-1(config-if)# no shutdown

Switch-2(config)# interface port-channel 25
Switch-2(config-if)# switch virtual link 2
Switch-2(config-if)# no shutdown
Switch-2(config-if)# interface tenGigabitEthernet 5/2
Switch-2(config-if)# channel-group 25 mode on
Switch-2(config-if)# no shutdown
```

Displaying VSL Information

To display information about the VSL, perform one of these tasks:

Command	Purpose
`Router# show switch virtual link`	Displays information about the VSL.
`Router# show switch virtual link port-channel`	Displays information about the VSL port channel.
`Router# show switch virtual link port`	Displays information about the VSL ports.

This example shows how to display VSL information:

```
Router# show switch virtual link
VSL Status : UP
VSL Uptime : 1 day, 3 hours, 39 minutes
VSL SCP Ping : Pass
VSL ICC Ping : Pass
VSL Control Link : Te 1/5/1

Router# show switch virtual link port-channel
VSL Port Channel Information

Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
```

```
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, no aggregation due to minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-------------------------------------------
10     Po10(RU)         -        Te1/5/4(P) Te1/5/5(P)
20     Po20(RU)         -        Te2/5/4(P) Te2/5/5(P)

Router# show switch virtual link port
VSL Link Info          : Configured: 2 Operational: 1


                          Peer         Peer    Peer
Interface   State         MAC          Switch  Interface
----------------------------------------------------------------------
Te1/5/4   operational   0013.5fcb.1480  2    Te2/5/4
Te1/5/5   link_down     -               -     -

            Last operational         Current packet       Last Diag   Time since
Interface   Failure state            State                Result      Last Diag
--------------------------------------------------------------------------------
Te1/5/4 No failure              Hello bidir            Never ran   7M:51S
Te1/5/5 No failure              No failure             Never ran   7M:51S


                            Hello Tx (T4) ms        Hello Rx (T5*) ms
 Interface  State        Cfg      Cur     Rem     Cfg      Cur     Rem
----------------------------------------------------------------------
Te1/5/4 operational  500     500     404     5000    5000    4916
Te1/5/5 link_down    500     -       -       500000  -       -
Te2/5/4 operational  500     500     404     500000  500000  499916
Te2/5/5 link_down    500     -       -       500000  -       -
 *T5 = min_rx * multiplier
```

Configuring VSL QoS

The VSS automatically configures VSL ports for trust CoS, using default CoS mappings (you cannot change the mappings on VSL ports).

For switching modules that support per-ASIC configuration, the VSL configuration applies to all ports on the same ASIC (including any non-VSL ports).

The VSS disables the QoS commands on VSL ports (and any non-VSL ports on the same ASIC). For example, you cannot use QoS queuing or map commands on VSL ports.

To ensure that all eight QoS receive queues are enabled for the 10-Gigabit Ethernet ports on the supervisor engine, enter the **mls qos 10g-only** global configuration command.

In Cisco IOS Release 12.2(33)SXI and later releases, when the **mls qos 10g-only** command is entered and only one of the two 10-Gigabit Ethernet ports on the supervisor engine is a VSL port, the non-VSL 10-Gigabit Ethernet port can be configured for QoS.

Subcommands for VSL Port Channels

On a VSL port channel, only a subset of interface subcommands are available in the command console. Table 4-2 describes the available interface subcommands.

Table 4-2 Interface Subcommands for VSL Port Channels

Subcommand	Description
default	Sets a command to its defaults.
description	Enters a text description for the interface.
exit	Exits from interface configuration mode.
load-interval	Specifies interval for load calculation for an interface.
logging	Configures logging for interface.
mls	Specifies multilayer switching subcommands.
no	Disables a command, or sets the command defaults.
shutdown	Shuts down the selected interface.
switch virtual link	Specifies the switch associated with this port channel.
vslp	Specifies VSLP interface configuration commands.

Subcommands for VSL Ports

If a port is included in a VSL port channel, only a subset of interface subcommands are available in the command console. Table 4-3 describes the available interface subcommands.

Table 4-3 Interface Subcommands for VSL Ports

Subcommand	Description
channel-group	Adds the interface to the specified channel group.
default	Sets a command to its defaults.
description	Adds a description to the interface.
exit	Exits from interface configuration mode.
load-interval	Specifies interval for load calculation for an interface.
logging	Configures logging for the interface.
no	Disables a command, or sets the command defaults.
shutdown	Shuts down the selected interface.

Configuring the Router MAC Address Assignment

When the VSS is started for the first time, the initial VSS active supervisor engine assigns a router MAC address for the VSS. By default, the supervisor engine assigns a MAC address from its own chassis. After a switchover to the second chassis, the VSS continues to use the MAC address from the previously VSS active chassis as the router MAC address.

In the rare case where both chassis later become inactive, and then they start up with the second supervisor engine becoming the initial VSS active supervisor engine, the VSS will start up with a router MAC address from the second chassis. Other Layer 2 hosts that do not respond to GARP and are not directly connected to the VSS will retain the earlier router MAC address of the VSS, and will not be able to communicate with the VSS. To avoid this possibility, you can configure the VSS to assign a router MAC address from a reserved pool of addresses with the domain ID encoded in the last octet of the MAC address, or you can specify a MAC address.

Note If you change the router MAC address, you must reload the virtual switch for the new router MAC address to take effect.

To specify that the router MAC address is assigned from a reserved pool of domain-based addresses, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain`** *`domain_id`*	Enters VSS configuration mode.
Step 2	`Router(config-vs-domain)#` **`mac-address use-virtual`**	The router MAC address is assigned from a reserved pool of domain-based addresses. **Note** The **no** form of this command reverts to the default setting, using a MAC address from the backplane of the initial VSS active chassis.

To specify a router MAC address, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain`** *`domain_id`*	Enters VSS configuration mode.
Step 2	`Router(config-vs-domain)#` **`mac-address`** *`mac-address`*	The router MAC address is specified in three 2-byte hexadecimal numbers.

This example shows how to configure router MAC address assignment from a reserved pool of domain-based addresses:

```
Router(config)# switch virtual domain 255
Router(config-vs-domain)# mac-address use-virtual
```

The following example shows how to specify the router MAC address in hexadecimal format:

```
Router(config)# switch virtual domain 255
Router(config-vs-domain)# mac-address 0123.4567.89ab
```

Configuring Deferred Port Activation During VSS Standby Recovery

Instead of allowing all ports to be activated simultaneously when a failed chassis is restarted as the VSS standby chassis, you can configure the system to defer activation of non-VSL ports and then activate the ports in groups over a period of time.

To specify deferred port activation, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain 1`**	Enters VSS configuration mode.
Step 2	`Router(config-vs-domain)#` **`standby port delay`** *`delay-time`*	Specifies that the port activation will be initially deferred and then performed in cycles. For *delay-time*, specify the period in seconds before port activation will begin. The range is 30 to 3600.
Step 3	`Router(config-vs-domain)#` **`standby port bringup`** *`number cycle-time`*	Specifies the number of ports to be activated per cycle and the waiting time between cycles. For *number*, specify the number of ports to be activated per cycle. The range is 1 to 100. The default value is 1 port. For *cycle-time*, specify the period in seconds between cycles. The range is 1 to 10. The default value is 1 second.

This example shows how to configure port activation to be deferred by 120 seconds, then activated in groups of 20 ports every 5 seconds:

```
Router(config)# switch virtual domain 1
Router(config-vs-domain)# standby port delay 120
Router(config-vs-domain)# standby port bringup 20 5
```

Configuring Multichassis EtherChannels

Configure multichassis EtherChannels (MECs) as you would for a regular EtherChannel. The VSS will recognize that the EtherChannel is an MEC when ports from both chassis are added to the EtherChannel. You can verify the MEC configuration by entering the **show etherchannel** command.

One VSS supports a maximum of 512 port channels.

Note Releases earlier than Cisco IOS Release 12.2(33)SXI support a maximum of 128 port channels.

The Configuring Port Load Share Deferral on the Peer Switch section provides additional details about MECs:

Configuring Port Load Share Deferral on the Peer Switch

To configure the load share deferral feature for a port channel, perform this task on the switch that is an MEC peer to the VSS:

	Command	Purpose
Step 1	`Router(config)# ` **`port-channel load-defer`** *`seconds`*	(Optional) Configures the port load share deferral interval for all port channels. • *seconds*—The time interval during which load sharing is initially 0 for deferred port channels. The range is 1 to 1800 seconds; the default is 120 seconds.
Step 2	`Router(config)# ` **`interface port-channel`** *`channel-num`*	Enters interface configuration mode for the port channel.
Step 3	`Router(config-if)# ` **`port-channel port load-defer`**	Enables port load share deferral on the port channel.

This example shows how to configure the load share deferral feature on port channel 10 of the switch that is an MEC peer to the VSS:

```
Router(config)# port-channel load-defer 60
Router(config)# interface port-channel 10
Router(config-if)# port-channel port load-defer
This will enable the load share deferral feature on this port-channel.
Do you wish to proceed? [yes/no]: yes
```

Configuring Dual-Active Detection

The following sections describe how to configure dual-active detection:

Configuring Enhanced PAgP Dual-Active Detection

If enhanced PAgP is running on the MECs between the VSS and its access switches, the VSS can use enhanced PAgP messaging to detect a dual-active scenario.

By default, PAgP dual-active detection is enabled. However, the enhanced messages are only sent on port channels with trust mode enabled (see the trust mode description in the note).

Note Before changing PAgP dual-active detection configuration, ensure that all port channels with trust mode enabled are in administrative down state. Use the **shutdown** command in interface configuration mode for the port channel. Remember to use the **no shutdown** command to reactivate the port channel when you are finished configuring dual-active detection.

To enable or disable PAgP dual-active detection, perform this task:

	Command	Purpose
Step 1	Router(config)# **switch virtual domain** *domain_id*	Enters virtual switch submode.
Step 2	Router(config-vs-domain)# **dual-active detection pagp**	Enables sending of the enhanced PAgP messages.

You must configure trust mode on the port channels that will detect PAgP dual-active detection. By default, trust mode is disabled.

Note If PAgP dual-active detection is enabled, you must place the port channel in administrative down state before changing the trust mode. Use the **shutdown** command in interface configuration mode for the port channel. Remember to use the **no shutdown** command to reactivate the port channels when you are finished configuring trust mode on the port channel.

To configure trust mode on a port channel, perform this task:

	Command	Purpose
Step 1	Router(config)# **switch virtual domain** *domain_id*	Enters virtual switch submode.
Step 2	Router(config-vs-domain)# **dual-active detection pagp trust channel-group** *group_number*	Enables trust mode for the specified port channel.

This example shows how to enable PAgP dual-active detection:

```
Router(config)# interface port-channel 20
Router(config-if)# shutdown
Router(config-if)# exit
Router(config)# switch virtual domain 100
Router(config-vs-domain)# dual-active detection pagp
Router(config-vs-domain)# dual-active detection pagp trust channel-group 20
Router(config-vs-domain)# exit
Router(config)# interface port-channel 20
Router(config-if)# no shutdown
Router(config-if)# exit
```

This example shows the error message if you try to enable PAgP dual-active detection when a trusted port channel is not shut down first:

```
Router(config)# switch virtual domain 100
Router(config-vs-domain)# dual-active detection pagp
Trusted port-channel 20 is not administratively down.
To change the pagp dual-active configuration, “shutdown” these port-channels first.
Remember to “no shutdown” these port-channels afterwards.
```

This example shows the error message if you try to configure trust mode for a port channel that is not shut down first:

```
Router(config)# switch virtual domain 100
```

```
Router(config-vs-domain)# dual-active detection pagp trust channel-group 20
Trusted port-channel 20 is not administratively down. To change the pagp dual-active trust
configuration, "shutdown" the port-channel first. Remember to "no shutdown" the
port-channel afterwards.
```

Configuring BFD Dual-Active Detection

For the BFD dual-active detection, you must configure dual-active interface pairs that will act as BFD messaging links. By default, BFD detection is enabled.

To configure BFD dual-active detection, perform this task:

	Command	Purpose
Step 1	Router(config)# **switch virtual domain** *domain_id*	Enters virtual switch submode.
Step 2	Router(config-vs-domain)# **dual-active detection bfd**	Enables BFD dual-active detection method. By default, BFD detection is enabled.
Step 3	Router(config-vs-domain)# **dual-active pair interface** *int_1* **interface** *int_2* **bfd**	Configures the dual-active pair of interfaces. The interfaces *int_1* and *int_2* are of the form *type*[1] *switch/slot/port*. The interfaces must be directly connected (a single Layer 3 hop).

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When you configure the dual-active interface pairs, note the following information:

- The individual ports must be configured first with both an IP address and BFD configuration. The configuration is validated when you add the dual-active interface pair.
- The IP addresses assigned to the dual-active pair must be from two different networks or subnetworks.
- The BFD timers must be configured with the same values on the ports at both ends of the link to ensure proper operation of Layer 3 BFD dual-active detection.
- The MAC address cannot be specified on the interface.

Note We recommend that you configure a short BFD interval and small multiplier value (such as 50 to 100 ms for the interval and 3 as the multiplier value). If the interval and multiplier values are large, there is a long delay before the system initiates dual-active mode recovery. This condition can cause network instability and poor convergence.

This example shows how to configure interfaces for BFD dual-active detection:

```
Router (config)# interface gigabitethernet 1/9/48
Router (config-if)# no switchport
Router (config-if)# ip address 200.230.230.231 255.255.255.0
Router (config-if)# bfd interval 100 min_rx 100 multiplier 3
Router (config-if)# no shutdown
Router (config-if)# interface gigabitethernet 2/1/48
Router (config-if)# no switchport
Router (config-if)# ip address 201.230.230.231 255.255.255.0
Router (config-if)# bfd interval 100 min_rx 100 multiplier 3
Router (config-if)# no shutdown
Router (config-if)# exit
Router (config)# switch virtual domain 100
```

```
Router (config-vs-domain)# dual-active detection bfd
Router (config-vs-domain)# dual-active pair interface g 1/9/48 interface g 2/1/48 bfd

adding a static route 200.230.230.0 255.255.255.0 Gi2/1/48 for this dual-active pair
adding a static route 201.230.230.0 255.255.255.0 Gi1/9/48 for this dual-active pair
Router(config-vs-domain)# exit
Router(config)# exit
Router# show switch virtual dual-active bfd
Bfd dual-active detection enabled: Yes
Bfd dual-active interface pairs configured:
  interface1 Gi1/9/48 interface2 Gi2/1/48
```

Configuring Fast Hello Dual-Active Detection

Fast hello dual-active detection is enabled by default; however, you must configure dual-active interface pairs to act as fast hello dual-active messaging links.

To configure fast hello dual-active detection, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`switch virtual domain`** *`domain_id`*	Enters the virtual switch submode.
Step 2	`Router(config-vs-domain)#` **`dual-active detection fast-hello`**	Enables the fast hello dual-active detection method. Fast hello dual-active detection is enabled by default.
Step 3	`Router(config-vs-domain)#` **`exit`**	Exits virtual switch submode.
Step 4	`Router(config)#` **`interface`** *`type`*[1] *`switch/slot/port`*	Selects the interface to configure. This interface must be directly connected to the other chassis and must not be a VSL link.
Step 5	`Router(config-if)#` **`dual-active fast-hello`**	Enables fast hello dual-active detection on the interface, automatically removes all other configuration from the interface, and restricts the interface to dual-active configuration commands.
Step 6	`Router(config-if)#` **`no shutdown`**	Activates the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When you configure fast hello dual-active interface pairs, note the following information:

- You can configure a maximum of four interfaces on each chassis to connect with the other chassis in dual-active interface pairs.
- Each interface must be directly connected to the other chassis and must not be a VSL link. We recommend using links from a switching module not used by the VSL.
- Each interface must be a physical port. Logical ports such as an SVI are not supported.
- Configuring fast hello dual-active mode will automatically remove all existing configuration from the interface and will restrict the interface to fast hello dual-active configuration commands.
- Unidirectional link detection (UDLD) will be disabled on fast hello dual-active interface pairs.

This example shows how to configure an interface for fast hello dual-active detection:

```
Router(config)# switch virtual domain 255
Router(config-vs-domain)# dual-active detection fast-hello
Router(config-vs-domain)# exit
Router(config)# interface fastethernet 1/2/40
```

```
Router(config-if)# dual-active fast-hello
WARNING: Interface FastEthernet1/2/40 placed in restricted config mode. All extraneous
configs removed!

Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# exit
Router# show run interface fastethernet 1/2/40
interface FastEthernet1/2/40
 no switchport
 no ip address
 dual-active fast-hello
end
```

Configuring the Exclusion List

When a dual-active scenario is detected, part of the recovery action is for the chassis to shut down all of its non-VSL interfaces. You can specify one or more interfaces to be excluded from this action (for example, to exclude the interface you use for remote access to the chassis).

To specify interfaces that are not to be shut down by dual-active recovery, perform this task:

	Command	Purpose
Step 1	Router(config)# **switch virtual domain** *domain_id*	Enters virtual switch submode.
Step 2	Router(config-vs-domain)# **dual-active exclude interface** *type*[1] *switch/slot/port*	Specifies an interface to exclude from shutting down in dual-active recovery.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When you configure the exclusion list, note the following information:

- The interface must be a physical port configured with an IP address.
- The interface must not be a VSL port.
- The interface must not be in use for IP BFD dual-active detection.
- The interface must not be in use for fast hello dual-active detection.

This example shows how to configure an interface as an exclusion:

```
Router(config)# switch virtual domain 100
Router (config-vs-domain)# dual-active exclude interface gigabitethernet 1/5/5
```

Displaying Dual-Active Detection

To display information about dual-active detection, perform this task:

Command	Purpose
Router# **show switch virtual dual-active** [**bfd** | **pagp** | **fast-hello** | **summary**]	Displays information about dual-active detection configuration and status.

This example shows how to display the summary status for dual-active detection:

```
Router# show switch virtual dual-active summary
Pagp dual-active detection enabled: Yes
```

```
Bfd dual-active detection enabled: Yes
Fast-hello dual-active detection enabled: Yes

No interfaces excluded from shutdown in recovery mode
In dual-active recovery mode: No
```

This example shows how to display information for BFD dual-active detection:

```
Router# show switch virtual dual-active bfd
Bfd dual-active detection enabled: Yes
Bfd dual-active interface pairs configured:
  interface1 Gi1/9/48 interface2 Gi2/1/48
```

This example shows how to display information for fast-hello dual-active detection:

```
Router# show switch virtual dual-active fast-hello
Fast-hello dual-active detection enabled: Yes

Fast-hello dual-active interfaces:
Port       State (local only)
-----------------------------
Gi1/4/47   Link dn
Gi2/4/47   -
```

This example shows how to display PAgP status and the channel groups with trust mode enabled:

```
Router# show pagp dual-active
PAgP dual-active detection enabled: Yes
PAgP dual-active version: 1.1

Channel group 3 dual-active detect capability w/nbrs Dual-Active trusted group: No
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Fa1/2/33  No              None                 None      N/A

Channel group 4
Dual-Active trusted group: Yes
No interfaces configured in the channel group

Channel group 5
Dual-Active trusted group: Yes
Channel group 5 is not participating in PAGP

Channel group 10 dual-active detect capability w/nbrs Dual-Active trusted group: Yes
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Gi1/6/1   Yes             partner-1            Gi1/5/1   1.1
Gi2/5/1   Yes             partner-1            Gi1/5/2   1.1

Channel group 11 dual-active detect capability w/nbrs Dual-Active trusted group: No
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Gi1/6/2   Yes             partner-1            Gi1/3/1   1.1
Gi2/5/2   Yes             partner-1            Gi1/3/2   1.1

Channel group 12 dual-active detect capability w/nbrs Dual-Active trusted group: Yes
          Dual-Active     Partner              Partner   Partner
Port      Detect Capable  Name                 Port      Version
Fa1/2/13  Yes             partner-1            Fa1/2/13  1.1
Fa1/2/14  Yes             partner-1            Fa1/2/14  1.1
Gi2/1/15  Yes             partner-1            Fa1/2/15  1.1
Gi2/1/16  Yes             partner-1            Fa1/2/16  1.1
```

> **Note** The **show switch virtual dual-active pagp** command displays the same output as the **show pagp dual-active** command.

Configuring Service Modules in a VSS

To configure a service module in a VSS, you must add the switch number to many of the configuration commands, as described in this section.

> **Note** For detailed instructions on configuring a service module in a VSS, see the configuration guide and command reference for the service module.

The following sections provide examples of how to configure a service module in a VSS:

Opening a Session with a Service Module in a VSS

To configure service modules that require opening a session, perform this task:

Command	Purpose
`Router# session switch num slot slot processor processor-id`	Opens a session with the specified module. • *num*—Specifies the switch to access; valid values are 1 and 2. • *slot*—Specifies the slot number of the module. • *processor-id*—Specifies the processor ID number. Range: 0 to 9.

This example shows how to open a session to a Firewall Service Module in a VSS:

```
Router# session switch 1 slot 4 processor 1
The default escape character is Ctrl-^, then x.
You can also type 'exit' at the remote prompt to end the session
Trying 127.0.0.41 ... Open
```

Assigning a VLAN Group to a Firewall Service Module in a VSS

To assign a VLAN group to a FWSM, perform this task:

Command	Purpose
`Router(config)# firewall switch num slot slot vlan-group [vlan_group | vlan_range]`	Assigns VLANs to a firewall group in the specified module. • *num*—Specifies the switch to access; valid values are 1 and 2. • *slot*—Specifies the slot number of the module. • *vlan_group*—Specifies the group ID as an integer. • *vlan_range*—Specifies the VLANs assigned to the group.

This example shows how to assign a VLAN group to a Firewall Service Module in a VSS:

```
Router(config)# firewall switch 1 slot 4 vlan-group 100,200
```

Assigning a VLAN Group to an ACE Service Module in a VSS

To assign a VLAN group to an ACE, perform this task:

	Command	Purpose
Step 1	`Router(config)# svclc multiple-vlan-interfaces`	Enables multiple VLAN interfaces mode for service modules.
Step 2	`Router(config)# svclc switch num slot slot vlan-group [vlan_group | vlan_range]`	Assign VLANs to a firewall group in the specified module. • *num*—Specifies the switch to access; valid values are 1 and 2. • *slot*—Specifies the slot number of the module. • *vlan_group*—Specifies the group ID as an integer. • *vlan_range*—Specifies the VLANs assigned to the group.

This example shows how to assign multiple VLAN groups to an ACE service module in a VSS:

```
Router(config)# svclc multiple-vlan-interfaces
Router(config)# svclc switch 1 slot 4 vlan-group 100,200
```

Verifying Injected Routes in a Service Module in a VSS

To view route health injection (RHI) routes, perform this task:

Command	Purpose
Router# **show svclc rhi-routes switch** *num* **slot** *slot*	Displays injected RHI routes in the specified service module. • *num*—Specifies the switch to access; valid values are 1 and 2. • *slot*—Specifies the slot number of the module.

This example shows how to view injected routes in a service module in a VSS:

```
Router# show svclc rhi-routes switch 1 slot 4
RHI routes added by slot 34
        ip              mask            nexthop      vlan  weight tableid
  --------------- --------------- --------------- ------ ------ -------
A 23.1.1.4        255.255.255.252 20.1.1.1        20     1      0
```

Viewing Chassis Status and Module Information in a VSS

To view chassis status and information about modules installed in either or both chassis of a VSS, perform the following task:

Command	Purpose
Router# **show module switch** { **1** | **2** | **all** }	Displays information about modules in the specified chassis (**1** or **2**), or in both chassis (**all**).

This example shows how to view the chassis status and module information for chassis number 1 of a VSS:

```
Router# show module switch 1
 Switch Number:     1   Role:   Virtual Switch Active
----------------------  -----------------------------
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  1   48  CEF720 48 port 10/100/1000mb Ethernet  WS-X6748-GE-TX     SAL1215M2YA
  2    8  CEF720 8 port 10GE with DFC            WS-X6708-10GE      SAL1215M55F
  3    1  Application Control Engine Module      ACE20-MOD-K9       SAD120603SU
.
.
.
```

Upgrading a VSS

Cisco IOS Rel ease 12.2(33)SXH supports a fast software upgrade (FSU) of the VSS using RPR. Cisco IOS Release 12.2(33)SXI and later releases support an enhanced fast software upgrade (eFSU) of the VSS using SSO.

This section describes both types of VSS upgrade:

Performing a Fast Software Upgrade of a VSS

The FSU of a VSS is similar to the RPR-based standalone chassis FSU described in the "Performing a Fast Software Upgrade" section on page 7-6. While the standalone chassis upgrade is initiated by reloading the VSS standby supervisor engine, the VSS upgrade is initiated by reloading the VSS standby chassis. During the FSU procedure, a software version mismatch between the VSS active and the VSS standby chassis causes the system to boot in RPR redundancy mode, which is stateless and causes a hard reset of the all modules. As a result, the FSU procedure requires system downtime corresponding to the RPR switchover time.

Note VSS mode supports only one supervisor engine in each chassis. If another supervisor engine resides in the chassis it will act as the DFC.

To perform an FSU of a VSS, perform this task:

	Command	Purpose
Step 1	`Router#` **`copy tftp`** *`disk_name`*	Uses TFTP to copy the new software image to flash memory on the VSS active and VSS standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 2	`Router#` **`config terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`no boot system`**	Removes any previously assigned boot variables.
Step 4	`Router(config)#` **`config-register 0x2102`**	Sets the configuration register.
Step 5	`Router(config)#` **`boot system flash`** *`device:file_name`*	Configures the chassis to boot the new image.
Step 6	`Router(config)#` **`end`**	Returns to privileged EXEC mode.
Step 7	`Router#` **`copy running-config startup-config`**	Saves the configuration.

	Command	Purpose
Step 8	`Router# redundancy reload peer`	Reloads the VSS standby chassis and brings it back online running the new version of the Cisco IOS software. Due to the software version mismatch between the two chassis, the VSS standby chassis will be in RPR redundancy mode. **Note** Before reloading the VSS standby chassis, make sure you wait long enough to ensure that all configuration synchronization changes have completed.
Step 9	`Router# redundancy force-switchover`	Forces the VSS standby chassis to assume the role of the VSS active chassis running the new Cisco IOS image. The modules are reloaded and the module software is downloaded from the new VSS active chassis. The old VSS active chassis reboots with the new image and becomes the VSS standby chassis.

This example shows how to perform an FSU:

```
Router# config terminal
Router(config)# no boot system
Router(config)# config-register 0x2102
Router(config)# boot system flash disk0:image_name
Router(config)# end
Router# copy running-config startup-config
Router# redundancy reload peer
Router# redundancy force-switchover
```

Performing an Enhanced Fast Software Upgrade of a VSS

An eFSU uses the same commands and software infrastructure as an in-service software upgrade (ISSU). The eFSU differs from an ISSU in that it resets the modules, which results in a brief traffic interruption. The eFSU sequence for a VSS follows the same logical steps as the single-chassis eFSU described in the "Performing an Enhanced Fast Software Upgrade" section on page 5-5, but the procedure applies to the VSS active and VSS standby supervisor engine in each chassis, instead of two supervisor engines in one chassis. During an eFSU, the VSS standby chassis, including the supervisor engine and modules, is upgraded and brought up in a stateful switchover (SSO) mode. The eFSU process then forces a switchover and performs the same upgrade on the other chassis, which becomes the new VSS standby.

Note VSS mode supports only one supervisor engine in each chassis. If another supervisor resides in the chassis it will act as the DFC.

This section contains the following topics:

eFSU Restrictions and Guidelines

When performing an eFSU, note the following guidelines and restrictions:

- 7600-SIP-400 is powered down during an eFSU and is powered up at the Commitversion Stage or at Abortversion (Optional).
- An eFSU can install a full image upgrade or a patch upgrade. Any patch upgrade will be installed by the same process as a full image upgrade, regardless of whether the patch requires a reload or a process restart.
- The new image file must reside in the file system of the supervisor engine in each chassis before the eFSU can be initiated. The **issu** commands will accept only global file system names (for example, disk0: or sup-bootdisk:). The **issu** commands will not accept switch number-specific file system names (for example, sw1-slot5-disk0:).
- When preparing for the eFSU, do not change the boot variable. Although a boot variable change is required in the FSU (RPR) procedure, changing the boot variable in the eFSU procedure will cause the CurrentVersion variable to be inconsistent, preventing execution of the eFSU.
- The **issu** commands for a VSS eFSU upgrade are similar to those for a single-chassis (standalone) eFSU, as described in the "Performing an Enhanced Fast Software Upgrade" section on page 5-5, with the following differences:
  - Where the standalone **issu** commands accept an argument of slot number, the VSS **issu** commands accept a switch and slot number, in the format of *switch/slot* (for example, 1/5 refers to switch 1, slot 5).
  - For a normal VSS eFSU, it is not necessary to specify a switch or slot number when entering the VSS **issu** commands.
- You cannot change the rollback timer period during the eFSU process.
- During the eFSU process, do not perform any manual switchover other than those caused by the **issu** commands.
- During the eFSU process, do not perform an online insertion or removal (OIR) of any module.
- During an eFSU downgrade, if the process is aborted (either due to an MCL error or by entering the **abortversion** command) just after executing the **loadversion** command, the SSO VSS standby is reloaded with the original image but the SSO VSS standby's ICS is not because the bootvar of the SSO VSS standby's ICS is not modified during an abort executed after the **loadversion** command.
- Images with different feature sets fail the eFSU compatibility check, regardless of the software release.
- The eFSU feature does not support upgrades or downgrades between modular and non-modular IOS versions.
- The eFSU feature does not support upgrades or downgrades between installed and binary modes of modular IOS. The Installed mode was removed after Cisco IOS Release 12.2(33)SXI3.
- Before you start a downgrade with eFSU (reverting to an earlier version of Cisco IOS software), remove configurations and disable any features or functions that are not supported in the earlier version. Otherwise the configuration files fail to synchronize and the standby supervisor engine reloads.
- The eFSU upgrade feature works with NSF/SSO. Software features that do not support NSF/SSO stop operating until after the software upgrade switchover, when they come back online.
- Images with release dates more than 18 months apart are not supported for eFSU. See the SX_SY_EFSU_Compatibility_Matrix to verify compatibility.

eFSU Stages for a VSS Upgrade

The eFSU sequence consists of several stages, each explicitly initiated by entering a specific **issu** command in the CLI. At each stage, you can verify the system status or roll back the upgrade before moving to the next stage.

The following sections describe the eFSU stages for a VSS upgrade:

Preparation

Before you can initiate the eFSU process, the upgrade image must reside in the file system of the supervisor engine in each chassis; otherwise, the initial command will be rejected. The VSS must be in a stable operating state with one chassis in the VSS active state and the other chassis in the hot VSS standby state.

Loadversion Stage

The eFSU process begins when you enter the **issu loadversion** command specifying the location in memory of the new upgrade images on the VSS active and VSS standby chassis. Although the **issu loadversion** command allows you to specify the switch and slot number of the VSS active and VSS standby chassis, it is not necessary to do so. When you enter the **issu loadversion** command, the entire VSS standby chassis, including the supervisor engine and modules, is reloaded with the new upgrade image. Because the VSS standby chassis modules are unavailable while reloading, the throughput of the VSS is temporarily reduced by 50 percent during this stage. After reloading, the VSS standby chassis boots with the new image and initializes in SSO mode, restoring traffic throughput. In this state, the VSS standby chassis runs a different software version than the VSS active chassis, which requires the VSS active chassis to communicate with modules running different image versions between the two chassis.

Runversion Stage

When the VSS standby chassis is successfully running the new image in SSO mode, you can enter the **issu runversion** command. This command forces a switchover in which the upgraded VSS standby chassis takes over as the new VSS active chassis. The formerly VSS active chassis reloads and initializes as the new VSS standby chassis in SSO mode, running the old image. As in the loadversion stage, the throughput of the VSS is temporarily reduced during the VSS standby chassis reload, and the VSS standby chassis runs a different software version than the VSS active chassis.

Acceptversion Stage (Optional)

When you enter the **issu runversion** command, a switchover to the chassis running the new image occurs, which starts an automatic rollback timer as a safeguard to ensure that the upgrade process does not cause the VSS to be nonoperational. Before the rollback timer expires, you must either accept or commit the new software image. If the timer expires, the upgraded chassis reloads and reverts to the previous software version. To stop the rollback timer, enter the **issu acceptversion** command. Prior to starting the eFSU process, you can disable the rollback timer or configure it to a value up to two hours (the default is 45 minutes).

Operating with an upgraded VSS active chassis, this stage allows you to examine the functionality of the new software image. When you are satisfied that the new image is acceptable, enter the **issu commitversion** command to complete the upgrade process.

Commitversion Stage

To apply the upgrade image to the second chassis, completing the eFSU, enter the **issu commitversion** command. The VSS standby chassis is reloaded and booted with the new upgrade image, initializing again as the VSS standby chassis. As in the loadversion stage, the throughput of the VSS is temporarily reduced while the modules are reloaded and initialized. After the successful reload and reboot of the VSS standby chassis, the VSS upgrade process is complete.

Abortversion (Optional)

At any time before you enter the **issu commitversion** command, you can roll back the upgrade by entering the **issu abortversion** command. The upgrade process also aborts automatically if the software detects a failure. The rollback process depends on the current state. If the eFSU is aborted before you enter the **issu runversion** command, the VSS standby chassis is reloaded with the old image. If the eFSU is aborted after the **issu runversion** command, a switchover is executed. The VSS standby chassis, running the old image, becomes the VSS active chassis. The formerly VSS active chassis is then reloaded with the old image, completing the rollback.

Configuring and Performing an eFSU Upgrade

The following sections describe how to configure and perform an eFSU upgrade:

- Changing the eFSU Rollback Timer, page 4-59
- Performing an eFSU Upgrade, page 4-59
- Performing an eFSU Upgrade from Previous Cisco IOS Releases to Cisco IOS Release 12.2(33)SXI4, page 4-60
- Performing an eFSU Upgrade from Cisco IOS Release 12.2(33)SXI4 to Future Cisco IOS Releases, page 4-61
- Performing an eFSU Downgrade from Cisco IOS Release 12.2(33)SXI4 to Earlier Cisco IOS Releases, page 4-62
- Performing an eFSU Downgrade from a Future Cisco IOS Release to Cisco IOS Release 12.2(33)SXI4, page 4-64
- Performing an eFSU Upgrade on an Installed Modular Image, page 4-65
- Aborting an eFSU Upgrade, page 4-66

Changing the eFSU Rollback Timer

To view or change the eFSU rollback timer, perform the following task before beginning an upgrade:

	Command	Purpose
Step 1	`Router# config terminal`	Enters configuration mode.
Step 2	`Router(config)# issu set rollback-timer {seconds | hh:mm:ss}`	(Optional) Sets the rollback timer to ensure that the upgrade process does not leave the VSS nonoperational. If the timer expires, the software image reverts to the previous software image. To stop the timer, you must either accept or commit the new software image. The timer duration can be set with one number (*seconds*), indicating the number of seconds, or as hours, minutes, and seconds with a colon as the delimiter (*hh:mm:ss*). The range is 0 to 7200 seconds (2 hours); the default is 2700 seconds (45 minutes). A setting of 0 disables the rollback timer.
Step 3	`Router(config)# exit`	Returns to privileged EXEC mode.
Step 4	`Router# show issu rollback timer`	Displays the current rollback timer value.

This example shows how to set the eFSU rollback timer to one hour using both command formats:

```
Router# config terminal
Router(config)# issu set rollback-timer 3600
% Rollback timer value set to [ 3600 ] seconds
Router(config)# issu set rollback-timer 01:00:00
% Rollback timer value set to [ 3600 ] seconds
Router(config)#
```

Performing an eFSU Upgrade

To perform an eFSU upgrade (or downgrade) of a VSS, perform this task:

	Command	Purpose
Step 1	`Router# copy tftp disk_name`	Uses TFTP to copy the new software image to flash memory on the VSS active and VSS standby chassis (disk0: and slavedisk0:) and to the ICS's, if they exist. Answer the prompts to identify the name and location of the new software image.
Step 2	`Router# show issu state [switch/slot] [detail]`	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	`Router# issu loadversion [active_switch/slot] active-image [standby_switch/slot] standby-image`	Starts the upgrade process by loading the new software image onto the VSS standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the VSS standby chassis to transition to SSO mode.

	Command	Purpose
Step 4	`Router# issu runversion`	Forces a switchover, causing the VSS standby chassis to become VSS active and begin running the new software. The previously VSS active chassis becomes VSS standby and boots with the old image.
Step 5	`Router# issu acceptversion`	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.
Step 6	`Router# issu commitversion`	Loads the new software image onto the VSS standby chassis.
Step 7	`Router# show issu state [switch/slot][detail]`	Verifies the status of the upgrade process. If the upgrade was successful, both the VSS active and VSS standby chassis are running the new software version.

For an example of the eFSU upgrade sequence, see the "eFSU Upgrade Example" section on page 4-66.

Performing an eFSU Upgrade from Previous Cisco IOS Releases to Cisco IOS Release 12.2(33)SXI4

With previous Cisco IOS releases if you have a second ICS in your chassis, it will be forced to ROMMON.

To perform an eFSU upgrade of a VSS from Cisco IOS Release 12.2(33)SXI to Cisco IOS Release 12.(33)SXI4, perform this task:

	Command	Purpose
Step 1	`Router# copy tftp disk_name`	Uses TFTP to copy the new software image to flash memory on the active and standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 2	`Router# show issu state [switch/slot] [detail]`	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	`Router# issu loadversion [active_switch/slot] active-image [standby_switch/slot] standby-image`	Starts the upgrade process by loading the new software image onto the standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode.
Step 4	`Router# issu runversion`	Forces a switchover, causing the standby chassis to become active and begin running the new software. The previously active chassis becomes standby and boots with the old image.
Step 5	`Router# issu acceptversion`	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.

	Command	Purpose
Step 6	`Router#` **`issu commitversion`**	Loads the new software image onto the standby chassis.
Step 7	`Router#` **`show issu state`** `[`*`switch/slot`*`][`**`detail`**`]`	Verifies the status of the upgrade process. If the upgrade was successful, both the active and standby chassis are running the new software version.

If you intend to bring up the ICS supervisor engine with Cisco IOS Release 12.2(33)SXI4, you will need to manually boot up the ICS supervisor engine after the eFSU cycle is complete.

Performing an eFSU Upgrade from Cisco IOS Release 12.2(33)SXI4 to Future Cisco IOS Releases

To perform an eFSU upgrade of a VSS from Cisco IOS Release 12.2(33)SXI4 to a future Cisco IOS Release, perform this task:

	Command	Purpose
Step 1	`Router#` **`copy tftp`** *`disk_name`*	Uses TFTP to copy the new software image to flash memory on the active and standby chassis (disk0: and slavedisk0:) and to the ICSs, if they exist. Answer the prompts to identify the name and location of the new software image.
Step 2	`Router#` **`show issu state`** `[`*`switch/slot`*`] [`**`detail`**`]`	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	`Router#` **`switch virtual in-chassis standby`** *`switch`* `[`**`disable`** `|` **`enable`**`]`	(Optional) Includes or removes the ICS, from the eFSU cycle. This command must be executed before the start of the eFSU cycle even if the ICS is in ROMMON.
Step 4	`Router#` **`issu loadversion`** `[`*`active_switch/slot`*`]` *`active-image`* `[`*`standby_switch/slot`*`]` *`standby-image`*	Starts the upgrade process by loading the new software image onto the standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode. **Note** This command is not extended for the ICS. The file system mentioned for the ICA is used for the respective ICS. For example, if the **issu loadversion disk0:***image_name* **slavesup-bootdisk:***image_name* command is executed the loadversion command is accepted. The presence of the image is checked in the disk0: for the active supervisor engine (both the ICA and ICS) and the SPs bootdisk for the SSO standby (both the ICA and ICS).

	Command	Purpose
Step 5	`Router# issu runversion`	Forces a switchover, causing the standby engine chassis to become active and begin running the new software. The previously active chassis becomes standby and boots with the old image. **Note** If there are two supervisor engines in the active chassis, an in-chassis role reversal will occur if the upgrade cycle starts with both supervisor engines in the active chassis unless you have configured a supervisor engine to not participate in the upgrade.
Step 6	`Router# issu acceptversion`	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.
Step 7	`Router# issu commitversion`	Loads the new software image onto the standby chassis.
Step 8	`Router# show issu state [switch/slot][detail]`	Verifies the status of the upgrade process. If the upgrade was successful, both the active and standby chassis are running the new software version. The ICS is forced to ROMMON.

If the ICS is participating in the eFSU upgrade, you must ensure that the ICS is up and running before performing each ISSU step. If the ICS is not up and running, you need to wait until it is online. You can verify that the ICS is online by entering the **show module** command.

Performing an eFSU Downgrade from Cisco IOS Release 12.2(33)SXI4 to Earlier Cisco IOS Releases

To perform an eFSU downgrade of a VSS from Cisco IOS Release 12.2(33)SXI4 to an earlier Cisco IOS release, perform this task:

	Command	Purpose
Step 1	`Router# copy tftp disk_name`	Uses TFTP to copy the new software image to flash memory on the active and standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 2	`Router# show issu state [switch/slot] [detail]`	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	`Router# switch virtual in-chassis standby switch [disable	enable]`	(Optional) Includes or removes the ICS from the eFSU cycle. This command must be executed before the start of the eFSU cycle even if the ICS is in ROMMON.

	Command	Purpose
Step 4	`Router# `**`issu loadversion`**` [`*`active_switch/slot`*`] `*`active-image`*` [`*`standby_switch/slot`*`] `*`standby-image`*	Starts the downgrade process by loading the new software image onto the standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode. **Note** If the active ICS is online when you enter the **issu loadversion** command, then an error message is displayed when the standby supervisor engine is booting up with the pre-12.2(33)SXI4 image, which prompts you to disable the active ICS. Once you disable the active ICS, the cycle will proceed. If you do not disable the active ICS and enter the **issu runversion** command, the command is not accepted. You will have to either abort the downgrade process or disable the active ICS to proceed with the downgrade. If the standby ICS is online when you enter the **issu loadversion** command, the pre-12.2(33)SXI4 image that comes up on the SSO standby forces the standby ICS to ROMMON.
Step 5	`Router# `**`issu runversion`**	Forces a switchover, causing the standby chassis to become active and begin running the new software. The previously active chassis becomes standby and boots with the old image.
Step 6	`Router# `**`issu acceptversion`**	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.
Step 7	`Router# `**`issu commitversion`**	Loads the new software image onto the standby chassis.
Step 8	`Router# `**`show issu state`**` [`*`switch/slot`*`][`**`detail`**`]`	Verifies the status of the downgrade process. If the downgrade was successful, both the active and standby chassis are running the new software version.

Performing an eFSU Downgrade from a Future Cisco IOS Release to Cisco IOS Release 12.2(33)SXI4

To perform an eFSU downgrade of a VSS from a future Cisco IOS Release to Cisco IOS Release 12.2(33)SXI4, perform this task:

	Command	Purpose
Step 1	`Router#` **`copy tftp`** *`disk_name`*	Uses TFTP to copy the new software image to the ICSs and flash memory on the active and standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 2	`Router#` **`show issu state`** [*`switch/slot`*] [**`detail`**]	(Optional) Verifies that the VSS is ready to run the eFSU. **Note** You can use the **show issu state** command at any stage of the upgrade to verify the progress and status of the upgrade.
Step 3	`Router#` **`switch virtual in-chassis standby`** *`switch`* [**`disable`** | **`enable`**]	(Optional) Includes or removes the ICS from the eFSU cycle. This command must be executed before the start of the eFSU cycle even if the ICS is in ROMMON. **Note** If you did not remove the ICS from the downgrade using the switch virtual disable command the loadversion cycle is aborted, the SSO standby reloads with the initial image.
Step 4	`Router#` **`issu loadversion`** [*`active_switch/slot`*] *`active-image`* [*`standby_switch/slot`*] *`standby-image`*	Starts the downgrade process by loading the new software image onto the standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode. **Note** This command is not extended for the ICS. The file system mentioned for the ICA is used for the respective ICS. For example, if the **issu loadversion disk0:***image_name* **slavesup-bootdisk:***image_name* command is executed the loadversion command is accepted. The presence of the image is checked in the disk0: for the active supervisor engine (both the ICA and ICS) and the SPs bootdisk for the SSO standby (both the ICA and ICS).
Step 5	`Router#` **`issu runversion`**	Forces a switchover, causing the standby chassis to become active and begin running the new software. The previously active chassis becomes standby and boots with the old image.
Step 6	`Router#` **`issu acceptversion`**	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.

	Command	Purpose
Step 7	Router# **issu commitversion**	Loads the new software image onto the standby chassis.
Step 8	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the downgrade process. If the downgrade was successful, both the active and standby chassis are running the new software version.

If the ICS is participating in the eFSU upgrade, you must ensure that the ICS is up and running before performing each ISSU step. If the ICS is not up and running you need to wait until it is online. You can verify that the ICS is online by entering the **show module** command.

Performing an eFSU Upgrade on an Installed Modular Image

To perform an eFSU upgrade (or downgrade) of an ION VSS, perform this task:

	Command	Purpose
Step 1	Router# **copy tftp** *disk_name*	Uses TFTP to copy the new software image to flash memory on the active and VSS standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image. **Note** You should have a console on both the active and VSS standby supervisor engines because you will go back and forth between them.
Step 2	Router# **install file bootdisk:***filename* **bootdisk:***/location*	Installs the modular image on to both the active and VSS standby supervisor engines.
Step 3	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the upgrade process; status should display 'Init'.
Step 4	Router# **issu loadversion** *new-image*	Starts the upgrade process by loading the installed software image onto the active and VSS standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the VSS standby chassis to transition to SSO mode. **Note** This command will cause the VSS standby chassis to reload.
Step 5	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the upgrade process; status should display 'Load Version'.
Step 6	Router# **issu runversion**	Forces a switchover, causing the VSS standby chassis to become active and begin running the new software. The previously active chassis becomes VSS standby and boots with the old image.
Step 7	Router# **show issu state** [*switch/slot*][**detail**]	Verifies the status of the upgrade process; status should display 'Run Version'.

	Command	Purpose
Step 8	`Router# issu commitversion`	Loads the new software image onto the VSS standby chassis. **Note** This command will cause the VSS standby chassis to reload.
Step 9	`Router# show issu state [switch/slot][detail]`	Verifies the status of the upgrade process; status should display 'Init'.
Step 10	`Router# redundancy force-switchover`	(Optional) Forces the VSS standby Route Processor (RP) to assume the role of the active RP.

For an example of the eFSU upgrade on an Installed Modular Image sequence, see the "eFSU Upgrade on an Installed Modular Image Example" section on page 4-67.

Aborting an eFSU Upgrade

To manually abort the eFSU and roll back the upgrade, perform this task:

Command	Purpose
`Router# issu abortversion`	Stops the upgrade process and forces a rollback to the previous software image.

This example shows how to abort an eFSU upgrade for a VSS:

```
Router# issu abortversion
```

eFSU Upgrade Example

This example shows how to perform and verify an eFSU upgrade for a VSS.

Verify the System Readiness

After copying the new image files into the file systems of the active and VSS standby chassis, enter the **show issu state detail** command and the **show redundancy status** command to check that the VSS is ready to perform the eFSU. One chassis must be in the active state and the other chassis in the hot VSS standby state. Both chassis should be in the ISSU Init state and in SSO redundancy state. In the example, both chassis are running an "oldversion" image.

```
Router# show issu state detail
                          Slot = 1/2
                      RP State = Active
                    ISSU State = Init
                 Boot Variable = disk0:s72033-oldversion.v1,12;
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:s72033-oldversion.v1
                Variable Store = PrstVbl

                          Slot = 2/7
                      RP State = Standby
                    ISSU State = Init
```

```
                 Boot Variable = disk0:s72033-oldversion.v1,12;
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:s72033-oldversion.v1

Router# show redundancy status
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Secondary
        Unit ID = 18

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
 Communications = Up

   client count = 132
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 18
           RF debug mask = 0x0
```

eFSU Upgrade on an Installed Modular Image Example

This example shows how to perform an an eFSU upgrade on an installed modular image:

```
Router# copy ftp://172.18.108.26/s72033-advipservicesk9_wan-vz.122-33.SXI2
sup-bootdisk:s72033-advipservicesk9_wan-vz.122-33.SXI2.bin
Router# copy ftp://172.18.108.26/s72033-advipservicesk9_wan-vz.122-33.SXI2
slavesup-bootdisk:s72033-advipservicesk9_wan-vz.122-33.SXI2.bin
Router# install file sup-bootdisk:s72033-advipservicesk9_wan-vz.122-33.SXI2.bin sup-bootdisk:/newsys
Router# install file slavesup-bootdisk:s72033-advipservicesk9_wan-vz.122-33.SXI2.bin
slavesup-bootdisk:/newsys
Router# show issu state
                      Slot = 1/6
                  RP State = Active
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;

                      Slot = 2/6
                  RP State = Standby
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;
Router# issu loadversion sup-bootdisk:/newsys/s72033/base/s72033-advipservicesk9_wan-vm
%issu loadversion executed successfully, Standby is being reloaded
Router# show issu state
                      Slot = 1/6
                  RP State = Active
                ISSU State = Load Version
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;

                      Slot = 2/6
                  RP State = Standby
                ISSU State = Load Version
             Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;
Router# issu runversion
This command will reload the Active unit. Proceed ? [confirm]
Router# show issu state
                      Slot = 2/6
```

```
                 RP State = Active
               ISSU State = Run Version
            Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;

                     Slot = 1/6
                 RP State = Standby
               ISSU State = Run Version
            Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;
Router# issu commitversion
%issu commitversion executed successfully
Router# show issu state
                     Slot = 2/6
                 RP State = Active
               ISSU State = Init
            Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;

                     Slot = 1/6
                 RP State = Standby
               ISSU State = Init
            Boot Variable = bootdisk:/sys/s72033/base/s72033-advipservicesk9_wan-vm,12;
Router# redundancy force-switchover
```

Load the New Image to the VSS Standby Chassis

Enter the **issu loadversion** command to start the upgrade process. In this step, the VSS standby chassis reboots, reloads with the new image, and initializes as the VSS standby chassis in SSO redundancy mode, running the new image. This step is complete when the chassis configuration is synchronized, as indicated by the “Bulk sync succeeded” message.

```
Router# issu loadversion disk0:s72033-newversion.v2

000133: Aug  6 16:17:44.486 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/2/4, changed state to down
000134: Aug  6 16:17:43.507 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet2/7/4, changed state to down
000135: Aug  6 16:17:43.563 PST: %LINK-3-UPDOWN: Interface TenGigabitEthernet2/7/4,
changed state to down
000136: Aug  6 16:17:44.919 PST: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/2/4,
changed state to down
```

(Deleted many interface and protocol down messages)

```
%issu loadversion executed successfully, Standby is being reloaded
```

(Deleted many interface and protocol down messages, then interface and protocol up messages)

```
0000148: Aug  6 16:27:54.154 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/2/5, changed state to up
000149: Aug  6 16:27:54.174 PST: %LINK-3-UPDOWN: Interface TenGigabitEthernet2/7/5,
changed state to up
000150: Aug  6 16:27:54.186 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet2/7/5, changed state to up
000151: Aug  6 16:32:58.030 PST: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync
succeeded
```

Verify the New Image on the VSS Standby Chassis

You can now enter the **show issu state detail** command and the **show redundancy** command to check that both chassis are in the ISSU Load Version state and SSO redundancy state. In this example, the VSS standby chassis is now running the "newversion" image.

```
Router# show issu state detail
                          Slot = 1/2
                      RP State = Active
                    ISSU State = Load Version
                 Boot Variable = disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = disk0:s72033-oldversion.v1
             Secondary Version = disk0:s72033-newversion.v2
               Current Version = disk0:s72033-oldversion.v1
                Variable Store = PrstVbl

                          Slot = 2/7
                      RP State = Standby
                    ISSU State = Load Version
                 Boot Variable =
disk0:s72033-newversion.v2,12;disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = disk0:s72033-oldversion.v1
             Secondary Version = disk0:s72033-newversion.v2
               Current Version = disk0:s72033-newversion.v2

Router# show redundancy status
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Secondary
        Unit ID = 18

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
 Communications = Up

   client count = 132
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

Execute a Switchover to the New Image

When the VSS standby chassis is successfully running the new image in SSO redundancy state, enter the **issu runversion** command to force a switchover. The upgraded VSS standby chassis takes over as the new active chassis, running the new image. The formerly active chassis reloads and initializes as the new VSS standby chassis in SSO mode, running the old image (in case the software upgrade needs to be aborted and the old image restored). This step is complete when the chassis configuration is synchronized, as indicated by the "Bulk sync succeeded" message.

```
Router# issu runversion
This command will reload the Active unit.  Proceed ? [confirm]
System Bootstrap, Version 12.2(17r)S4, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2005 by cisco Systems, Inc.
```

```
Cat6k-Sup720/RP platform with 1048576 Kbytes of main memory

Download Start
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
(Deleted many lines)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Download Completed! Booting the image.
Self decompressing the image :
##########################################################################################
(Deleted many lines)
################################################################################ [OK]
running startup....

(Deleted many lines)

000147: Aug  6 16:53:43.199 PST: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync
succeeded
```

Verify the Switchover

You can now enter the **show issu state detail** command and the **show redundancy** command to check that both chassis are in the ISSU Run Version state and SSO redundancy state. In this example, the active chassis is now running the “newversion” image.

```
Router# show issu state detail
                          Slot = 2/7
                      RP State = Active
                    ISSU State = Run Version
                 Boot Variable =
disk0:s72033-newversion.v2,12;disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = disk0:s72033-newversion.v2
             Secondary Version = disk0:s72033-oldversion.v1
               Current Version = disk0:s72033-newversion.v2
                Variable Store = PrstVbl

                          Slot = 1/2
                      RP State = Standby
                    ISSU State = Run Version
                 Boot Variable = disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = disk0:s72033-newversion.v2
             Secondary Version = disk0:s72033-oldversion.v1
               Current Version = disk0:s72033-oldversion.v1

Router# show redundancy status
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 39

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
 Communications = Up

   client count = 134
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
```

```
         keep_alive count = 1
     keep_alive threshold = 18
            RF debug mask = 0x0
```

Commit the New Image to the VSS Standby Chassis

When the active chassis is successfully running the new image in the SSO redundancy state, you can enter either the **issu acceptversion** command to stop the rollback timer and hold this state indefinitely, or the **issu commitversion** command to continue with the eFSU. To continue, enter the **issu commitversion** command to upgrade the VSS standby chassis and complete the eFSU sequence. The VSS standby chassis reboots, reloads with the new image, and initializes as the VSS standby chassis in the SSO redundancy state, running the new image. This step is complete when the chassis configuration is synchronized, as indicated by the "Bulk sync succeeded" message.

```
Router# issu commitversion
Building configuration...
[OK]
000148: Aug  6 17:17:28.267 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet2/7/4, changed state to down
000149: Aug  6 17:17:28.287 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/2/4, changed state to down
```

(Deleted many interface and protocol down messages)

```
%issu commitversion executed successfully
```

(Deleted many interface and protocol down messages, then interface and protocol up messages)

```
000181: Aug  6 17:41:51.086 PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/2/5, changed state to up
000182: Aug  6 17:42:52.290 PST: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync
succeeded
```

Verify That the Upgrade is Complete

You can now enter the **show issu state detail** command and the **show redundancy** command to check the results of the eFSU. In this example, both chassis are now running the "newversion" image, indicating that the eFSU was successful. Because the eFSU has completed, the two chassis will be once again in the ISSU Init Version state, as they were before the eFSU was initiated.

```
Router# show issu state detail
                          Slot = 2/7
                      RP State = Active
                    ISSU State = Init
                 Boot Variable =
disk0:s72033-newversion.v2,12;disk0:s72033-oldversion.v1,12
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:s72033-newversion.v2
                Variable Store = PrstVbl

                          Slot = 1/2
                      RP State = Standby
                    ISSU State = Init
                 Boot Variable =
disk0:s72033-newversion.v2,12;disk0:s72033-oldversion.v1,12
```

```
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:s72033-newversion.v2

Router# show redundancy status
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 39

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
 Communications = Up

   client count = 134
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

Tip For additional information about Cisco Catalyst 6500 series switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

PART 2

High Availability

Performing an Enhanced Fast Software Upgrade

This chapter provides information about how to perform a software upgrade using the Enhanced Fast Software Upgrade (eFSU) feature.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

---

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains the following sections:

eFSU Overview

The following sections provide an overview of how eFSU works:

Note eFSU is supported in VSS mode. See the "VSS Configuration Guidelines and Restrictions" section on page 4-27 for more information.

eFSU Operation

eFSU is an enhanced software upgrade procedure. Cisco IOS Release 12.2(33)SXI and later releases support eFSU.

Non-eFSU (FSU) software upgrades require system downtime, because a software version mismatch between the active and the standby supervisor engines forces the system to boot in RPR redundancy mode, which is stateless and causes a hard reset of the all modules.

eFSU enables an increase in network availability by reducing the downtime caused by software upgrades. eFSU does this by:

- Bringing up the standby supervisor engine in SSO mode even when the active and the standby supervisor engines have different software versions, or with VSS configured, when the supervisor engines in the two chassis have different software versions.

  During an eFSU, new software is loaded onto the standby supervisor engine while the active supervisor engine continues to operate using the previous software. As part of the upgrade, the standby processor reaches the SSO Standby Hot stage, a switchover occurs, and the standby becomes active, running the new software. In previous releases Supervisor Engines running different software versions ran in the Route Processor Redundancy Mode.

  You can continue with the upgrade to load the new software onto the other processor, or you can abort the upgrade and resume operation with the old software.

- Preloading new module software into memory on supported modules to avoid a hard reset.

  If the new software release contains new module software, eFSU preloads the new module software onto any modules in the switch that support eFSU preload. When the switchover occurs between the active and standby supervisor engines, the modules are restarted with the new software image.

  The following modules support eFSU preload:

  – WS-X67*xx* modules
  – SIP-400 and SIP-600

  All other modules undergo a hard reset at switchover, and the software image loads after the module restarts.

  During a software upgrade, the switch performs the following steps automatically on modules that support eFSU preload:

  – Reserves the necessary memory for the new Cisco IOS software image on each module.
  – Preloads a new software image onto the modules as part of the **issu loadversion** command.
  – Restarts the modules with the new software image when a switchover occurs (**issu runversion**).
  – During the restart, the software features and routing protocols are not available.
  – If a rollback or abort occurs, to minimize disruption, the switch preloads the original software version onto the module. Once the rollback or abort is completed, the module is restarted with the original software version.

Note All modules that support eFSU preload must have at least 512 MB of memory, with enough memory free to hold the new software image. If there is insufficient free memory, eFSU does not attempt the preload, but instead resets the modules during the switchover.

Outage Time and Support Considerations

During an eFSU upgrade, modules are restarted or reset after the switchover that occurs between the supervisor engines. Because the modules are restarted or reset, any links attached to the modules go up and down and traffic processing is disrupted until protocols and software features are brought back online. The length of time that module processing is disrupted (outage time) depends on whether the eFSU process was able to preload a new software image onto the module.

- For modules that support eFSU preload, the outage time for an eFSU module warm reload is faster than an RPR mode module reload.
- For modules that do not support eFSU preload, the outage time for module reload is similar to an RPR mode module reload.

Once the new software is loaded (**issu loadversion**), you can use the **show issu outage slot all** command to display the maximum outage time for installed modules. See the "Displaying the Maximum Outage Time for Installed Modules (Optional)" section on page 5-10 for a command example.

Reserving Module Memory

On modules that support eFSU, the supervisor engine automatically reserves memory on the module to store the new software image (decompressed format). The amount of memory needed varies according to the module type.

Although we do not recommend it, you can enter the following command to keep the switch from reserving memory for the software preload (where *slot-num* specifies which slot the module is installed in):

no mdr download reserve memory image slot *slot-num*

Note All modules that support eFSU preload must have at least 512 MB of memory, with enough memory free to hold the new software image. If there is insufficient free memory, eFSU does not attempt the preload, but instead resets the modules during the switchover.

To display whether or not the memory reservation was successful on a module, use the **show issu outage slot all** command See the "Displaying the Maximum Outage Time for Installed Modules (Optional)" section on page 5-10 for a command example.

Error Handling for eFSU Preload

If problems occur during eFSU preload, the switch takes the following actions:

- Module crash during loadversion—The module is reset when a switchover occurs.
- Module not active when eFSU started—No power is provided to the module during the software upgrade, and the module is reset when the process ends. The same action is applied to a module that is inserted into the switch after the software upgrade process has begun.
- Module crash during run version or during rollback—The module boots with the software image version that corresponds to the software image that is present on the active supervisor engine.

eFSU Restrictions

- eFSU requires two supervisor engines, one active and one standby.
- Both the active and standby supervisor engines must have enough flash memory to store both the old and new software images prior to the upgrade process.
- Images from different features sets, regardless of release, fail the eFSU compatibility check.
- When downgrading with eFSU to an earlier version of Cisco IOS Software, the configuration files fail to synchronize and the standby supervisor engine reloads unless you disable any features or functions that are not supported in the earlier version before you start the process. Remove any configuration commands that are not available in the earlier version.
- During an eFSU upgrade, the modules are restarted.
- The switch examines the old and new software images and automatically performs the appropriate process (eFSU) to upgrade the software image:
  - For a patch upgrade, if the module software is the same in both the old and the new software images, because no module software upgfrade is required, the eFSU upgrades only the supervisor engine software. The system downtime is from 0 to 3 seconds.
  - If the module software in the images is different, the modules are restarted or reset during the upgrade process. System downtime depends on whether the modules support eFSU (see the "Outage Time and Support Considerations" section on page 5-3 for more information).
- The eFSU upgrade feature works with NSF/SSO. Software features that do not support NSF/SSO stop operating until they come back online after the switchover that occurs during the software upgrade.
- Release 12.2(33)SXJ and later releases support NTPv4. Earlier releases support NTPv3. With NTPv3, be aware of CSCec87418, which can cause eFSU to fail.
- All modules that support eFSU preload must have at least 512 MB of memory, with enough memory free to hold the new software image. If there is insufficient free memory, eFSU does not attempt the preload, but instead resets the modules during the switchover.
- Online insertion and replacement (OIR) is not supported during an eFSU. If you attempt to insert a new module in the switch while the upgrade is active, the switch does not provide power for the module. When the upgrade ends, the switch resets the newly inserted module.
- Do not perform a manual switchover between supervisor engines during the upgrade.
- Make sure that the configuration register is set to allow autoboot (the lowest byte of the register should be set to 2).

- Before you enter the **issu abortversion** command (to abort a software upgrade), make sure that the standby supervisor engine is Up (STANDBY HOT [in SSO] or COLD [in RPR]).
- The Fast Software Upgrade (FSU) process supports upgrade from earlier releases to Release 12.2(33)SXI or later releases. During this process, the module software image is also upgraded on those modules that support eFSU.
- The enhanced Fast Software Upgrade (eFSU) process supports upgrades from Release 12.2(33)SXI and rebuilds to Release 12.2(33)SXJ. During this process, the module software image is also upgraded on those modules that support eFSU.

Performing an Enhanced Fast Software Upgrade

The following sections describe the process for performing an enhanced fast software upgrade (eFSU) on the Catalyst 6500 series switch:

- Software Upgrade Process Summary For a Switch, page 5-5
- Preparing for the Upgrade, page 5-6
- Copying the New Software Image, page 5-8
- Loading the New Software onto the Standby Supervisor Engine, page 5-8
- Displaying the Maximum Outage Time for Installed Modules (Optional), page 5-10
- Forcing a Switchover from Active to Standby, page 5-10
- Accepting the New Software Version and Stopping the Rollback Process (Optional), page 5-11
- Committing the New Software to the Standby, page 5-12
- Verifying the Software Installation, page 5-12
- Aborting the Upgrade Process, page 5-13

Each section briefly describes a particular step in the upgrade process and provides command examples. In the command examples, important fields in the command output are shown in bold. Check these fields to verify the status of the command.

Software Upgrade Process Summary For a Switch

The following sections provide examples of the software upgrade process. To upgrade the software, perform the following tasks:

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router#` **`copy tftp`** *`disk_name`*	Uses TFTP to copy the new software image to flash memory on the active and standby supervisor engines (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.

	Command	Purpose

Step 3	`Router# show version | in image`
	`Router# show bootvar`
	
	`Router# show redundancy`
	`Router# show issu state [detail]`
		These **show** commands verify that the switch is ready to run eFSU. The **show version** and **show bootvar** commands verify the boot image settings.
		The **show redundancy** and **show issu state** commands verify that redundancy mode is enabled and that SSO and NSF are configured.
		Note Use the **show redundancy** and **show issu state** commands throughout the upgrade to verify the status of the upgrade.

Step 4	`Router# issu loadversion` *`active-slot active-image standby-slot standby-image`*	Starts the upgrade process and loads the new software image onto the standby supervisor engine. It may take several seconds for the new image to load and for the standby supervisor engine to transition to SSO mode.
Step 5	`Router# show issu outage slot all`	(Optional) Displays the maximum outage time for installed modules. Enter the command on the switch processor of the supervisor engine.
Step 6	`Router# issu runversion`	Forces a switchover, which causes the standby supervisor engine to become active and begin running the new software. The previously active processor becomes standby and boots with the old image.
Step 7	`Router# issu acceptversion`	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.
Step 8	`Router# issu commitversion`	Loads the new software image onto the standby supervisor engine in the specified slot.
Step 9	`Router# show redundancy` `Router# show issu state [detail]`	Verifies the status of the upgrade process. If the upgrade was successful, both the active and standby supervisor engines are running the new software version.

Preparing for the Upgrade

Before attempting to perform a software upgrade, be sure to review the "eFSU Restrictions" section on page 5-4.

To prepare for eFSU, perform the tasks in the following sections:

- Verifying the Boot Image Version and Boot Variable, page 5-6
- Verifying Redundancy Mode, page 5-7
- Verifying eFSU State, page 5-8

Verifying the Boot Image Version and Boot Variable

Before starting, enter the **show version** and **show bootvar** commands to verify the boot image version and BOOT environment variable, as shown in the following examples:

```
Router# show version | in image
BOOT variable = disk0:image_name;
CONFIG_FILE variable =
BOOTLDR variable =
```

```
Configuration register is 0x2002

Standby is up
Standby has 1048576K/65536K bytes of memory.

Standby BOOT variable = disk0:image_name;
Standby CONFIG_FILE variable =
Standby BOOTLDR variable =
```

Verifying Redundancy Mode

Verify that redundancy mode is enabled and that NSF and SSO are configured. The following command example shows how to verify redundancy:

```
Router# show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 45 minutes
Switchovers system experienced = 0
              Standby failures = 0
        Last switchover reason = none

                 Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 6
        Current Software state = ACTIVE
       Uptime in current state = 44 minutes
                 Image Version = Cisco IOS Software, image_details
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Wed 18-Feb-09 12:48 by kchristi
                          BOOT = disk0:image_name;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2002

Peer Processor Information :
----------------------------
              Standby Location = slot 5
        Current Software state = STANDBY HOT
       Uptime in current state = 28 minutes
                 Image Version = Cisco IOS Software, image_details
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled image_details
                          BOOT = disk0:image_name ;
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2002
```

Verifying eFSU State

Verify that the the ISSU state is **Init**, rather than an intermediate eFSU upgrade state. Enter this command:

```
Router# show issu state detail
                          Slot = 6
                      RP State = Active
                    ISSU State = Load Version
                 Boot Variable = disk0:image_name
                Operating Mode = sso
               Primary Version = disk0:sierra.0217
             Secondary Version = disk0:sierra.0217
               Current Version = disk0:sierra.0217
                Variable Store = PrstVbl
                     ROMMON CV = [disk0:image_name]

                          Slot = 5
                      RP State = Standby
                    ISSU State = Load Version
                 Boot Variable = disk0:image_name
                Operating Mode = sso
               Primary Version = disk0:image_name
             Secondary Version = disk0:image_name
               Current Version = disk0:image_name
```

Copying the New Software Image

Before starting the eFSU process, copy the new software image to flash memory (disk0: and slavedisk0:) on the active and standby supervisor engines.

Loading the New Software onto the Standby Supervisor Engine

Enter the **issu loadversion** command to start the upgrade process. This command reboots the standby supervisor engine and loads the new software image onto the standby supervisor engine. When the download is complete, you are prompted to enter the **runversion** command.

Note Do not automatically disable the features that are not common to both images. During the standby initialization, after you enter the **issu loadversion** command, if there are any enabled features that are not supported on the standby supervisor engine, a message is displayed that states that the standby supervisor engine cannot initialize while this feature is enabled, and the standby supervisor engine is forced to RPR (in the load-version state).

```
Router# issu loadversion device:filename
%issu loadversion executed successfully, Standby is being reloaded
```

When execution of the **issu loadversion** command completes, the standby supervisor engine is loaded with the new software image and the supervisor engine is in SSO mode. The **issu loadversion** command might take several seconds to complete. If you enter the **show** commands too soon, you might not see the information that you need.

These examples show how to check the status of the upgrade using the **show redundancy** and **show issu state detail** commands:

```
Router# show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 1 hour, 0 minutes
Switchovers system experienced = 0
              Standby failures = 1
        Last switchover reason = none

                 Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 6
        Current Software state = ACTIVE
       Uptime in current state = 59 minutes
                 Image Version = Cisco IOS Software, image_details
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled ...
                          BOOT = disk0:image_name
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2002

Peer Processor Information :
----------------------------
              Standby Location = slot 5
        Current Software state = STANDBY HOT
       Uptime in current state = 3 minutes
                 Image Version = Cisco IOS Software, image_name
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled ...
                          BOOT = disk0:image_name
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2002

Router# show issu state detail
                          Slot = 6
                      RP State = Active
                    ISSU State = Load Version
                 Boot Variable = disk0:image_name
                Operating Mode = sso
               Primary Version = disk0:image_name
             Secondary Version = disk0:image_name
               Current Version = disk0:image_name
                Variable Store = PrstVbl
                    ROMMON CV = [disk0:image_name]

                          Slot = 5
                      RP State = Standby
                    ISSU State = Load Version
                 Boot Variable = disk0:image_name
                Operating Mode = sso
               Primary Version = disk0:image_name
             Secondary Version = disk0:image_name
               Current Version = disk0:image_name
```

Displaying the Maximum Outage Time for Installed Modules (Optional)

Once the new software is downloaded, you can enter the **show issu outage slot all** command on the switch processor to display the maximum outage time for the installed modules:

```
Router# show issu outage slot all
Slot # Card Type                                   MDR Mode    Max Outage Time
------ ------------------------------------------- ----------- ---------------
     1 CEF720 8 port 10GE with DFC                 WARM_RELOAD        300 secs
     2 96-port 10/100 Mbps RJ45                    RELOAD             360 secs
     4 CEF720 48 port 1000mb SFP                   RELOAD             360 secs

Slot # Reason                                      Error Number
------ ------------------------------------------- ------------
     1 PLATFORM_INIT                                          3
     2 PLATFORM_INIT                                          3
     4 PREDOWNLOAD_LC_MIMIMUM_MEMORY_FAILURE                  5
Router#
```

Forcing a Switchover from Active to Standby

Enter the **issu runversion** command to force a switchover between the active and standby supervisor engines. The standby supervisor engine, which has the new software image loaded, becomes active. The previously active supervisor engine becomes the standby and boots with the old software image (in case the software upgrade needs to be aborted and the old image restored).

```
Router# issu runversion

This command will reload the Active unit.  Proceed ? [confirm] y
```

A switchover between the supervisor engines occurs now. The previous standby supervisor engine becomes active and is running the new software version. The previous active supervisor engine, now the standby supervisor engine, boots with the old software.

Note At this point, the new active supervisor engine is running the new software image and the standby is running the old software image. You should verify the state of the active and standby supervisor engines as shown in the following examples (**show redundancy** and **show issu state detail**).

```
Router# show redundancy
------------------------------
       Available system uptime = 1 hour, 9 minutes
Switchovers system experienced = 1
              Standby failures = 0
        Last switchover reason = user forced

                 Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 5
        Current Software state = ACTIVE
       Uptime in current state = 7 minutes
                 Image Version = Cisco IOS Software, image_details
```

```
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled ...
                          BOOT = disk0:image_name
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2002

Peer Processor Information :
----------------------------
              Standby Location = slot 6
        Current Software state = STANDBY HOT
       Uptime in current state = 0 minutes
                 Image Version = Cisco IOS Software, image_details
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Wed 18-Feb-09 12:48 by kchristi
                          BOOT = disk0:image_name
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2002

Router# show issu state detail
                          Slot = 5
                      RP State = Active
                    ISSU State = Run Version
                 Boot Variable = disk0:image_name
                Operating Mode = sso
               Primary Version = disk0:image_name
             Secondary Version = disk0:image_name
               Current Version = disk0:image_name
                Variable Store = PrstVbl
                     ROMMON CV = [disk0:image_name]

                          Slot = 6
                      RP State = Standby
                    ISSU State = Run Version
                 Boot Variable = disk0:image_name
                Operating Mode = sso
               Primary Version = disk0:image_name
             Secondary Version = disk0:image_name
               Current Version = disk0:image_name
```

Note To complete the upgrade process, enter the **issu acceptversion** (optional) and **issu commitversion** commands (as described in the following sections).

Accepting the New Software Version and Stopping the Rollback Process (Optional)

You must either accept or commit the new software image, or the rollback timer will expire and stop the upgrade process. If that occurs, the software image reverts to the previous software version. The rollback timer is a safeguard to ensure that the upgrade process does not leave the switch nonoperational.

Note New features that are not supported by the previous image are allowed to be enabled only after you enter the **issu commitversion** command.

The following command sequence shows how the **issu acceptversion** command stops the rollback timer to enable you to examine the functionality of the new software image. When you are satisfied that the new image is acceptable, enter the **issu commitversion** command to end the upgrade process.

```
Router# show issu rollback-timer
        Rollback Process State = In progress
      Configured Rollback Time = 00:45:00
       Automatic Rollback Time = 00:37:28

Router# issu acceptversion
% Rollback timer stopped. Please issue the commitversion command.
```

View the rollback timer to see that the rollback process has been stopped:

```
Router# show issu rollback-timer
        Rollback Process State = Not in progress
      Configured Rollback Time = 00:45:00
```

Committing the New Software to the Standby

Enter the **issu commitversion** command to load the new software image onto the standby supervisor engine and complete the software upgrade process. In the following example, the new image is loaded onto the standby supervisor engine in slot 5:

```
Router# issu commitversion
Building configuration...
[OK]
%issu commitversion executed successfully
```

Note The software upgrade process is now complete. Both the active and standby supervisor engines are running the new software version.

Verifying the Software Installation

You should verify the status of the software upgrade. If the upgrade was successful, both the active and standby supervisor engines are running the new software version.

```
Router# show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 1 hour, 17 minutes
Switchovers system experienced = 1
              Standby failures = 1
        Last switchover reason = user forced

                 Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 5
        Current Software state = ACTIVE
       Uptime in current state = 15 minutes
                 Image Version = Cisco IOS Software, image_name
```

```
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled ...
                          BOOT = disk0:image_name
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2002

Peer Processor Information :
----------------------------
              Standby Location = slot 6
        Current Software state = STANDBY HOT
       Uptime in current state = 0 minutes
                 Image Version = Cisco IOS Software, image_details
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled ...
                          BOOT = disk0:image_name
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2002

Router# show issu state detail

                          Slot = 5
                      RP State = Active
                    ISSU State = Init
                 Boot Variable = disk0:image_name
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:image_name
                Variable Store = PrstVbl
                     ROMMON CV = [disk0:simage_name ]

                          Slot = 6
                      RP State = Standby
                    ISSU State = Init
                 Boot Variable = disk0:image_name
                Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:image_name
```

Aborting the Upgrade Process

You can manually abort the software upgrade at any stage by entering the **issu abortversion** command. The upgrade process also aborts on its own if the software detects a failure.

If you abort the process after you enter the **issu loadversion** command, the standby supervisor engine is reset and reloaded with the original software.

The following is an example of the **issu abortversion** *slot image* command that shows how to abort the software upgrade process:

```
Router# issu abortversion 6 c7600s72033
```

Note Before you enter the **issu abortversion** command, make sure that the standby supervisor engine is Up (STANDBY HOT [in SSO] or COLD [in RPR]).

Performing an eFSU Upgrade on an Installed Modular Image

The following sections describe the process for performing an enhanced fast software upgrade (eFSU) on an Installed Modular Image (ION):

- Upgrading an Installed Modular Image, page 5-14
- Example an eFSU Upgrade on an Installed Modular Image, page 5-15

Upgrading an Installed Modular Image

To perform an eFSU upgrade (or downgrade) of an ION VSS, perform this task:

	Command	Purpose
Step 1	`Router#` **`copy tftp`** *`disk_name`*	Uses TFTP to copy the new software image to flash memory on the active and standby chassis (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image. **Note** It is best to have a console on both the active and standby supervisor engines as you will go back and forth between them.
Step 2	`Router#` **`install file bootdisk:`***`filename`* **`bootdisk:`***`/location`*	Installs the modular image onto both the active and standby supervisor engines.
Step 3	`Router#` **`show issu state`** `[`*`switch/slot`*`][`**`detail`**`]`	Verifies the status of the upgrade process; status should display 'Init'.
Step 4	`Router#` **`issu loadversion`** *`new-image`*	Starts the upgrade process by loading the installed software image onto the active and standby chassis. The image name includes the path of the target image to be loaded, in the format *devicename:filename*. It may take several seconds for the new image to load and for the standby chassis to transition to SSO mode. **Note** This command will cause the standby chassis to reload.
Step 5	`Router#` **`show issu state`** `[`*`switch/slot`*`][`**`detail`**`]`	Verifies the status of the upgrade process; status should display 'Load Version'.
Step 6	`Router#` **`issu runversion`**	Forces a switchover, causing the standby chassis to become active, and begins running the new software. The previously active chassis becomes standby and boots with the old image.
Step 7	`Router#` **`show issu state`** `[`*`switch/slot`*`][`**`detail`**`]`	Verifies the status of the upgrade process; status should display 'Run Version'.

	Command	Purpose
Step 8	`Router# issu commitversion`	Loads the new software image onto the standby chassis. **Note** This command will cause the standby chassis to reload.
Step 9	`Router# show issu state [switch/slot][detail]`	Verifies the status of the upgrade process; status should display 'Init'.
Step 10	`Router# redundancy force-switchover`	(Optional) Forces the standby Route Processor (RP) to assume the role of the active RP.

For an example of the eFSU upgrade on an Installed Modular Image sequence, see the "Example an eFSU Upgrade on an Installed Modular Image" section on page 5-15.

Example an eFSU Upgrade on an Installed Modular Image

This example shows how to perform an an eFSU upgrade on an Installed Modular Image.

```
Router# copy ftp://172.18.108.26/image_name sup-bootdisk:simage_name
Router# copy ftp://172.18.108.26/image_name slavesup-bootdisk:image_name
Router# install file sup-bootdisk:image_name sup-bootdisk:/newsys
Router# install file slavesup-bootdisk:image_name slavesup-bootdisk:/newsys
Router# show issu state
                      Slot = 1/6
                  RP State = Active
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/image_name ,12;

                      Slot = 2/6
                  RP State = Standby
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/image_name ,12;
Router# issu loadversion sup-bootdisk:/newsys/s72033/base/image_name
%issu loadversion executed successfully, Standby is being reloaded
Router# show issu state
                      Slot = 1/6
                  RP State = Active
                ISSU State = Load Version
             Boot Variable = bootdisk:/sys/s72033/base/image_name ,12;

                      Slot = 2/6
                  RP State = Standby
                ISSU State = Load Version
             Boot Variable = bootdisk:/sys/s72033/base/image_name ,12;
Router# issu runversion
This command will reload the Active unit. Proceed ? [confirm]
Router# show issu state
                      Slot = 2/6
                  RP State = Active
                ISSU State = Run Version
             Boot Variable = bootdisk:/sys/s72033/base/image_name,12;

                      Slot = 1/6
                  RP State = Standby
                ISSU State = Run Version
             Boot Variable = bootdisk:/sys/s72033/base/image_name,12;
```

```
Router# issu commitversion
%issu commitversion executed successfully
Router# show issu state
                      Slot = 2/6
                  RP State = Active
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/image_name,12;

                      Slot = 1/6
                  RP State = Standby
                ISSU State = Init
             Boot Variable = bootdisk:/sys/s72033/base/image_name,12;
Router# redundancy force-switchover
```

Upgrading a Non-eFSU Image to an eFSU Image

If the new Cisco IOS software image does not support eFSU, you must manually upgrade the software image. To do so, you must upgrade the software image on the standby supervisor engine and then perform a manual switchover so that the standby takes over processing with the new image. You can then upgrade the software image on the previously active, and now standby, supervisor engine. For more information, see the "Software Upgrade Process Summary For a Switch" section on page 5-5.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring NSF with SSO Supervisor Engine Redundancy

This chapter describes how to configure supervisor engine redundancy using Cisco nonstop forwarding (NSF) with stateful switchover (SSO) on Catalyst 6500 series switches.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html
- Cisco ME 6500 Series Ethernet switches do not support redundancy.
- NSF with SSO does not support IPv6 multicast traffic.

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding NSF with SSO Supervisor Engine Redundancy

These sections describe supervisor engine redundancy using NSF with SSO:

- Cisco Express Forwarding, page 6-3
- Multicast MLS NSF with SSO, page 6-4
- Routing Protocols, page 6-4
- NSF Benefits and Restrictions, page 6-8

NSF with SSO Supervisor Engine Redundancy Overview

Note

- The two Gigabit Ethernet interfaces on the redundant supervisor engine are always active unless you enter the **fabric switching-mode allow dcef-only** command.
- With a Supervisor Engine 720, if all the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on both supervisor engines, which ensures that all modules are operating in dCEF mode and simplifies switchover to the redundant supervisor engine. (CSCec05612)

Catalyst 6500 series switches support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. Cisco NSF works with SSO to minimize the amount of time a network is unavailable to its users following a switchover while continuing to forward IP packets. Catalyst 6500 series switches also support route processor redundancy (RPR) for redundancy. For information about these redundancy modes, see Chapter 7, "Configuring RPR Supervisor Engine Redundancy."

The following events cause a switchover:

- A hardware failure on the active supervisor engine
- Clock synchronization failure between supervisor engines
- A manual switchover

SSO Operation

SSO establishes one of the supervisor engines as active while the other supervisor engine is designated as standby, and then SSO synchronizes information between them. A switchover from the active to the redundant supervisor engine occurs when the active supervisor engine fails, or is removed from the switch, or is manually shut down for maintenance. This type of switchover ensures that Layer 2 traffic is not interrupted.

In networking devices running SSO, both supervisor engines must be running the same configuration so that the redundant supervisor engine is always ready to assume control following a fault on the active supervisor engine. SSO switchover also preserves FIB and adjacency entries and can forward Layer 3 traffic after a switchover. Configuration information and data structures are synchronized from the active to the redundant supervisor engine at startup and whenever changes to the active supervisor engine configuration occur. Following an initial synchronization between the two supervisor engines, SSO maintains state information between them, including forwarding information.

During switchover, system control and routing protocol execution is transferred from the active supervisor engine to the redundant supervisor engine. The switch requires between 0 and 3 seconds to switchover from the active to the redundant supervisor engine.

NSF Operation

Cisco NSF always runs with SSO and provides redundancy for Layer 3 traffic. NSF works with SSO to minimize the amount of time that a network is unavailable to its users following a switchover. The main purpose of NSF is to continue forwarding IP packets following a supervisor engine switchover.

Cisco NSF is supported by the BGP, OSPF, EIGRP, and IS-IS protocols for routing and is supported by Cisco Express Forwarding (CEF) for forwarding. The routing protocols have been enhanced with NSF-capability and awareness, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices. The IS-IS protocol can be configured to use state information that has been synchronized between the active and the redundant supervisor engine to recover route information following a switchover instead of information received from peer devices.

A networking device is NSF-aware if it is running NSF-compatible software. A device is NSF-capable if it has been configured to support NSF; it will rebuild routing information from NSF-aware or NSF-capable neighbors.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. After the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF then updates the modules with the new FIB information.

Cisco Express Forwarding

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by Cisco Express Forwarding (CEF). CEF maintains the FIB, and uses the FIB information that was current at the time of the switchover to continue forwarding packets during a switchover. This feature reduces traffic interruption during the switchover.

During normal NSF operation, CEF on the active supervisor engine synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the redundant supervisor engine. Upon switchover of the active supervisor engine, the redundant supervisor engine initially has FIB and adjacency databases that are mirror images of those that were current on the active supervisor engine. For platforms with intelligent modules, the modules will maintain the current forwarding information over a switchover. For platforms with forwarding engines, CEF will keep the forwarding engine on the redundant supervisor engine current with changes that are sent to it by CEF on the active supervisor engine. The modules or forwarding engines will be able to continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates will cause prefix-by-prefix updates to CEF, which it uses to update the FIB and adjacency databases. Existing and new entries will receive the new version (“epoch”) number, indicating that they have been refreshed. The forwarding information is updated on the modules or forwarding engine during convergence. The supervisor engine signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information.

Multicast MLS NSF with SSO

Note NSF with SSO does not support IPv6 multicast traffic. If you configure support for IPv6 multicast traffic, configure RPR redundancy.

Multicast multilayer switching (MMLS) NSF with SSO is required so that Layer 3 multicast traffic that is switched by the router is not dropped during switchover. Without MMLS NSF with SSO, the Layer 3 multicast traffic is dropped until the multicast protocols converge.

During the switchover process, traffic is forwarded using the old database (from the previously active supervisor engine). After multicast routing protocol convergence has taken place, the shortcuts downloaded by the newly active route processor (RP) will be merged with the existing flows and marked as new shortcuts. Stale entries will slowly be purged from the database allowing NSF to function during the switchover while ensuring a smooth transition to the new cache.

Because multicast routing protocols such as Protocol Independent Multicast (PIM) sparse mode and PIM dense mode are data driven, multicast packets are leaked to the router during switchover so that the protocols can converge.

Because the traffic does not need to be forwarded by software for control-driven protocols such as bidirectional PIM, the switch will continue to leak packets using the old cache for these protocols. The router builds the mroute cache and installs the shortcuts in hardware. After the new routes are learned, a timer is triggered to go through the database and purge the old flows.

Note Multicast MLS NSF with SSO requires NSF support in the unicast protocols.

Routing Protocols

The routing protocols run only on the RP of the active supervisor engine, and they receive routing updates from their neighbor routers. Routing protocols do not run on the RP of the redundant supervisor engine. Following a switchover, the routing protocols request that the NSF-aware neighbor devices send state information to help rebuild the routing tables. Alternately, the IS-IS protocol can be configured to synchronize state information from the active to the redundant supervisor engine to help rebuild the routing table on the NSF-capable device in environments where neighbor devices are not NSF-aware. Cisco NSF supports the BGP, OSPF, IS-IS, and EIGRP protocols

Note For NSF operation, the routing protocols depend on CEF to continue forwarding packets while the routing protocols rebuild the routing information.

BGP Operation

When an NSF-capable router begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a statement that the NSF-capable device has “graceful” restart capability. Graceful restart is the mechanism by which BGP routing peers avoid a routing flap following a switchover. If the BGP peer has received this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable router and its BGP peers need to exchange the graceful restart capability in their OPEN messages at the time of session establishment. If both the peers do not exchange the graceful restart capability, the session will not be graceful restart capable.

If the BGP session is lost during the supervisor engine switchover, the NSF-aware BGP peer marks all the routes associated with the NSF-capable router as stale; however, it continues to use these routes to make forwarding decisions for a set period of time. This functionality prevents packets from being lost while the newly active supervisor engine is waiting for convergence of the routing information with the BGP peers.

After a supervisor engine switchover occurs, the NSF-capable router reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable router as having restarted.

At this point, the routing information is exchanged between the two BGP peers. After this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The NSF-aware device uses the network information to remove stale routes from its BGP table; the BGP protocol then is fully converged.

If a BGP peer does not support the graceful restart capability, it will ignore the graceful restart capability in an OPEN message but will establish a BGP session with the NSF-capable device. This function will allow interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers will not be graceful restart capable.

Note BGP support in NSF requires that neighbor networking devices be NSF-aware; that is, the devices must have the graceful restart capability and advertise that capability in their OPEN message during session establishment. If an NSF-capable router discovers that a particular BGP neighbor does not have graceful restart capability, it will not establish an NSF-capable session with that neighbor. All other neighbors that have graceful restart capability will continue to have NSF-capable sessions with this NSF-capable networking device.

OSPF Operation

When an OSPF NSF-capable router performs a supervisor engine switchover, it must perform the following tasks in order to resynchronize its link state database with its OSPF neighbors:

- Relearn the available OSPF neighbors on the network without causing a reset of the neighbor relationship
- Reacquire the contents of the link state database for the network

As quickly as possible after a supervisor engine switchover, the NSF-capable router sends an OSPF NSF signal to neighboring NSF-aware devices. Neighbor networking devices recognize this signal as an indicator that the neighbor relationship with this router should not be reset. As the NSF-capable router receives signals from other routers on the network, it can begin to rebuild its neighbor list.

After neighbor relationships are reestablished, the NSF-capable router begins to resynchronize its database with all of its NSF-aware neighbors. At this point, the routing information is exchanged between the OSPF neighbors. Once this exchange is complete, the NSF-capable device uses the routing information to remove stale routes, update the RIB, and update the FIB with the new forwarding information. The OSPF protocols are then fully converged.

Note OSPF NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable router discovers that it has non-NSF-aware neighbors on a particular network segment, it will disable NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware routers will continue to provide NSF capabilities.

IS-IS Operation

When an IS-IS NSF-capable router performs a supervisor engine switchover, it must perform the following tasks in order to resynchronize its link state database with its IS-IS neighbors:

- Relearn the available IS-IS neighbors on the network without causing a reset of the neighbor relationship
- Reacquire the contents of the link state database for the network

The IS-IS NSF feature offers two options when you configure NSF:

- Internet Engineering Task Force (IETF) IS-IS
- Cisco IS-IS

If neighbor routers on a network segment are running a software version that supports the IETF Internet draft for router restartability, they will assist an IETF NSF router that is restarting. With IETF, neighbor routers provide adjacency and link-state information to help rebuild the routing information following a switchover. A benefit of IETF IS-IS configuration is operation between peer devices based on a proposed standard.

Note If you configure IETF on the networking device, but neighbor routers are not IETF-compatible, NSF will abort following a switchover.

If the neighbor routers on a network segment are not NSF-aware, you must use the Cisco configuration option. The Cisco IS-IS configuration transfers both protocol adjacency and link-state information from the active to the redundant supervisor engine. An advantage of Cisco configuration is that it does not rely on NSF-aware neighbors.

IETF IS-IS Configuration

As quickly as possible after a supervisor engine switchover, the NSF-capable router sends IS-IS NSF restart requests to neighboring NSF-aware devices using the IETF IS-IS configuration. Neighbor networking devices recognize this restart request as an indicator that the neighbor relationship with this router should not be reset, but that they should initiate database resynchronization with the restarting router. As the restarting router receives restart request responses from routers on the network, it can begin to rebuild its neighbor list.

After this exchange is complete, the NSF-capable device uses the link-state information to remove stale routes, update the RIB, and update the FIB with the new forwarding information; IS-IS is then fully converged.

The switchover from one supervisor engine to the other happens within seconds. IS-IS reestablishes its routing table and resynchronizes with the network within a few additional seconds. At this point, IS-IS waits for a specified interval before it will attempt a second NSF restart. During this time, the new redundant supervisor engine will boot up and synchronize its configuration with the active supervisor engine. The IS-IS NSF operation waits for a specified interval to ensure that connections are stable before attempting another restart of IS-IS NSF. This functionality prevents IS-IS from attempting back-to-back NSF restarts with stale information.

Cisco IS-IS Configuration

Using the Cisco configuration option, full adjacency and LSP information is saved, or *checkpointed*, to the redundant supervisor engine. Following a switchover, the newly active supervisor engine maintains its adjacencies using the checkpointed data, and can quickly rebuild its routing tables.

Note Following a switchover, Cisco IS-IS NSF has complete neighbor adjacency and LSP information; however, it must wait for all interfaces to come on line that had adjacencies prior to the switchover. If an interface does not come on line within the allocated interface wait time, the routes learned from these neighbor devices are not considered in routing table recalculation. IS-IS NSF provides a command to extend the wait time for interfaces that, for whatever reason, do not come on line in a timely fashion.

The switchover from one supervisor engine to the other happens within seconds. IS-IS reestablishes its routing table and resynchronizes with the network within a few additional seconds. At this point, IS-IS waits for a specified interval before it will attempt a second NSF restart. During this time, the new redundant supervisor engine will boot up and synchronize its configuration with the active supervisor engine. After this synchronization is completed, IS-IS adjacency and LSP data is check-pointed to the redundant supervisor engine; however, a new NSF restart will not be attempted by IS-IS until the interval time expires. This functionality prevents IS-IS from attempting back-to-back NSF restarts.

EIGRP Operation

When an EIGRP NSF-capable router initially comes back up from an NSF restart, it has no neighbor and its topology table is empty. The router is notified by the redundant (now active) supervisor engine when it needs to bring up the interfaces, reacquire neighbors, and rebuild the topology and routing tables. The restarting router and its peers must accomplish these tasks without interrupting the data traffic directed toward the restarting router. EIGRP peer routers maintain the routes learned from the restarting router and continue forwarding traffic through the NSF restart process.

To prevent an adjacency reset by the neighbors, the restarting router will use a new Restart (RS) bit in the EIGRP packet header to indicate a restart. The RS bit will be set in the hello packets and in the initial INIT update packets during the NSF restart period. The RS bit in the hello packets allows the neighbors to be quickly notified of the NSF restart. Without seeing the RS bit, the neighbor can only detect an adjacency reset by receiving an INIT update or by the expiration of the hello hold timer. Without the RS bit, a neighbor does not know if the adjacency reset should be handled using NSF or the normal startup method.

When the neighbor receives the restart indication, either by receiving the hello packet or the INIT packet, it will recognize the restarting peer in its peer list and will maintain the adjacency with the restarting router. The neighbor then sends it topology table to the restarting router with the RS bit set in the first update packet indicating that it is NSF-aware and is helping out the restarting router. The neighbor does not set the RS bit in their hello packets, unless it is also a NSF restarting neighbor.

Note A router may be NSF-aware but may not be participating in helping out the NSF restarting neighbor because it is coming up from a cold start.

If at least one of the peer routers is NSF-aware, the restarting router would then receive updates and rebuild its database. The restarting router must then find out if it had converged so that it can notify the routing information base (RIB). Each NSF-aware router is required to send an end of table (EOT) marker in the last update packet to indicate the end of the table content. The restarting router knows it has converged when it receives the EOT marker. The restarting router can then begin sending updates.

An NSF-aware peer would know when the restarting router had converged when it receives an EOT indication from the restarting router. The peer then scans its topology table to search for the routes with the restarted neighbor as the source. The peer compares the route timestamp with the restart event timestamp to determine if the route is still available. The peer then goes active to find alternate paths for the routes that are no longer available through the restarted router.

When the restarting router has received all EOT indications from its neighbors or when the NSF converge timer expires, EIGRP will notify the RIB of convergence. EIGRP waits for the RIB convergence signal and then floods its topology table to all awaiting NSF-aware peers.

NSF Benefits and Restrictions

Cisco NSF provides these benefits:

- Improved network availability—NSF continues forwarding network traffic and application state information so that user session information is maintained after a switchover.
- Overall network stability—Network stability may be improved with the reduction in the number of route flaps that had been created when routers in the network failed and lost their routing tables.
- Neighboring routers do not detect a link flap—Because the interfaces remain up throughout a switchover, neighboring routers do not detect a link flap (the link does not go down and come back up).
- Prevents routing flaps—Because SSO continues forwarding network traffic in the event of a switchover, routing flaps are avoided.
- No loss of user sessions—User sessions established before the switchover are maintained.

Cisco NSF with SSO has these restrictions:

- For NSF operation, you must have SSO configured on the device.
- NSF with SSO supports IP Version 4 traffic and protocols only.
- Enhanced Object Tracking is not SSO-aware and cannot be used with Hot Standby Routing Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) in SSO mode.
- The VRRP is not SSO-aware, meaning state information is not maintained between the active and standby supervisor engine during normal operation. VRRP and SSO can coexist but both features work independently. Traffic that relies on VRRP may switch to the VRRP standby in the event of a supervisor switchover.
- NSF with SSO does not support L2VPN.
- Except in VSS mode, Release 12.2(33)SXH and later releases support IPv4 Multiprotocol Label Switching (MPLS) with Cisco NSF with SSO. In VSS mode, Release 12.2(33)SXI2 and later releases support IPv4 MPLS with Cisco NSF with SSO. For additional information about supported MPLS features, search for “MPLS” in the *Release Notes for Cisco IOS Release 12.2(33)SXH and Later Releases*:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/features.html
- All neighboring devices participating in BGP NSF must be NSF-capable and configured for BGP graceful restart.
- OSPF NSF for virtual links is not supported.
- All OSPF networking devices on the same network segment must be NSF-aware (running an NSF software image).
- For IETF IS-IS, all neighboring devices must be running an NSF-aware software image.
- IPv4 Multicast NSF with SSO is supported by the PFC3 only.
- The underlying unicast protocols must be NSF-aware in order to use multicast NSF with SSO.
- Bidirectional forwarding detection (BFD) is not SSO-aware and is not supported by NSF with SSO.

Supervisor Engine Configuration Synchronization

These sections describe supervisor engine configuration synchronization:

- Supervisor Engine Redundancy Guidelines and Restrictions, page 6-9
- Redundancy Configuration Guidelines and Restrictions, page 6-9

Note Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine.

Supervisor Engine Redundancy Guidelines and Restrictions

These sections describe supervisor engine redundancy guidelines and restrictions:

- Redundancy Configuration Guidelines and Restrictions, page 6-9
- Hardware Configuration Guidelines and Restrictions, page 6-10
- Configuration Mode Restrictions, page 6-11

Redundancy Configuration Guidelines and Restrictions

These guidelines and restrictions apply to all redundancy modes:

- Because the Supervisor Engine 720 and Supervisor Engine 720-10GE integrate the switch fabric functionality on the supervisor engine, a supervisor engine switchover will force a fabric switchover as well. During the fabric switchover, data will be lost for a period of between 0.5 seconds and 1.5 seconds. When a Supervisor Engine 720 or Supervisor Engine 720-10GE is installed in a Catalyst 6500 series switch E series chassis (such as the WS-C6509-E), Release 12.2(33)SXH incorporates a fast fabric switchover mechanism, called enhanced hot-standby fabric switchover time, to reduce the data loss to a period of between 50 milliseconds and 0.5 seconds for feature-capable cards.
- When a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active unless you enter the **fabric switching-mode allow dcef-only** command.
- With a Supervisor Engine 720, if all the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on both supervisor engines, which ensures that all modules are operating in dCEF mode and simplifies switchover to the redundant supervisor engine.

- Using the supervisor engine Ethernet ports as routed ports or EtherChannel ports will significantly increase the switchover time.

Note In Cisco IOS Release 12.2(33)SXI2 and later releases, the switchover recovery time is greatly improved for EtherChannels containing the Ethernet ports of both the active and standby supervisor engines. This improvement, which reduces the data disruption period from up to 4 seconds to less than one second, applies only to Supervisor Engine 720 and Supervisor Engine 32, and applies only when the EtherChannel contains no ports other than those of the supervisor engines.

- Supervisor engine redundancy does not provide supervisor engine mirroring or supervisor engine load balancing. Only one supervisor engine is active.
- Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine.
- Supervisor engine switchover takes place after the failed supervisor engine completes a core dump. A core dump can take up to 15 minutes. To get faster switchover time, disable core dump on the supervisor engines.
- If a fabric synchronization error occurs, the default behavior is to switchover to the redundant supervisor engine. In some cases, a switchover to the redundant supervisor engine is more disruptive than powering down the module that caused the fabric synchronization error. Enter the **no fabric error-recovery fabric-switchover** command to disable the switchover and power down the module with the fabric synchronization error.

Hardware Configuration Guidelines and Restrictions

For redundant operation, the following guidelines and restrictions must be met:

- Cisco IOS running on the SP and the RP supports redundant configurations where the supervisor engines are identical. If they are not identical, one will boot first and become active and hold the other supervisor engine in a reset condition.
- Each supervisor engine must have the resources to run the switch on its own, which means all supervisor engine resources are duplicated, including all flash devices.
- Make separate console connections to each supervisor engine. Do not connect a Y cable to the console ports.
- Both supervisor engines must have the same system image (see the "Copying Files to the Redundant Supervisor Engine" section on page 6-20).

Note If a newly installed redundant supervisor engine has the Catalyst operating system installed, remove the active supervisor engine and boot the switch with only the redundant supervisor engine installed. Follow the procedures in the current release notes to convert the redundant supervisor engine from the Catalyst operating system.

- The configuration register in the startup-config must be set to autoboot.

Note There is no support for booting from the network.

Configuration Mode Restrictions

The following configuration restrictions apply during the startup synchronization process:

- You cannot perform configuration changes during the startup (bulk) synchronization. If you attempt to make configuration changes during this process, the following message is generated:

```
Config mode locked out till standby initializes
```

- If configuration changes occur at the same time as a supervisor engine switchover, these configuration changes are lost.

NSF Configuration Tasks

The following sections describe the configuration tasks for the NSF feature:

Configuring SSO

You must configure SSO in order to use NSF with any supported protocol. To configure SSO, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`redundancy`**	Enters redundancy configuration mode.
Step 2	`Router(config-red)#` **`mode sso`**	Configures SSO. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO mode.

	Command	Purpose
Step 3	Router# **show running-config**	Verifies that SSO is enabled.
Step 4	Router# **show redundancy states**	Displays the operating redundancy mode.

This example shows how to configure the system for SSO:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# redundancy
Router(config-red)# mode sso
Router(config-red)# end
```

Verifying the Redundancy States

To verify the redundancy states, enter the **show redundancy states** command:

```
Router# show redundancy states
my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 5

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up

   client count = 29
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

In this example, the system cannot enter the redundancy state because the second supervisor engine is disabled or missing:

```
Router# show redundancy states
       my state = 13 -ACTIVE
     peer state = 1  -DISABLED
           Mode = Simplex
           Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = Non Redundant
     Maintenance Mode = Disabled
 Communications = Down      Reason: Simplex mode

   client count = 11
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 4000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 7
           RF debug mask = 0x0
```

Configuring Multicast MLS NSF with SSO

Note The commands in this section are optional and can be used to customize your configuration. For most users, the default settings are adequate.

Multicast MLS NSF with SSO is on by default when SSO is selected as the redundancy mode. To configure multicast NSF with SSO parameters, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# mls ip multicast sso convergence-time time`	Specifies the maximum time to wait for protocol convergence; valid values are from 0 to 3600 seconds.
Step 3	`Router(config)# mls ip multicast sso leak interval`	Specifies the packet leak interval; valid values are from 0 to 3600 seconds. For PIM sparse mode and PIM dense mode this is the period of time after which packet leaking for existing PIM sparse mode and PIM dense mode multicast forwarding entries should be completed.
Step 4	`Router(config)# mls ip multicast sso leak percentage`	Specifies the percentage of multicast flows; valid values are from 1 to 100 percent. The value represents the percentage of the total number of existing PIM sparse mode and PIM dense mode multicast flows that should be flagged for packet leaking.

Verifying Multicast NSF with SSO

To verify the multicast NSF with SSO settings, enter the **show mls ip multicast sso** command:

```
router# show mls ip multicast sso
Multicast SSO is enabled
Multicast HA Parameters
----------------------------------------------------+------+
protocol convergence timeout                         120 secs
flow leak percent                                    10
flow leak interval                                   60 secs
```

Configuring CEF NSF

The CEF NSF feature operates by default while the networking device is running in SSO mode. No configuration is necessary.

Verifying CEF NSF

To verify that CEF is NSF-capable, enter the **show cef state** command:

```
router# show cef state

CEF Status [RP]
CEF enabled/running
dCEF enabled/running
CEF switching enabled/running
CEF default capabilities:
Always FIB switching:      yes
Default CEF switching:     yes
Default dCEF switching:    yes
Update HWIDB counters:     no
Drop multicast packets:    no
.
.
.
CEF NSF capable:           yes
IPC delayed func on SSO:   no
RRP state:
I am standby RRP:          no
My logical slot:           0
RF PeerComm:               no
```

Configuring BGP NSF

Note You must configure BGP graceful restart on all peer devices participating in BGP NSF.

To configure BGP for NSF, perform this task (repeat this procedure on each of the BGP NSF peer devices):

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **router bgp** *as-number*	Enables a BGP routing process, which places the router in router configuration mode.
Step 3	Router(config-router)# **bgp graceful-restart**	Enables the BGP graceful restart capability, starting BGP NSF. If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. Use this command on the restarting router and all of its peers.

Verifying BGP NSF

To verify BGP NSF, you must check that the graceful restart function is configured on the SSO-enabled networking device and on the neighbor devices. To verify, follow these steps:

Step 1 Verify that "bgp graceful-restart" appears in the BGP configuration of the SSO-enabled router by entering the **show running-config** command:

```
Router# show running-config

.
.
.
router bgp 120
.
.
.
bgp graceful-restart
 neighbor 10.2.2.2 remote-as 300
.
.
.
```

Step 2 Repeat step 1 on each of the BGP neighbors.

Step 3 On the SSO device and the neighbor device, verify that the graceful restart function is shown as both advertised and received, and confirm the address families that have the graceful restart capability. If no address families are listed, then BGP NSF also will not occur:

```
router# show ip bgp neighbors x.x.x.x

BGP neighbor is 192.168.2.2,  remote AS YY, external link
  BGP version 4, remote router ID 192.168.2.2
  BGP state = Established, up for 00:01:18
  Last read 00:00:17, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh:advertised and received(new)
    Address family IPv4 Unicast:advertised and received
    Address famiiy IPv4 Multicast:advertised and received
    Graceful Restart Capabilty:advertised and received
      Remote Restart timer is 120 seconds
      Address families preserved by peer:
        IPv4 Unicast, IPv4 Multicast
  Received 1539 messages, 0 notifications, 0 in queue
  Sent 1544 messages, 0 notifications, 0 in queue
  Default minimum time between advertisement runs is 30 seconds
```

Configuring OSPF NSF

Note All peer devices participating in OSPF NSF must be made OSPF NSF-aware, which happens automatically once you install an NSF software image on the device.

To configure OSPF NSF, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.

	Command	Purpose
Step 2	`Router(config)# router ospf processID`	Enables an OSPF routing process, which places the router in router configuration mode.
Step 3	`Router(config-router)# nsf`	Enables NSF operations for OSPF.

Verifying OSPF NSF

To verify OSPF NSF, you must check that the NSF function is configured on the SSO-enabled networking device. To verify OSPF NSF, follow these steps:

Step 1 Verify that 'nsf' appears in the OSPF configuration of the SSO-enabled device by entering the **show running-config** command:

```
Router# show running-config

router ospf 120
log-adjacency-changes
nsf
network 192.168.20.0 0.0.0.255 area 0
network 192.168.30.0 0.0.0.255 area 1
network 192.168.40.0 0.0.0.255 area 2
.
.
.
```

Step 2 Enter the **show ip ospf** command to verify that NSF is enabled on the device:

```
router> show ip ospf

Routing Process "ospf 1" with ID 192.168.2.1 and Domain ID 0.0.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
Non-Stop Forwarding enabled, last NSF restart 00:02:06 ago (took 44 secs)
Area BACKBONE(0)
Number of interfaces in this area is 1 (0 loopback)
 Area has no authentication
 SPF algorithm executed 3 times
```

Configuring IS-IS NSF

To configure IS-IS NSF, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# router isis [tag]`	Enables an IS-IS routing process, which places the router in router configuration mode.
Step 3	`Router(config-router)# nsf [cisco | ietf]`	Enables NSF operation for IS-IS. Enter the **ietf** keyword to enable IS-IS in a homogeneous network where adjacencies with networking devices supporting IETF draft-based restartability is guaranteed. Enter the **cisco** keyword to run IS-IS in heterogeneous networks that might not have adjacencies with NSF-aware networking devices.
Step 4	`Router(config-router)# nsf interval [minutes]`	(Optional) Specifies the minimum time between NSF restart attempts. The default time between *consecutive* NSF restart attempts is 5 minutes.
Step 5	`Router(config-router)# nsf t3 {manual [seconds] | adjacency}`	(Optional) Specifies the time IS-IS will wait for the IS-IS database to synchronize before generating overloaded link-state information for itself and flooding that information out to its neighbors. The **t3** keyword applies only if you selected IETF operation. When you specify **adjacency**, the router that is restarting obtains its wait time from neighboring devices.
Step 6	`Router(config-router)# nsf interface wait seconds`	(Optional) Specifies how long an IS-IS NSF restart will wait for all interfaces with IS-IS adjacencies to come up before completing the restart. The default is 10 seconds.

Verifying IS-IS NSF

To verify IS-IS NSF, you must check that the NSF function is configured on the SSO-enabled networking device. To verify IS-IS NSF, follow these steps:

Step 1 Verify that “nsf” appears in the IS-IS configuration of the SSO-enabled device by entering the **show running-config** command. The display will show either the Cisco IS-IS or the IETF IS-IS configuration. The following display indicates that the device uses the Cisco implementation of IS-IS NSF:

```
Router# show running-config
<...Output Truncated...>
router isis
nsf cisco
<...Output Truncated...>
```

Step 2 If the NSF configuration is set to **cisco**, enter the **show isis nsf** command to verify that NSF is enabled on the device. Using the Cisco configuration, the display output will be different on the active and redundant RPs. The following display shows sample output for the Cisco configuration on the active RP. In this example, note the presence of “NSF restart enabled”:

```
router# show isis nsf

NSF is ENABLED, mode 'cisco'

RP is ACTIVE, standby ready, bulk sync complete
NSF interval timer expired (NSF restart enabled)
Checkpointing enabled, no errors
Local state:ACTIVE,  Peer state:STANDBY HOT,  Mode:SSO
```

The following display shows sample output for the Cisco configuration on the standby RP. In this example, note the presence of “NSF restart enabled”:

```
router# show isis nsf

NSF enabled, mode 'cisco'
RP is STANDBY, chkpt msg receive count:ADJ 2, LSP 7
NSF interval timer notification received (NSF restart enabled)
Checkpointing enabled, no errors
Local state:STANDBY HOT,  Peer state:ACTIVE,  Mode:SSO
```

Step 3 If the NSF configuration is set to **ietf**, enter the **show isis nsf** command to verify that NSF is enabled on the device. The following display shows sample output for the IETF IS-IS configuration on the networking device:

```
router# show isis nsf

NSF is ENABLED, mode IETF
NSF pdb state:Inactive
NSF L1 active interfaces:0
NSF L1 active LSPs:0
NSF interfaces awaiting L1 CSNP:0
Awaiting L1 LSPs:
NSF L2 active interfaces:0
NSF L2 active LSPs:0
NSF interfaces awaiting L2 CSNP:0
Awaiting L2 LSPs:
Interface:Serial3/0/2
    NSF L1 Restart state:Running
    NSF p2p Restart retransmissions:0
    Maximum L1 NSF Restart retransmissions:3
    L1 NSF ACK requested:FALSE
    L1 NSF CSNP requested:FALSE
    NSF L2 Restart state:Running
    NSF p2p Restart retransmissions:0
    Maximum L2 NSF Restart retransmissions:3
    L2 NSF ACK requested:FALSE
Interface:GigabitEthernet2/0/0
    NSF L1 Restart state:Running
    NSF L1 Restart retransmissions:0
    Maximum L1 NSF Restart retransmissions:3
    L1 NSF ACK requested:FALSE
    L1 NSF CSNP requested:FALSE
    NSF L2 Restart state:Running
    NSF L2 Restart retransmissions:0
    Maximum L2 NSF Restart retransmissions:3
    L2 NSF ACK requested:FALSE
    L2 NSF CSNP requested:FALSE
Interface:Loopback1
```

```
    NSF L1 Restart state:Running
    NSF L1 Restart retransmissions:0
    Maximum L1 NSF Restart retransmissions:3
    L1 NSF ACK requested:FALSE
    L1 NSF CSNP requested:FALSE
    NSF L2 Restart state:Running
    NSF L2 Restart retransmissions:0
    Maximum L2 NSF Restart retransmissions:3
    L2 NSF ACK requested:FALSE
    L2 NSF CSNP requested:FALSE
```

Configuring EIGRP NSF

To configure EIGRP NSF, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **router eigrp** *as-number*	Enables an EIGRP routing process, which places the router in router configuration mode.
Step 3	Router(config-router)# **nsf**	Enables EIGRP NSF. Use this command on the restarting router and all of its peers.

Verifying EIGRP NSF

To verify EIGRP NSF, you must check that the NSF function is configured on the SSO-enabled networking device. To verify EIGRP NSF, follow these steps:

Step 1 Verify that "nsf" appears in the EIGRP configuration of the SSO-enabled device by entering the **show running-config** command:

```
Router# show running-config
.
.
.
router eigrp 100
 auto-summary
 nsf
.
.
.
```

Step 2 Enter the **show ip protocols** command to verify that NSF is enabled on the device:

```
Router# show ip protocols
*** IP Routing is NSF aware ***
Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
```

```
EIGRP maximum metric variance 1
Redistributing: eigrp 100
EIGRP NSF-aware route hold timer is 240s
EIGRP NSF enabled
   NSF signal timer is 20s
   NSF converge timer is 120s
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
Routing Information Sources:
  Gateway         Distance      Last Update
Distance: internal 90 external 170
```

Synchronizing the Supervisor Engine Configurations

During normal operation, the startup-config and config-registers configurations are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.

Copying Files to the Redundant Supervisor Engine

Enter this command to copy a file to the **disk0:** device on a redundant supervisor engine:

Router# **copy** *source_device***:***source_filename* **slavedisk0:***target_filename*

Enter this command to copy a file to the **bootflash:** device on a redundant supervisor engine:

Router# **copy** *source_device***:***source_filename* **slavesup-bootflash:***target_filename*

Enter this command to copy a file to the **bootflash:** device on a redundant RP:

Router# **copy** *source_device***:***source_filename* **slavebootflash:***target_filename*

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring RPR Supervisor Engine Redundancy

This chapter describes how to configure supervisor engine redundancy using route processor redundancy (RPR).

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html
- Cisco ME 6500 Series Ethernet switches do not support redundancy.
- In RPR redundancy mode, the ports on a Supervisor Engine 720-10GE in standby mode are disabled.
- RPR supports IPv6 multicast traffic.
- Release 12.2(33)SXH and later releases do not support Route Processor Redundancy Plus (RPR+).

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding RPR

These sections describe RPR supervisor engine redundancy:

Supervisor Engine Redundancy Overview

Catalyst 6500 series switches support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. RPR supports a switchover time of 2 or more minutes.

The following events cause a switchover:

- A hardware failure on the active supervisor engine
- Clock synchronization failure between supervisor engines
- A manual switchover

RPR Operation

RPR supports the following features:

- Auto-startup and bootvar synchronization between active and redundant supervisor engines
- Hardware signals that detect and decide the active or redundant status of supervisor engines
- Clock synchronization every 60 seconds from the active to the redundant supervisor engine
- A redundant supervisor engine that is booted but not all subsystems are up: if the active supervisor engine fails, the redundant supervisor engine become fully operational
- An operational supervisor engine present in place of the failed unit becomes the redundant supervisor engine
- Support for fast software upgrade (FSU) (See the "Performing a Fast Software Upgrade" section on page 7-6.)

When the switch is powered on, RPR runs between the two supervisor engines. The supervisor engine that boots first becomes the RPR active supervisor engine. The Multilayer Switch Feature Card and Policy Feature Card become fully operational. The route processor (RP) and PFC on the redundant supervisor engine come out of reset but are not operational.

In a switchover, the redundant supervisor engine become fully operational and the following occurs:

- All switching modules power up again
- Remaining subsystems on the RP (including Layer 2 and Layer 3 protocols) are brought up
- Access control lists (ACLs) are reprogrammed into supervisor engine hardware

Note In a switchover, there is a disruption of traffic because some address states are lost and then restored after they are dynamically redetermined.

Supervisor Engine Configuration Synchronization

Note Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine.

During RPR mode operation, the startup-config files and the config-register configurations are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.

Supervisor Engine Redundancy Guidelines and Restrictions

These sections describe supervisor engine redundancy guidelines and restrictions:

- Redundancy Guidelines and Restrictions, page 7-3
- Hardware Configuration Guidelines and Restrictions, page 7-3
- Configuration Mode Restrictions, page 7-4

Redundancy Guidelines and Restrictions

These guidelines and restrictions apply to RPR:

- When a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.
- Supervisor engine redundancy does not provide supervisor engine mirroring or supervisor engine load balancing. Only one supervisor engine is active.
- Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine.
- Supervisor engine switchover takes place after the failed supervisor engine completes a core dump. A core dump can take up to 15 minutes. To get faster switchover time, disable core dump on the supervisor engines.

Hardware Configuration Guidelines and Restrictions

For redundant operation, the following guidelines and restrictions must be met:

- Cisco IOS running on the SP and RP supports redundant configurations where the supervisor engines are identical. If they are not identical, one will boot first and become active and hold the other supervisor engine in a reset condition.
- Each supervisor engine must have the resources to run the switch on its own, which means all supervisor engine resources are duplicated, including all flash devices.
- Make separate console connections to each supervisor engine. Do not connect a Y cable to the console ports.

- Except during an FSU, both supervisor engines must have the same system image (see the "Copying Files to the RP" section on page 7-7).

  Note If a newly installed redundant supervisor engine has the Catalyst operating system installed, remove the active supervisor engine and boot the switch with only the redundant supervisor engine installed. Follow the procedures in the current release notes to convert the redundant supervisor engine from the Catalyst operating system.

- The configuration register must be set to 0x2102 (**config-register 0x2102**).

Note There is no support for booting from the network.

Configuration Mode Restrictions

The following configuration restrictions apply during the startup synchronization process:

- You cannot perform configuration changes during the startup (bulk) synchronization. If you attempt to make configuration changes during this process, the following message is generated:

```
Config mode locked out till standby initializes
```

- If configuration changes occur at the same time as a supervisor engine switchover, these configuration changes are lost.

Configuring Supervisor Engine Redundancy

These sections describe how to configure supervisor engine redundancy:

Configuring Redundancy

To configure redundancy, perform this task:

	Command	Purpose
Step 1	`Router(config)# `**`redundancy`**	Enters redundancy configuration mode.
Step 2	`Router(config-red)# `**`mode rpr`**	Configures RPR. When this command is entered, the redundant supervisor engine is reloaded and begins to work in RPR mode.
Step 3	`Router# `**`show running-config`**	Verifies that RPR is enabled.
Step 4	`Router# `**`show redundancy states`**	Displays the operating redundancy mode.

This example shows how to configure the system for RPR:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# redundancy
Router(config-red)# mode rpr
Router(config-red)# end
```

Synchronizing the Supervisor Engine Configurations

During normal operation, the startup-config and config-registers configuration are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.

Note Do not change the default auto-sync configuration.

Displaying the Redundancy States

To display the redundancy states, perform this task:

Command	Purpose
Router# **show redundancy states**	Displays the redundancy states.

This example shows how to display the redundancy states:

```
Router# show redundancy states
my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Route Processor Redundancy
Redundancy Mode (Configured)  = Route Processor Redundancy
     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up

   client count = 11
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 18
           RF debug mask = 0x0
```

In this example, the system cannot enter the redundancy state because the second supervisor engine is disabled or missing:

```
Router# show redundancy states
       my state = 13 -ACTIVE
     peer state = 1  -DISABLED
           Mode = Simplex
           Unit = Primary
        Unit ID = 1
```

```
Redundancy Mode (Operational) = rpr
Redundancy Mode (Configured)  = rpr
Redundancy State              = Non Redundant
     Maintenance Mode = Disabled
 Communications = Down      Reason: Simplex mode

   client count = 11
 client_notification_TMR = 30000 milliseconds
          keep_alive TMR = 4000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 7
           RF debug mask = 0x0
```

Performing a Fast Software Upgrade

The fast software upgrade (FSU) procedure supported by RPR allows you to upgrade the Cisco IOS image on the supervisor engines without reloading the system.

Note

- If you are performing a first-time upgrade to RPR from EHSA, you must reload both supervisor engines. FSU from EHSA is not supported.
- FSU from an IOS image to a modular IOS image is not supported. FSU from a modular IOS image to an IOS image is not supported. (CSCsb07831)

To perform an FSU, perform this task:

	Command	Purpose
Step 1	Router# **copy** *source_device***:***source_filename* {**disk0** | **disk1**}**:***target_filename*	Copies the new Cisco IOS image to the **disk0:** device or the **disk1:** device on the active supervisor engine.
	Or:	
	Router# **copy** *source_device***:***source_filename* **sup-bootflash:***target_filename*	Copies the new Cisco IOS image to the **bootflash:** device on the active supervisor engine.
	Or:	
	Router# **copy** *source_device***:***source_filename* {**slavedisk0** | **slavedisk1**}**:***target_filename*	Copies the new Cisco IOS image to the **disk0:** device or the **disk1:** device on the redundant supervisor engine.
	Or:	
	Router# **copy** *source_device***:***source_filename* **slavesup-bootflash:***target_filename*	Copies the new Cisco IOS image to the **bootflash:** device on the redundant supervisor engine.
Step 2	Router# **config terminal** Router(config)# **config-register 0x2102** Router(config)# **boot system flash** *device*:*file_name*	Configures the supervisor engines to boot the new image.
Step 3	Router# **copy running-config start-config**	Saves the configuration.

	Command	Purpose
Step 4	`Router# hw-module module num reset`	Reloads the redundant supervisor engine and brings it back online (running the new version of the Cisco IOS software). **Note** Before reloading the redundant supervisor engine, make sure you wait long enough to ensure that all configuration synchronization changes have completed.
Step 5	`Router# redundancy force-switchover`	Conducts a manual switchover to the redundant supervisor engine. The redundant supervisor engine becomes the new active supervisor engine running the new Cisco IOS image. The modules are reloaded and the module software is downloaded from the new active supervisor engine. The old active supervisor engine reboots with the new image and becomes the redundant supervisor engine. **Note** To perform an EHSA to RPR FSU, use the **reload** command in Step 5.

This example shows how to perform an FSU:

```
Router# config terminal
Router(config)# config-register 0x2102
Router(config)# boot system flash disk0:image_name
Router# copy running-config start-config
Router# hw-module module redundant_spervisor_reset
Router# redundancy force-switchover
```

Copying Files to the RP

Use the following command to copy a file to the **bootflash:** device on an active RP:

```
Router# copy source_device:source_filename bootflash:target_filename
```

Use the following command to copy a file to the **bootflash:** device on a redundant RP:

```
Router# copy source_device:source_filename slavebootflash:target_filename
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 3

Interface and Hardware Components

Configuring Interfaces

This chapter describes how to configure interfaces in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Interface Configuration

Many features in the software are enabled on a per-interface basis. When you enter the **interface** command, you must specify the following information:

- Interface type:
  - Fast Ethernet (use the **fastethernet** keyword)
  - Gigabit Ethernet (use the **gigabitethernet** keyword)
  - 10-Gigabit Ethernet (use the **tengigabitethernet** keyword)

Note For WAN interfaces, see the configuration note for the WAN module.

- Slot number—The slot in which the module is installed. On switches supported by Cisco IOS Release 12.2SX, slots are numbered starting with 1 from top to bottom.
- Port number—The physical port number on the module. On switches supported by Cisco IOS Release 12.2SX, the port numbers always begin with 1. When facing the rear of the switch, ports are numbered from the left to the right.

You can identify ports from the physical location. You also can use **show** commands to display information about a specific port, or all the ports.

Using the Interface Command

Note You use the commands described in this section to configure both physical ports and logical interfaces.

These procedures apply to all interface configuration processes. Begin the interface configuration process in global configuration mode. To use the interface command, follow these steps:

Step 1 Enter the **configure terminal** command at the privileged EXEC prompt to enter global configuration mode:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Step 2 In the global configuration mode, enter the **interfaces** command. Identify the interface type and the number of the connector or interface card.

The following example shows how to select Fast Ethernet, slot 5, interface 1:

```
Router(config)# interfaces fastethernet 5/1
Router(config-if)#
```

Step 3 Enter the **show interfaces** EXEC command to see a list of all interfaces that are installed. A report is provided for each interface that the device supports, as shown in this display:

```
Router# show interfaces fastethernet 5/48
FastEthernet5/48 is up, line protocol is up
  Hardware is C6k 100Mb 802.3, address is 0050.f0ac.3083 (bia 0050.f0ac.3083)
  Internet address is 172.20.52.18/27
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     4834677 packets input, 329545368 bytes, 0 no buffer
     Received 4796465 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     51926 packets output, 15070051 bytes, 0 underruns
     0 output errors, 2 collisions, 2 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Step 4 Enter the **show hardware** EXEC command to see a list of the system software and hardware:

```
Router# show hardware
Cisco Internetwork Operating System Software
IOS (tm) c6sup2_rp Software (c6sup2_rp-JSV-M), Version 12.1(5c)EX, EARLY DEPLOY)
Synced to mainline version: 12.1(5c)
TAC:Home:Software:Ios General:CiscoIOSRoadmap:12.1
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Wed 28-Mar-01 17:52 by hqluong
Image text-base: 0x30008980, data-base: 0x315D0000

ROM: System Bootstrap, Version 12.1(3r)E2, RELEASE SOFTWARE (fc1)
BOOTFLASH: c6sup2_rp Software (c6sup2_rp-JSV-M), Version 12.1(5c)EX, EARLY DEPL)

Router uptime is 2 hours, 55 minutes
System returned to ROM by power-on (SP by power-on)
Running default software

cisco Catalyst 6000 (R7000) processor with 114688K/16384K bytes of memory.
Processor board ID SAD04430J9K
R7000 CPU at 300Mhz, Implementation 39, Rev 2.1, 256KB L2, 1024KB L3 Cache
Last reset from power-on
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
1 Virtual Ethernet/IEEE 802.3  interface(s)
48 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)
381K bytes of non-volatile configuration memory.

16384K bytes of Flash internal SIMM (Sector size 512K).
Configuration register is 0x2
```

Step 5 To begin configuring Fast Ethernet port 5/5, enter the **interface** keyword, interface type, and slot number/port number at the privileged EXEC prompt, as shown in the following example:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/5
Router(config-if)#
```

Note You do not need to add a space between the interface type and interface number. For example, in the preceding line you can specify either *fastethernet 5/5* or *fastethernet5/5*.

Step 6 After each **interface** command, enter the interface configuration commands your particular interface requires.

The commands you enter define the protocols and applications that will run on the interface. The commands are collected and applied to the **interface** command until you enter another **interface** command or press **Ctrl-Z** to get out of interface configuration mode and return to privileged EXEC mode.

Step 7 After you configure an interface, check its status by using the EXEC **show** commands listed in "Monitoring and Maintaining Interfaces" section on page 8-17.

Configuring a Range of Interfaces

The interface-range configuration mode allows you to configure multiple interfaces with the same configuration parameters. After you enter the interface-range configuration mode, all command parameters you enter are attributed to all interfaces within that range until you exit out of the interface-range configuration mode.

To configure a range of interfaces with the same configuration, perform this task:

Command	Purpose
Router(config)# [**no**] **interface range** {{**vlan** *vlan_ID* - *vlan_ID* [, **vlan** *vlan_ID* - *vlan_ID*]} | {*type*[1] *slot/port* - *port* [, *type*[1] *slot/port* - *port*]} | {*macro_name* [, *macro_name*]}}	Selects the range of interfaces to be configured.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring a range of interfaces, note the following information:

- For information about macros, see the "Defining and Using Interface-Range Macros" section on page 8-6.
- You can enter up to five comma-separated ranges.
- You are not required to enter spaces before or after the comma.
- You do not need to add a space between the interface numbers and the dash when using the **interface range** command.

- The **no interface range** command supports VLAN interfaces.
- The **interface range** command supports VLAN interfaces for which Layer 2 VLANs have not been created with the **interface vlan** command.

Note The link state messages (LINK-3-UPDOWN and LINEPROTO-5-UPDOWN) are disabled by default. Enter the **logging event link status** command on each interface where you want the messages enabled.

This example shows how to reenable all Fast Ethernet ports 5/1 to 5/5:

```
Router(config)# interface range fastethernet 5/1 - 5
Router(config-if)# no shutdown
Router(config-if)#
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/2, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/3, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/4, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
5, changed state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
3, changed state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
4, changed state to up
Router(config-if)#
```

This example shows how to use a comma to add different interface type strings to the range to reenable all Fast Ethernet ports in the range 5/1 to 5/5 and both Gigabit Ethernet ports (1/1 and 1/2):

```
Router(config-if)# interface range fastethernet 5/1 - 5, gigabitethernet 1/1 - 2
Router(config-if)# no shutdown
Router(config-if)#
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/1, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/2, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/3, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/4, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet1/1, changed state to
 up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet1/2, changed state to
 up
*Oct  6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
5, changed state to up
*Oct  6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
3, changed state to up
*Oct  6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
4, changed state to up
Router(config-if)#
```

If you enter multiple configuration commands while you are in interface-range configuration mode, each command is executed as it is entered (they are not batched together and executed after you exit interface-range configuration mode).

If you exit interface-range configuration mode while the commands are being executed, some commands may not be executed on all interfaces in the range. Wait until the command prompt reappears before exiting interface-range configuration mode.

Defining and Using Interface-Range Macros

You can define an interface-range macro to automatically select a range of interfaces for configuration. Before you can use the **macro** keyword in the **interface range macro** command string, you must define the macro.

To define an interface-range macro, perform this task:

Command	Purpose
Router(config)# **define interface-range** *macro_name* {**vlan** *vlan_ID* - *vlan_ID*} | {*type*[1] *slot/port* - *port*} [**,** {*type*[1] *slot/port* - *port*}]	Defines the interface-range macro and save it in NVRAM.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to define an interface-range macro named enet_list to select Fast Ethernet ports 5/1 through 5/4:

```
Router(config)# define interface-range enet_list fastethernet 5/1 - 4
```

To show the defined interface-range macro configuration, perform this task:

Command	Purpose
Router# **show running-config**	Shows the defined interface-range macro configuration.

This example shows how to display the defined interface-range macro named enet_list:

```
Router# show running-config | include define
define interface-range enet_list FastEthernet5/1 - 4
Router#
```

To use an interface-range macro in the **interface range** command, perform this task:

Command	Purpose
Router(config)# **interface range macro** *macro_name*	Selects the interface range to be configured using the values saved in a named interface-range macro.

This example shows how to change to the interface-range configuration mode using the interface-range macro enet_list:

```
Router(config)# interface range macro enet_list
Router(config-if)#
```

Configuring Optional Interface Features

These sections describe optional interface features:

- Configuring Ethernet Interface Speed and Duplex Mode, page 8-7
- Configuring Jumbo Frame Support, page 8-10
- Configuring IEEE 802.3x Flow Control, page 8-13

Configuring Ethernet Interface Speed and Duplex Mode

These sections describe how to configure Ethernet port speed and duplex mode:

Speed and Duplex Mode Configuration Guidelines

You usually configure Ethernet port speed and duplex mode parameters to auto and allow ports to negotiate the speed and duplex mode. If you decide to configure the port speed and duplex modes manually, consider the following information:

- You cannot set the Ethernet port speed to auto (the **no speed** command) if the duplex mode in not set to auto (the **no duplex** command).
- If you configure an Ethernet port speed to a value other than auto (for example, 10, 100, or 1000 Mbps), configure the connecting port to match. Do not configure the connecting port to negotiate the speed.
- If you manually configure the Ethernet port speed to either 10 Mbps or 100 Mbps, the switch prompts you to also configure the duplex mode on the port.

Note A LAN port cannot automatically negotiate Ethernet port speed and duplex mode if the connecting port is configured to a value other than auto.

Caution Changing the Ethernet port speed and duplex mode configuration might shut down and reenable the interface during the reconfiguration.

Configuring the Ethernet Interface Speed

Note If you configure the Ethernet port speed to **auto** on a 10/100-Mbps or 10/100/1000-Mbps Ethernet port, both speed and duplex are autonegotiated. 10-Gigabit Ethernet ports do not support autonegotiation.

To configure the port speed for a 10/100 or a 10/100/1000-Mbps Ethernet port, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface fastethernet** *slot/port*	Selects the Ethernet port to be configured.
Step 2	Router(config-if)# **speed** {**10** | **100** | **1000** | {**auto** [**10 100** [**1000**]]}}	Configures the speed of the Ethernet interface.

When configuring the port speed for a 10/100/1000-Mbps Ethernet port, note the following:

- Enter the **auto 10 100** keywords to restrict the negotiated speed to 10-Mbps or 100-Mbps.
- The **auto 10 100 1000** keywords have the same effect as the **auto** keyword by itself.

This example shows how to configure the speed to 100 Mbps on the Fast Ethernet port 5/4:

```
Router(config)# interface fastethernet 5/4
Router(config-if)# speed 100
```

Setting the Interface Duplex Mode

Note

- 10-Gigabit Ethernet and Gigabit Ethernet are full duplex only. You cannot change the duplex mode on 10-Gigabit Ethernet or Gigabit Ethernet ports or on a 10/100/1000-Mps port configured for Gigabit Ethernet.
- If you set the port speed to auto on a 10/100-Mbps or a 10/100/1000-Mbps Ethernet port, both speed and duplex are autonegotiated. You cannot change the duplex mode of autonegotiation ports.

To set the duplex mode of an Ethernet or Fast Ethernet port, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface fastethernet** *slot/port*	Selects the Ethernet port to be configured.
Step 2	Router(config-if)# **duplex** [**auto** | **full** | **half**]	Sets the duplex mode of the Ethernet port.

This example shows how to set the duplex mode to full on Fast Ethernet port 5/4:

```
Router(config)# interface fastethernet 5/4
Router(config-if)# duplex full
```

Configuring Link Negotiation on Gigabit Ethernet Ports

Note Link negotiation does not negotiate port speed.

On Gigabit Ethernet ports, link negotiation exchanges flow-control parameters, remote fault information, and duplex information. Link negotiation is enabled by default.

The ports on both ends of a link must have the same setting. The link will not come up if the ports at each end of the link are set inconsistently (link negotiation enabled on one port and disabled on the other port).

Table 8-1 shows the four possible link negotiation configurations and the resulting link status for each configuration.

Table 8-1 Link Negotiation Configuration and Possible Link Status

Link Negotiation State		Link Status	
Local Port	**Remote Port**	**Local Port**	**Remote Port**
Off	Off	Up	Up
On	On	Up	Up
Off	On	Up	Down
On	Off	Down	Up

To configure link negotiation on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet slot/port`	Selects the port to be configured.
Step 2	`Router(config-if)# speed nonegotiate`	Disables link negotiation.

This example shows how to enable link negotiation on Gigabit Ethernet port 5/4:

```
Router(config)# interface gigabitethernet 5/4
Router(config-if)# no speed nonegotiate
```

Displaying the Speed and Duplex Mode Configuration

To display the speed and duplex mode configuration for a port, perform this task:

Command	Purpose
`Router# show interfaces type[1] slot/port [transceiver properties]`	Displays the speed and duplex mode configuration. To display autonegotiation status for speed and duplex, add the **transceiver properties** option.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to display the speed and duplex mode of Fast Ethernet port 5/4:

```
Router# show interfaces fastethernet 5/4
FastEthernet5/4 is up, line protocol is up
  Hardware is Cat6K 100Mb Ethernet, address is 0050.f0ac.3058 (bia 0050.f0ac.3058)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:33, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
```

```
  5 minute output rate 0 bits/sec, 0 packets/sec
     1238 packets input, 273598 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     1380 packets output, 514382 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Router#
```

This example shows how to display the speed and duplex autonegotiation status for Gigabit Ethernet port 3/1:

```
Router# show interfaces GigabitEthernet 3/1 transceiver properties
Name : Gi3/1   Admin Speed Nego : Enable ,  Admin Duplex Nego : Enable
Router#
```

Configuring Jumbo Frame Support

These sections describe jumbo frame support:

- Understanding Jumbo Frame Support, page 8-10
- Configuring MTU Sizes, page 8-12

Caution The following switching modules support a maximum ingress frame size of 8092 bytes:

- WS-X6516-GE-TX when operating at 100 Mbps
- WS-X6148-RJ-45, WS-X6148-RJ-45V and WS-X6148-RJ21, WS-X6148-RJ21V
- WS-X6248-RJ-45 and WS-X6248-TEL
- WS-X6248A-RJ-45 and WS-X6248A-TEL
- WS-X6348-RJ-45, WS-X6348-RJ45V and WS-X6348-RJ-21, WX-X6348-RJ21V

When jumbo frame support is configured, these modules drop ingress frames larger than 8092 bytes.

Note The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX do not support jumbo frames.

Understanding Jumbo Frame Support

These sections describe jumbo frame support:

- Jumbo Frame Support Overview, page 8-11
- Ethernet Ports, page 8-11
- VLAN Interfaces, page 8-12

Jumbo Frame Support Overview

A jumbo frame is a frame larger than the default Ethernet size. You enable jumbo frame support by configuring a larger-than-default maximum transmission unit (MTU) size on a port or VLAN interface and configuring the global LAN port MTU size.

Note
- Jumbo frame support fragments routed traffic in software on the route processor (RP).
- Jumbo frame support does not fragment bridged traffic.

Bridged and Routed Traffic Size Check at Ingress 10, 10/100, and 100 Mbps Ethernet and 10-Gigabit Ethernet Ports

Jumbo frame support compares ingress traffic size with the global LAN port MTU size at ingress 10, 10/100, and 100 Mbps Ethernet and 10-Gigabit Ethernet LAN ports that have a nondefault MTU size configured. The port drops traffic that is oversized. You can configure the global LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-13).

Bridged and Routed Traffic Size Check at Ingress Gigabit Ethernet Ports

Gigabit Ethernet LAN ports configured with a nondefault MTU size accept frames containing packets of any size larger than 64 bytes. With a nondefault MTU size configured, Gigabit Ethernet LAN ports do not check for oversize ingress frames.

Routed Traffic Size Check on the PFC

For traffic that needs to be routed, Jumbo frame support on the PFC compares traffic sizes to the configured MTU sizes and provides Layer 3 switching for jumbo traffic between interfaces configured with MTU sizes large enough to accommodate the traffic. Between interfaces that are not configured with large enough MTU sizes, if the "do not fragment bit" is not set, the PFC sends the traffic to the RP to be fragmented and routed in software. If the "do not fragment bit" is set, the PFC drops the traffic.

Bridged and Routed Traffic Size Check at Egress 10, 10/100, and 100 Mbps Ethernet Ports

10, 10/100, and 100 Mbps Ethernet LAN ports configured with a nondefault MTU size transmit frames containing packets of any size larger than 64 bytes. With a nondefault MTU size configured, 10, 10/100, and 100 Mbps Ethernet LAN ports do not check for oversize egress frames.

Bridged and Routed Traffic Size Check at Egress Gigabit Ethernet and 10-Gigabit Ethernet Ports

Jumbo frame support compares egress traffic size with the global egress LAN port MTU size at egress Gigabit Ethernet and 10-Gigabit Ethernet LAN ports that have a nondefault MTU size configured. The port drops traffic that is oversized. You can configure the global LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-13).

Ethernet Ports

These sections describe configuring nondefault MTU sizes on Ethernet ports:

Ethernet Port Overview

Configuring a nondefault MTU size on a 10, 10/100, or 100 Mbps Ethernet port limits ingress packets to the global LAN port MTU size and permits egress traffic of any size larger than 64 bytes.

Configuring a nondefault MTU size on a Gigabit Ethernet port permits ingress packets of any size larger than 64 bytes and limits egress traffic to the global LAN port MTU size.

Configuring a nondefault MTU size on a 10-Gigabit Ethernet port limits ingress and egress packets to the global LAN port MTU size.

Configuring a nondefault MTU size on an Ethernet port limits routed traffic to the configured MTU size.

You can configure the MTU size on any Ethernet port.

Layer 3 Ethernet Ports

On a Layer 3 port, you can configure an MTU size on each Layer 3 Ethernet port that is different than the global LAN port MTU size.

Note Traffic through a Layer 3 Ethernet LAN port that is configured with a nondefault MTU size is also subject to the global LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-13).

Layer 2 Ethernet Ports

On a Layer 2 port, you can only configure an MTU size that matches the global LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-13).

VLAN Interfaces

You can configure a different MTU size on each Layer 3 VLAN interface. Configuring a nondefault MTU size on a VLAN interface limits traffic to the nondefault MTU size. You can configure the MTU size on VLAN interfaces to support jumbo frames.

Configuring MTU Sizes

These sections describe how to configure MTU sizes:

Configuring the MTU Size

To configure the MTU size, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {{**vlan** *vlan_ID*} | {{*type*[1] *slot/port*} | {**port-channel** *port_channel_number*} *slot/port*}}	Selects the interface to configure.
Step 2	Router(config-if)# **mtu** *mtu_size*	Configures the MTU size.

	Command	Purpose
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show running-config interface [{gigabitethernet | tengigabitethernet} slot/port]`	Displays the running configuration.

1. *type* = **fastethernet**, **gigabitethernet**, **tengigabitethernet**, or **ge-wan**

When configuring the MTU size, note the following information:

- For VLAN interfaces and Layer 3 Ethernet ports, supported MTU values are from 64 to 9216 bytes.
- For Layer 2 Ethernet ports, you can configure only the global egress LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-13).

This example shows how to configure the MTU size on Gigabit Ethernet port 1/2:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/2
Router(config-if)# mtu 9216
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show interface gigabitethernet 1/2
GigabitEthernet1/2 is administratively down, line protocol is down
  Hardware is C6k 1000Mb 802.3, address is 0030.9629.9f88 (bia 0030.9629.9f88)
  MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,
<...Output Truncated...>
Router#
```

Configuring the Global Egress LAN Port MTU Size

To configure the global egress LAN port MTU size, perform this task:

	Command	Purpose
Step 1	`Router(config)# system jumbomtu mtu_size`	Configures the global egress LAN port MTU size. **Note** Because it would change all the interface MTU sizes to the default (1500), rather than to any configured nondefault interface MTU size, do not use the system jumbomtu command to set the MTU size to 1500. (CSCtq52016)
Step 2	`Router(config)# end`	Exits configuration mode.

Configuring IEEE 802.3x Flow Control

Gigabit Ethernet and 10-Gigabit Ethernet ports use flow control to stop the transmission of frames to the port for a specified time; other Ethernet ports use flow control to respond to flow-control requests.

If a Gigabit Ethernet or 10-Gigabit Ethernet port receive buffer becomes full, the port can be configured to transmit an IEEE 802.3x pause frame that requests the remote port to delay sending frames for a specified time. All Ethernet ports can can be configured to respond to IEEE 802.3x pause frames from other devices.

To configure flow control on an Ethernet port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects the port to configure.
Step 2	`Router(config-if)# flowcontrol {receive | send} {desired | off | on}`	Configures a port to send or respond to pause frames.
Step 3	`Router# show interfaces [type[1] slot/port] flowcontrol`	Displays the flow-control configuration for all ports.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring flow control, note the following information:

- Because auto negotiation does not work on 10 Gigbit Ethernet fiber optic ports, they respond to pause frames by default. On 10 Gigbit Ethernet fiber optic ports, the flow-control operational mode is always the same as administrative mode.
- You cannot configure how WS-X6502-10GE 10-Gigabit Ethernet ports respond to pause frames. WS-X6502-10GE 10-Gigabit Ethernet ports are permanently configured to respond to pause frames.
- When configuring how a port responds to pause frames, note the following information:
  - For a Gigabit Ethernet port, when the configuration of a remote port is unknown, you can use the **receive desired** keywords to configure the Gigabit Ethernet port to respond to received pause frames. (Supported only on Gigabit Ethernet ports.)
  - Use the **receive on** keywords to configure a port to respond to received pause frames.
  - Use the **receive off** keywords to configure a port to ignore received pause frames.
- When configuring transmission of pause frames on a port, note the following information:
  - For a Gigabit Ethernet port, when the configuration of the remote ports is unknown, you can use the **send desired** keywords to configure the Gigabit Ethernet port to send pause frames. (Supported only on Gigabit Ethernet ports.)
  - Use the **send on** keywords to configure a port to send pause frames.
  - Use the **send off** keywords to configure a port not to send pause frames.

This example shows how to turn on receive flow control and how to verify the flow-control configuration:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/2
Router(config-if)# flowcontrol receive on
Router(config-if)# end
Router# show interfaces flowcontrol

Interface Send         Receive
Gi1/1     Desired      OFF
Gi1/2     Desired      ON
Fa5/1     Not capable  OFF
<output truncated>
```

Configuring the Port Debounce Timer

The port debounce timer delays notification of a link change. Delayed notification of a link changecan decrease traffic loss due to network reconfiguration.

If the status of a link changes quickly from up to down and then back to up, the port debounce timer suppresses the link status notification. If the link transitions from up to down, but does not come back up, the port debounce timer delays the link status notification.

Delayed link status notification allows a quick port status change and recovery to occur without triggering any of the changes that are necessary when a port stays down. The normal operation of DWDM links sometimes includes quick port status changes and recoveries. Delayed link status notification can also be used to mitigate link flaps because of bad cabling.

You can configure the port debounce timer separately on each LAN port.

Caution Enabling the port debounce timer causes link down detections to be delayed, resulting in loss of traffic during the debouncing period. This situation might affect the convergence and reconvergence of some Layer 2 and Layer 3 protocols.

To configure the debounce timer on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the port to configure.
Step 2	`Router(config-if)#` **`link debounce`** `[`**`time`** *`debounce_time`*`]`	Configures the debounce timer. **Note** The default port debounce timer values provide appropriate link change notification delay on DWDM links.
Step 3	`Router#` **`show interfaces debounce`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring the debounce timer on a port, note the following information:

- The **time** keyword is supported only on fiber Gigabit Ethernet ports. With the **time** keyword, you can increase the port debounce timer value in increments of 100 milliseconds up to 5000 milliseconds on ports operating at 1000 Mpbs over copper media.
- The debounce timer recognizes 10-Gbps copper media and detects media-only changes.

Table 8-2 lists the time delay that occurs before notification of a link change.

Table 8-2 Default Port Debounce Timer Delay Times

Port Type	Debounce Timer Disabled	Debounce Timer Enabled
Ports operating at 10 Mpbs or 100 Mbps:	300 milliseconds	3100 milliseconds
Ports operating at 1000 Mpbs or 10 Gbps over copper media:	300 milliseconds	3100 milliseconds
Ports operating at 1000 Mpbs or 10 Gbps over fiber media (except WS-X6502-10GE ports):	10 milliseconds	100 milliseconds
WS-X6502-10GE ports:	1000 milliseconds	3100 milliseconds

Note The show interfaces debounce command does not display the default value for 10-GigabitEthernet ports when the port debounce timer is disabled.

This example shows how to enable the port debounce timer on Fast Ethernet port 5/12:

```
Router(config)# interface fastethernet 5/12
Router(config-if)# link debounce
Router(config-if)# end
```

This example shows how to display the port debounce timer settings:

```
Router# show interfaces debounce | include enable
Fa5/12  enable          3100
```

Adding a Description for an Interface

You can add a description about an interface to help you remember its function. The description appears in the output of the following commands: **show configuration**, **show running-config**, and **show interfaces**.

To add a description for an interface, perform this task:

Command	Purpose
`Router(config-if)# description string`	Adds a description for an interface.

This example shows how to add a description on Fast Ethernet port 5/5:

```
Router(config)# interface fastethernet 5/5
Router(config-if)# description Channel-group to "Marketing"
```

Understanding Online Insertion and Removal

The online insertion and removal (OIR) feature supported on the Catalyst 6500 series switches allows you to remove and replace modules while the system is online. You can shut down the modules before removal and restart it after insertion without causing other software or interfaces to shut down.

Note Do not remove or install more than one module at a time. After you remove or install a module, check the LEDs before continuing. For module LED descriptions, see the *Catalyst 6500 Series Switch Installation Guide*.

When a module has been removed or installed, the Catalyst 6500 series switch stops processing traffic for the module and scans the system for a configuration change. Each interface type is verified against the system configuration, and then the system runs diagnostics on the new module. There is no disruption to normal operation during module insertion or removal.

The switch can bring only an identical replacement module online. To support OIR of an identical module, the module configuration is not removed from the running-config file when you remove a module.

If the replacement module is different from the removed module, you must configure it before the switch can bring it online.

Layer 2 MAC addresses are stored in an EEPROM, which allows modules to be replaced online without requiring the system to update switching tables and data structures. Regardless of the types of modules installed, the Layer 2 MAC addresses do not change unless you replace the supervisor engine. If you do replace the supervisor engine, the Layer 2 MAC addresses of *all* ports change to those specified in the address allocator on the new supervisor engine.

Monitoring and Maintaining Interfaces

You can perform the tasks in the following sections to monitor and maintain interfaces:

- Monitoring Interface Status, page 8-17
- Clearing Counters on an Interface, page 8-18
- Resetting an Interface, page 8-18
- Shutting Down and Restarting an Interface, page 8-19

Monitoring Interface Status

The software contains commands that you can enter at the EXEC prompt to display information about the interface including the version of the software and the hardware and statistics about interfaces. The following table lists some of the interface monitoring commands. (You can display the complete list of **show** commands by using the **show ?** command at the EXEC prompt.) These commands are described in the *Cisco IOS Interface Command Reference* publication.

To display information about the interface, perform these tasks:

Command	Purpose
`Router# show ibc`	Displays current internal status information.
`Router# show eobc`	Displays current internal out-of-band information.
`Router# show interfaces [type slot/port]`	Displays the status and configuration of all or a specific interface.
`Router# show running-config`	Displays the currently running configuration.
`Router# show rif`	Displays the current contents of the routing information field (RIF) cache.

Command	Purpose
Router# **show protocols** [*type slot/port*]	Displays the global (system-wide) and interface-specific status of any configured protocol.
Router# **show version**	Displays the hardware configuration, software version, the names and sources of configuration files, and the boot images.

This example shows how to display the status of Fast Ethernet port 5/5:

```
Router# show protocols fastethernet 5/5
FastEthernet5/5 is up, line protocol is up
Router#
```

Clearing Counters on an Interface

To clear the interface counters shown with the **show interfaces** command, perform this task:

Command	Purpose
Router# **clear counters** {{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *channel_ID*}}	Clears interface counters.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to clear and reset the counters on Fast Ethernet port 5/5:

```
Router# clear counters fastethernet 5/5
Clear "show interface" counters on this interface [confirm] y
Router#
*Sep 30 08:42:55: %CLEAR-5-COUNTERS: Clear counter on interface FastEthernet5/5
```

The **clear counters** command clears all the current counters from the interface unless the optional arguments specify a specific interface.

Note The **clear counters** command clears counters displayed with the EXEC **show interfaces** command, not counters retrieved using SNMP.

Resetting an Interface

To reset an interface, perform this task:

Command	Purpose
Router# **clear interface** *type*[1] *slot/port*	Resets an interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to reset Fast Ethernet port 5/5:

```
Router# clear interface fastethernet 5/5
Router#
```

Shutting Down and Restarting an Interface

You can shut down an interface, which disables all functions on the specified interface and shows the interface as unavailable on all monitoring command displays. This information is communicated to other network servers through all dynamic routing protocols. The interface is not included in any routing updates.

To shut down an interface and then restart it, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`vlan`** *`vlan_ID`*`}	{`*`type`*[1] *`slot/port`*`}	{`**`port-channel`** *`channel_ID`*`}}`	Selects the interface to be configured.
Step 2	`Router(config-if)#` **`shutdown`**	Shuts down the interface.
Step 3	`Router(config-if)#` **`no shutdown`**	Reenables the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to shut down Fast Ethernet port 5/5:

```
Router(config)# interface fastethernet 5/5
Router(config-if)# shutdown
Router(config-if)#
*Sep 30 08:33:47: %LINK-5-CHANGED: Interface FastEthernet5/5, changed state to
administratively down
```

This example shows how to reenable Fast Ethernet port 5/5:

```
Router(config-if)# no shutdown
Router(config-if)#
*Sep 30 08:36:00: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
```

To check if an interface is disabled, enter the EXEC **show interfaces** command. An interface that has been shut down is shown as administratively down in the **show interfaces** command display.

Checking the Cable Status Using the TDR

You can check the status of copper cables using the time domain reflectometer (TDR). The TDR detects a cable fault by sending a signal through the cable and reading the signal that is reflected back to it. All or part of the signal can be reflected back by any number of cable defects or by the end of the cable itself.

Use the TDR to determine if the cabling is at fault if you cannot establish a link. This test is especially important when replacing an existing switch, upgrading to Gigabit Ethernet, or installing new cables.

Note

- TDR can test cables up to a maximum length of 115 meters.
- For information about which modules support the TDR, see this document:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html
- TDR results are not meaningful for a link that is operating successfully.
- The port must be up before running the TDR test. If the port is down, you cannot enter the **test cable-diagnostics tdr** command successfully, and the following message is displayed:

```
Router# test cable-diagnostics tdr interface gigabitethernet2/12
% Interface Gi2/12 is administratively down
% Use 'no shutdown' to enable interface before TDR test start.
```

To start or stop the TDR test, perform this task:

Command	Purpose
test cable-diagnostics tdr interface {**interface** *interface_number*}	Starts or stops the TDR test.

This example shows how to run the TDR-cable diagnostics:

```
Router # test cable-diagnostics tdr interface gigabitethernet2/1
TDR test started on interface Gi2/1
A TDR test can take a few seconds to run on an interface
Use 'show cable-diagnostics tdr' to read the TDR results.
Router #
```

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring UDLD

This chapter describes how to configure the UniDirectional Link Detection (UDLD) protocol. Release 12.2(33)SXI4 and later releases support fast UDLD, which provides faster detection times.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding UDLD

Normal-mode UDLD classifies a link as unidirectional if the received UDLD packets do not contain information that is correct for the neighbor device. In addition to the functionality of normal mode UDLD, aggressive-mode UDLD puts ports into the errdisabled state if the relationship between two previously synchronized neighbors cannot be reestablished.

These sections describe how UDLD works:

UDLD Overview

The Cisco-proprietary UDLD protocol monitors the physical configuration of the links between devices and ports that support UDLD. UDLD detects the existence of unidirectional links. When a unidirectional link is detected, UDLD puts the affected port into the errdisabled state and alerts the user. UDLD can operate in either normal or aggressive mode.

For example, UDLD can help prevent these problems:

- Spanning tree topology loops caused by unidirectional links
- Incorrect cabling of unbundled fiber strands
- Transceiver or link hardware malfunction
- Incorrect or excessive flooding of packets
- Loss of traffic without notice (also known as black holing)

UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both autonegotiation and UDLD, Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

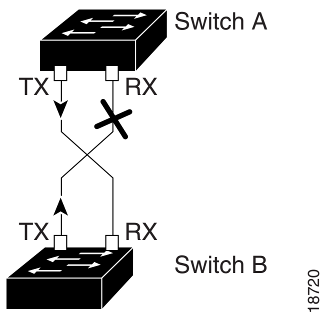
A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor, but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally at Layer 1, then UDLD at Layer 2 determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation, because autonegotiation operates at Layer 1.

LAN ports with UDLD enabled periodically transmit UDLD packets to neighbor devices. If the packets are echoed back within a specific time frame (the message detection time is 3 times as long as the message interval, plus a timeout period) and they are lacking a specific acknowledgment (echo), the link is flagged as unidirectional and the LAN port is shut down. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links.

Note By default, UDLD is disabled on nonfiber LAN ports to avoid sending unnecessary control traffic on this type of media because it is often used for access ports.

Figure 9-1 shows an example of a unidirectional link condition. Switch B successfully receives traffic from Switch A on the port. However, Switch A does not receive traffic from Switch B on the same port. UDLD detects the problem and disables the port.

Figure 9-1 Unidirectional Link



UDLD Aggressive Mode

UDLD aggressive mode is disabled by default. Configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. With UDLD aggressive mode enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD packets, UDLD tries to reestablish the connection with the neighbor. After eight failed retries, the port is disabled.

To prevent spanning tree loops, the default nonaggressive UDLD interval of 15 seconds is fast enough to shut down a unidirectional link before a blocking port transitions to the forwarding state (with default spanning tree parameters).

When you enable UDLD aggressive mode, you receive additional benefits in the following situations:

- One side of a link has a port stuck (both Tx and Rx).
- One side of a link remains up while the other side of the link has gone down.

In these cases, UDLD aggressive mode disables one of the ports on the link, which prevents traffic from being discarding.

Fast UDLD

Release 12.2(33)SXI4 and later releases support fast UDLD.

Fast UDLD is a per-port configuration option that supports UDLD message time intervals between 200 and 1000 milliseconds. Fast UDLD can be configured to provide subsecond unidirectional link detection. (Without fast UDLD, the message time intervals are 7 through 90 seconds).

When configuring fast UDLD, note the following guidelines and restrictions:

- Cisco IOS Software Modularity images do not support fast UDLD.
- Fast UDLD is disabled by default.
- Normal and aggressive mode both support fast UDLD.
- Fast UDLD ports do not support the **link debounce** command.
- Fast UDLD supports only point-to-point links between network devices that support fast UDLD.
- Configure fast UDLD on at least two links between each connected network device. Fast UDLD does not support single-link connections to neighbor devices.
- Fast UDLD does not report a unidirectional link if the same error occurs simultaneously on more than one link to the same neighbor device.
- Fast UDLD cannot detect unidirectional links when the CPU utilization exceeds 60 percent.
- Fast UDLD is supported on a limited number of ports:
  - 60 ports with a Supervisor Engine 720
  - 10 ports with a Supervisor Engine 32
  - 10 ports with a Cisco ME 6500 Series Ethernet Switch (ME6524)

Default UDLD Configuration

Table 9-1 shows the default UDLD configuration.

Table 9-1 UDLD Default Configuration

Feature	Default Value
UDLD global enable state	Globally disabled.
UDLD aggressive mode	Disabled.
Fast UDLD	Disabled.
Fast UDLD error reporting	Disabled.
UDLD per-port enable state for fiber-optic media	Enabled on all Ethernet fiber-optic LAN ports.
UDLD per-port enable state for twisted-pair (copper) media	Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports.

Configuring UDLD

These sections describe how to configure UDLD:

Enabling UDLD Globally

To enable UDLD globally on all fiber-optic LAN ports, perform this task:

Command	Purpose
`Router(config)#` **`udld {enable	aggressive}`**	Enables UDLD globally on fiber-optic LAN ports. **Note** This command only configures fiber-optic LAN ports. Individual LAN port configuration overrides the setting of this command.

Enabling UDLD on LAN Ports

By default, UDLD is disabled on nonfiber-optic LAN ports. To enable UDLD on a LAN port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`udld port [aggressive]`**	Enables UDLD on a specific LAN port. Enter the **aggressive** keyword to enable aggressive mode. On a fiber-optic LAN port, this command overrides the **udld enable** global configuration command setting.
	`Router(config-if)#` **`no udld port [aggressive]`**	Disables UDLD on a nonfiber-optic LAN port. **Note** On a fiber-optic LAN port, the **no udld port** command reverts the LAN port configuration to the **udld enable** global configuration command setting.
Step 3	`Router#` **`show udld`** *`type`*[1] *`slot/number`*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Disabling UDLD on Fiber-Optic LAN Ports

By default, UDLD is enabled on fiber-optic LAN ports. To disable UDLD on a fiber-optic LAN port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects the LAN port to configure.
Step 2	`Router(config-if)# udld port disable`	Disables UDLD on a fiber-optic LAN port. **Note** The **no** form of this command, which reverts the port to the **udld enable** global configuration command setting, is only supported on fiber-optic LAN ports.
Step 3	`Router# show udld type[1] slot/number`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Configuring the Global UDLD Probe Message Interval

To configure the time between UDLD probe messages globally on all ports that are in advertisement mode and are currently determined to be bidirectional, perform this task:

	Command	Purpose
Step 1	`Router(config)# udld message time interval`	Configures the time between UDLD probe messages on ports that are in advertisement mode and are currently determined to be bidirectional; valid values are from 7 to 90 seconds.
Step 2	`Router# show udld type[1] slot/number`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Configuring Fast UDLD

Release 12.2(33)SXI4 and later releases support fast UDLD. These sections describe how to configure fast UDLD:

- Configuring Fast UDLD on a Port, page 9-7
- Enabling Fast UDLD Error Reporting, page 9-7

Note You can configure fast UDLD on ports where UDLD is not enabled, but fast UDLD is active only when UDLD is enabled on the port.

Configuring Fast UDLD on a Port

To configure fast UDLD on a port, perform this task:

	Command	Purpose
Step 1	`Router(config-if)# udld fast-hello interval`	Configures the fast UDLD probe message interval on a port. **Note** • See the guidelines and restrictions in the "Fast UDLD" section on page 9-4. • When selecting the value, follow these guidelines: – Valid values are from 200 to 1000 milliseconds. – Adjust the fast UDLD probe message interval to the longest interval possible that will provide the desired link failure detection time. A shorter message interval increases the likelihood that UDLD will falsely report link failures under stressful conditions.
Step 2	`Router# show udld fast-hello`	Displays fast UDLD configuration and operational state.
Step 3	`Router# show udld fast-hello type`[1] `slot/number`	Verifies the per-port fast UDLD configuration and operational state.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Enabling Fast UDLD Error Reporting

By default, fast UDLD error-disables ports with unidirectional links. You can globally enable fast UDLD to report unidirectional links with a message displayed on the console instead of error-disabling ports with unidirectional links.

Note When fast UDLD error reporting is enabled, you must manually take the action appropriate for the state of the link.

To globally enable fast UDLD error reporting, perform this task:

Command	Purpose
`Router(config)# udld fast-hello error-reporting`	Enables fast UDLD error reporting.

Resetting Disabled LAN Interfaces

To reset all LAN ports that have been shut down by UDLD, perform this task:

Command	Purpose
`Router# udld reset`	Resets all LAN ports that have been shut down by UDLD.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Power Management and Environmental Monitoring

This chapter describes the power management and environmental monitoring features in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

---

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Power Management

These sections describe power management:

Note In systems with redundant power supplies, both power supplies must be of the same wattage. The Catalyst 6500 series switches allow you to use both AC-input and DC-input power supplies in the same chassis. For detailed information on supported power supply configurations, see the *Catalyst 6500 Series Switch Installation Guide*.

The modules have different power requirements, and some configurations require more power than a single power supply can provide. The power management feature allows you to power all installed modules with two power supplies. However, redundancy is not supported in this configuration because the total power drawn from both power supplies is at no time greater than the capability of one supply. Redundant and nonredundant power configurations are described in the following sections.

To determine the power requirements for your system, see the "Determining System Power Requirements" section on page 10-5.

Enabling or Disabling Power Redundancy

To disable or enable redundancy (redundancy is enabled by default) from global configuration mode, enter the **power redundancy-mode combined | redundant** commands. You can change the configuration of the power supplies to redundant or nonredundant at any time.

To disable redundancy, use the **combined** keyword. In a nonredundant configuration, the power available to the system is the combined power capability of both power supplies. The system powers up as many modules as the combined capacity allows. However, if one power supply fails and there is not enough power for all of the previously powered-up modules, the system powers down those modules.

To enable redundancy, use the **redundant** keyword. In a redundant configuration, the total power drawn from both power supplies is not greater than the capability of one power supply. If one supply malfunctions, the other supply can take over the entire system load. When you install and power up two power supplies, each concurrently provides approximately half of the required power to the system. Load sharing and redundancy are enabled automatically; no software configuration is required.

To view the current state of modules and the total power available for modules, enter the **show power** command (see the "Viewing System Power Status" section on page 10-4).

Table 10-1 describes how the system responds to changes in the power supply configuration.

Table 10-1 Effects of Power Supply Configuration Changes

Configuration Change	Effect
Redundant to nonredundant	• System log and syslog messages are generated. • System power is increased to the combined power capability of both power supplies. • Modules marked *power-deny* in the **show power** oper state field are brought up if there is sufficient power.
Nonredundant to redundant (both power supplies must be of equal wattage)	• System log and syslog messages are generated. • System power is decreased to the power capability of one supply. • If there is not enough power for all previously powered-up modules, some modules are powered down and marked as *power-deny* in the **show power** oper state field.

Table 10-1 Effects of Power Supply Configuration Changes (continued)

Configuration Change	Effect
Equal wattage power supply is inserted with redundancy enabled	• System log and syslog messages are generated. • System power equals the power capability of one supply. • No change in module status because the power capability is unchanged.
Equal wattage power supply is inserted with redundancy disabled	• System log and syslog messages are generated. • System power is increased to the combined power capability of both power supplies. • Modules marked *power-deny* in the **show power** oper state field are brought up if there is sufficient power.
Higher or lower wattage power supply is inserted with redundancy enabled	• System log and syslog messages are generated. • The system does not allow you to operate a power supply of different wattage even if the wattage is higher than the installed supply. The inserted supply shuts down.
Higher or lower wattage power supply is inserted with redundancy disabled	• System log and syslog messages are generated. • System power is increased to the combined power capability of both power supplies. • Modules marked *power-deny* in the **show power** oper state field are brought up if there is sufficient power.
Power supply is removed with redundancy enabled	• System log and syslog messages are generated. • No change in module status because the power capability is unchanged.
Power supply is removed with redundancy disabled	• System log and syslog messages are generated. • System power is decreased to the power capability of one supply. • If there is not enough power for all previously powered-up modules, some modules are powered down and marked as *power-deny* in the **show power** oper state field.
System is booted with power supplies of different wattage installed and redundancy enabled	• System log and syslog messages are generated. • The system does not allow you to have power supplies of different wattage installed in a redundant configuration. The lower wattage supply shuts down.
System is booted with power supplies of equal or different wattage installed and redundancy disabled	• System log and syslog messages are generated. • System power equals the combined power capability of both power supplies. • The system powers up as many modules as the combined capacity allows.

Powering Modules Off and On

To power modules off and on from the CLI, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`power enable module`** *`slot_number`*	Powers a module on.
Step 3	`Router(config)#` **`no power enable module`** *`slot_number`*	Powers a module off.

Note When you enter the **no power enable module** *slot* command to power down a module, the module's configuration is not saved.

This example shows how to power on the module in slot 3:

```
Router# configure terminal
Router(config)# power enable module 3
```

Viewing System Power Status

You can view the current power status of system components by entering the **show power** command as follows:

```
Router# show power
system power redundancy mode = redundant
system power total =     1153.32 Watts (27.46 Amps @ 42V)
system power used =       397.74 Watts ( 9.47 Amps @ 42V)
system power available =  755.58 Watts (17.99 Amps @ 42V)
                        Power-Capacity PS-Fan Output Oper
PS   Type               Watts   A @42V Status Status State
---- ------------------ ------- ------ ------ ------ -----
1    WS-CAC-2500W       1153.32 27.46  OK     OK     on
2    none
                        Pwr-Requested  Pwr-Allocated  Admin Oper
Slot Card-Type          Watts   A @42V Watts   A @42V State State
---- ------------------ ------- ------ ------- ------ ----- -----
1    WS-X6K-SUP2-2GE     142.38  3.39   142.38  3.39  on    on
2                          -     -      142.38  3.39  -     -
5    WS-X6248-RJ-45      112.98  2.69   112.98  2.69  on    on
Router#
```

You can view the current power status of a specific power supply by entering the **show power** command as follows:

```
Router# show power status power-supply 2
                        Power-Capacity PS-Fan Output Oper
PS   Type               Watts   A @42V Status Status State
---- ------------------ ------- ------ ------ ------ -----
1    WS-CAC-6000W       2672.04  63.62 OK     OK     on
2    WS-CAC-9000W-E     2773.68  66.04 OK     OK     on
Router#
```

You can display power supply input fields by specifying the power supply number in the command. A new power-output field with operating mode is displayed for power supplies with more than one output mode. Enter the **show env status power-supply** command as follows:

```
Router# show env status power-supply 1
power-supply 1:
  power-supply 1 fan-fail: OK
  power-supply 1 power-input 1: AC low
  power-supply 1 power-output-fail: OK
Router# show env status power-supply 2
power-supply 2:
  power-supply 2 fan-fail: OK
  power-supply 2 power-input 1: none<<< new
  power-supply 2 power-input 2: AC low<<< new
  power-supply 2 power-input 3: AC high<<< new
  power-supply 2 power-output: low (mode 1)<<< high for highest mode only
  power-supply 2 power-output-fail: OK
```

Power Cycling Modules

You can power cycle (reset) a module from global configuration mode by entering the **power cycle module** *slot* command. The module powers off for 5 seconds, and then powers on.

Determining System Power Requirements

For information about power consumption, see the *Release Notes for Cisco IOS Release 12.2SX* publication at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html

Determining System Hardware Capacity

You can determine the system hardware capacity by entering the **show platform hardware capacity** command. This command displays the current system utilization of the hardware resources and displays a list of the currently available hardware capacities, including the following:

- Hardware forwarding table utilization
- Switch fabric utilization
- CPU(s) utilization
- Memory device (flash, DRAM, NVRAM) utilization

This example shows how to display CPU capacity and utilization information for the route processor, the switch processor, and a switching module:

```
Router# show platform hardware capacity cpu
CPU Resources
  CPU utilization: Module             5 seconds       1 minute       5 minutes
                    1  RP              0% /  0%             1%              1%
                    1  SP              5% /  0%             5%              4%
                    7                 69% /  0%            69%             69%
                    8                 78% /  0%            74%             74%

  Processor memory: Module   Bytes:       Total           Used        %Used
                     1  RP            176730048       51774704        29%
                     1  SP            192825092       51978936        27%
                     7                195111584       35769704        18%
                     8                195111584       35798632        18%
  I/O memory: Module         Bytes:       Total           Used        %Used
               1  RP                   35651584       12226672        34%
               1  SP                   35651584        9747952        27%
               7                       35651584        9616816        27%
               8                       35651584        9616816        27%
Router#
```

This example shows how to display EOBC-related statistics for the route processor, the switch processor, and the DFCs:

```
Router# show platform hardware capacity eobc EOBC Resources
  Module                     Packets/sec     Total packets     Dropped packets
  1  RP        Rx:                    61            108982                   0
               Tx:                    37             77298                   0
  1  SP        Rx:                    34            101627                   0
               Tx:                    39            115417                   0
  7            Rx:                     5             10358                   0
```

```
            Tx:                     8            18543                0
  8         Rx:                     5            12130                0
            Tx:                    10            20317                0
Router#
```

This example shows how to display the current and peak switching utilization:

```
Router# show platform hardware capacity fabric Switch Fabric Resources
  Bus utilization: current is 100%, peak was 100% at 12:34 12mar45
  Fabric utilization:      ingress                     egress
      Module channel speed current peak                current peak
       1      0        20G   100%  100% 12:34 12mar45  100%    100% 12:34 12mar45
       1      1        20G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
       4      0        20G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
      13      0         8G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
Router#
```

This example shows how to display information about the total capacity, the bytes used, and the percentage that is used for the flash and NVRAM resources present in the system:

```
Router# show platform hardware capacity flash
Flash/NVRAM Resources
  Usage: Module Device              Bytes:      Total          Used     %Used
           1  RP  bootflash:                 31981568      15688048       49%
           1  SP  disk0:                    128577536     105621504       82%
           1  SP  sup-bootflash:             31981568      29700644       93%
           1  SP  const_nvram:                 129004           856        1%
           1  SP  nvram:                       391160         22065        6%
           7      dfc#7-bootflash:           15204352        616540        4%
           8      dfc#8-bootflash:           15204352             0        0%
Router#
```

This example shows how to display the capacity and utilization of the EARLs present in the system:

```
Router# show platform hardware capacity forwarding
L2 Forwarding Resources
           MAC Table usage:   Module  Collisions  Total       Used     %Used
                              6                0  65536         11        1%
             VPN CAM usage:                       Total       Used     %Used
                                                    512          0        0%
L3 Forwarding Resources
             FIB TCAM usage:                     Total        Used     %Used
                  72 bits (IPv4, MPLS, EoM)     196608          36        1%
                 144 bits (IP mcast, IPv6)       32768           7        1%

                     detail:      Protocol                    Used     %Used
                                  IPv4                          36        1%
                                  MPLS                           0        0%
                                  EoM                            0        0%

                                  IPv6                           4        1%
                                  IPv4 mcast                     3        1%
                                  IPv6 mcast                     0        0%

            Adjacency usage:                     Total        Used     %Used
                                               1048576         175        1%

     Forwarding engine load:
                     Module       pps   peak-pps  peak-time
                     6              8       1972  02:02:17 UTC Thu Apr 21 2005

Netflow Resources
```

```
          TCAM utilization:       Module       Created      Failed     %Used
                                  6                  1           0        0%
          ICAM utilization:       Module       Created      Failed     %Used
                                  6                  0           0        0%

                 Flowmasks:   Mask#   Type        Features
                        IPv4:     0   reserved    none
                        IPv4:     1   Intf FulNAT_INGRESS NAT_EGRESS FM_GUARDIAN
                        IPv4:     2   unused      none
                        IPv4:     3   reserved    none

                        IPv6:     0   reserved    none
                        IPv6:     1   unused      none
                        IPv6:     2   unused      none
                        IPv6:     3   reserved    none

CPU Rate Limiters Resources
             Rate limiters:       Total         Used      Reserved     %Used
                    Layer 3           9            4             1       44%
                    Layer 2           4            2             2       50%

ACL/QoS TCAM Resources
  Key: ACLent - ACL TCAM entries, ACLmsk - ACL TCAM masks, AND - ANDOR,
       QoSent - QoS TCAM entries, QOSmsk - QoS TCAM masks, OR - ORAND,
       Lbl-in - ingress label, Lbl-eg - egress label, LOUsrc - LOU source,
       LOUdst - LOU destination, ADJ - ACL adjacency

  Module ACLent ACLmsk QoSent QoSmsk Lbl-in Lbl-eg LOUsrc LOUdst  AND OR  ADJ
  6          1%     1%     1%     1%     1%     1%     0%     0%   0% 0%   1%

Router#
```

This example shows how to display the interface resources:

```
Router# show platform hardware capacity interface Interface Resources
  Interface drops:
    Module    Total drops:    Tx            Rx      Highest drop port: Tx  Rx
    9                          0             2                          0  48

  Interface buffer sizes:
    Module                           Bytes:     Tx buffer           Rx buffer
         1                                          12345               12345
         5                                          12345               12345
Router#
```

This example shows how to display SPAN information:

```
Router# show platform hardware capacity monitor SPAN Resources
  Source sessions: 2 maximum, 0 used
    Type                                   Used
    Local                                     0
    RSPAN source                              0
    ERSPAN source                             0
    Service module                            0
  Destination sessions: 64 maximum, 0 used
    Type                                   Used
    RSPAN destination                         0
    ERSPAN destination (max  24)              0
Router#
```

This example shows how to display the capacity and utilization of resources for Layer 3 multicast functionality:

```
Router# show platform hardware capacity multicast
L3 Multicast Resources
  IPv4 replication mode: ingress
  IPv6 replication mode: ingress
  Bi-directional PIM Designated Forwarder Table usage: 4 total, 0 (0%) used
  Replication capability: Module                              IPv4     IPv6
                          5                                 egress   egress
                          9                                ingress  ingress
  MET table Entries: Module                             Total    Used    %Used
                     5                                  65526       6       0%
Router#
```

This example shows how to display information about the system power capacities and utilizations:

```
Router# show platform hardware capacity power
Power Resources
  Power supply redundancy mode: administratively combined operationally combined
  System power: 1922W, 0W (0%) inline, 1289W (67%) total allocated
  Powered devices: 0 total
Router#
```

This example shows how to display the capacity and utilization of QoS policer resources for each PFC and DFC:

```
Router# show platform hardware capacity qos
QoS Policer Resources
  Aggregate policers: Module                      Total         Used     %Used
                      1                            1024          102       10%
                      5                            1024            1        1%
  Microflow policer configurations: Module        Total         Used      %Used
                                    1                64           32        50%
                                    5                64            1         1%
Router#
```

This example shows how to display information about the key system resources:

```
Router# show platform hardware capacity systems System Resources
  PFC operating mode: PFC3BXL
  Supervisor redundancy mode: administratively rpr-plus, operationally rpr-plus
  Switching Resources: Module   Part number              Series       CEF mode
                       5        WS-SUP720-BASE        supervisor            CEF
                       9        WS-X6548-RJ-45            CEF256            CEF
Router#
```

This example shows how to display VLAN information:

```
Router# show platform hardware capacity vlan VLAN Resources
  VLANs: 4094 total, 10 VTP, 0 extended, 0 internal, 4084 free Router#
```

Determining Sensor Temperature Threshold

The system sensors set off alarms based on different temperature threshold settings. You can determine the allowed temperatures for the sensors by using the **show environment alarm threshold** command.

This example shows how to determine sensor temperature thresholds:

```
Router> show environment alarm threshold
environmental alarm thresholds:

power-supply 1 fan-fail: OK
  threshold #1 for power-supply 1 fan-fail:
    (sensor value != 0) is system minor alarm power-supply 1 power-output-fail: OK
  threshold #1 for power-supply 1 power-output-fail:
    (sensor value != 0) is system minor alarm fantray fan operation sensor: OK
  threshold #1 for fantray fan operation sensor:
    (sensor value != 0) is system minor alarm operating clock count: 2
  threshold #1 for operating clock count:
    (sensor value < 2) is system minor alarm
  threshold #2 for operating clock count:
    (sensor value < 1) is system major alarm operating VTT count: 3
  threshold #1 for operating VTT count:
    (sensor value < 3) is system minor alarm
  threshold #2 for operating VTT count:
    (sensor value < 2) is system major alarm VTT 1 OK: OK
  threshold #1 for VTT 1 OK:
    (sensor value != 0) is system minor alarm VTT 2 OK: OK
  threshold #1 for VTT 2 OK:
    (sensor value != 0) is system minor alarm VTT 3 OK: OK
  threshold #1 for VTT 3 OK:
    (sensor value != 0) is system minor alarm clock 1 OK: OK
  threshold #1 for clock 1 OK:
    (sensor value != 0) is system minor alarm clock 2 OK: OK
  threshold #1 for clock 2 OK:
    (sensor value != 0) is system minor alarm module 1 power-output-fail: OK
  threshold #1 for module 1 power-output-fail:
    (sensor value != 0) is system major alarm module 1 outlet temperature: 21C
  threshold #1 for module 1 outlet temperature:
    (sensor value > 60) is system minor alarm
  threshold #2 for module 1 outlet temperature:
    (sensor value > 70) is system major alarm module 1 inlet temperature: 25C
  threshold #1 for module 1 inlet temperature:
    (sensor value > 60) is system minor alarm
  threshold #2 for module 1 inlet temperature:
    (sensor value > 70) is system major alarm module 1 device-1 temperature: 30C
  threshold #1 for module 1 device-1 temperature:
    (sensor value > 60) is system minor alarm
  threshold #2 for module 1 device-1 temperature:
    (sensor value > 70) is system major alarm module 1 device-2 temperature: 29C
  threshold #1 for module 1 device-2 temperature:
    (sensor value > 60) is system minor alarm
  threshold #2 for module 1 device-2 temperature:
    (sensor value > 70) is system major alarm module 5 power-output-fail: OK
  threshold #1 for module 5 power-output-fail:
    (sensor value != 0) is system major alarm module 5 outlet temperature: 26C
  threshold #1 for module 5 outlet temperature:
    (sensor value > 60) is system minor alarm
  threshold #2 for module 5 outlet temperature:
    (sensor value > 75) is system major alarm module 5 inlet temperature: 23C
  threshold #1 for module 5 inlet temperature:
    (sensor value > 50) is system minor alarm
  threshold #2 for module 5 inlet temperature:
    (sensor value > 65) is system major alarm EARL 1 outlet temperature: N/O
```

```
  threshold #1 for EARL 1 outlet temperature:
    (sensor value > 60) is system minor alarm
  threshold #2 for EARL 1 outlet temperature:
    (sensor value > 75) is system major alarm EARL 1 inlet temperature: N/O
  threshold #1 for EARL 1 inlet temperature:
    (sensor value > 50) is system minor alarm
  threshold #2 for EARL 1 inlet temperature:
    (sensor value > 65) is system major alarm
```

Understanding Environmental Monitoring

These sections describe environmental monitoring:

Overview

Environmental monitoring of chassis components provides early-warning indications of possible component failures, which ensures a safe and reliable system operation and avoids network interruptions. This section describes the monitoring of these critical system components, which allows you to identify and rapidly correct hardware-related problems in your system.

Monitoring System Environmental Status

To display system status information, enter the **show environment** [**alarm** | **cooling** | **status** | **temperature**] command. The keywords display the following information:

- **alarm**—Displays environmental alarms.
  - **status**—Displays alarm status.
  - **thresholds**—Displays alarm thresholds.
- **cooling—**Displays fan tray status, chassis cooling capacity, ambient temperature, and per-slot cooling capacity.
- **status**—Displays field-replaceable unit (FRU) operational status and power and temperature information.
- **temperature—**Displays FRU temperature information.

To view the system status information, enter the **show environment** command:

```
Router# show environment
environmental alarms:
  no alarms

Router# show environment alarm
environmental alarms:
  no alarms

Router# show environment cooling
fan-tray 1:
  fan-tray 1 fan-fail: failed
fan-tray 2:
```

```
  fan 2 type: FAN-MOD-9
  fan-tray 2 fan-fail: OK
chassis cooling capacity: 690 cfm
ambient temperature: 55C                      ["40C (user-specified)" if temp-controlled]
chassis per slot cooling capacity: 75 cfm

  module 1 cooling requirement: 70 cfm
  module 2 cooling requirement: 70 cfm
  module 5 cooling requirement: 30 cfm
  module 6 cooling requirement: 70 cfm
  module 8 cooling requirement: 70 cfm
  module 9 cooling requirement: 30 cfm

Router# show environment status
backplane:
  operating clock count: 2
  operating VTT count: 3
fan-tray 1:
  fan-tray 1 type: WS-9SLOT-FAN
  fan-tray 1 fan-fail: OK
VTT 1:
  VTT 1 OK: OK
  VTT 1 outlet temperature: 33C
VTT 2:
  VTT 2 OK: OK
  VTT 2 outlet temperature: 35C
VTT 3:
  VTT 3 OK: OK
  VTT 3 outlet temperature: 33C
clock 1:
  clock 1 OK: OK, clock 1 clock-inuse: in-use
clock 2:
  clock 2 OK: OK, clock 2 clock-inuse: not-in-use
power-supply 1:
  power-supply 1 fan-fail: OK
  power-supply 1 power-output-fail: OK
module 1:
  module 1 power-output-fail: OK
  module 1 outlet temperature: 30C
  module 1 device-2 temperature: 35C
  RP 1 outlet temperature: 35C
  RP 1 inlet temperature: 36C
  EARL 1 outlet temperature: 33C
  EARL 1 inlet temperature: 31C
module 2:
  module 2 power-output-fail: OK
  module 2 outlet temperature: 31C
  module 2 inlet temperature: 29C
module 3:
  module 3 power-output-fail: OK
  module 3 outlet temperature: 36C
  module 3 inlet temperature: 29C
module 4:
  module 4 power-output-fail: OK
  module 4 outlet temperature: 32C
  module 4 inlet temperature: 32C
module 5:
  module 5 power-output-fail: OK
  module 5 outlet temperature: 39C
  module 5 inlet temperature: 34C
module 7:
  module 7 power-output-fail: OK
  module 7 outlet temperature: 42C
  module 7 inlet temperature: 29C
```

```
  EARL 7 outlet temperature: 45C
  EARL 7 inlet temperature: 32C
module 9:
  module 9 power-output-fail: OK
  module 9 outlet temperature: 41C
  module 9 inlet temperature: 36C
  EARL 9 outlet temperature: 33C
  EARL 9 inlet temperature: N/O
```

Understanding LED Environmental Indications

All modules have an LED labeled STATUS. There are LEDs on power supplies and fan trays that indicate problems. A nongreen LED indicates a problem.

Supervisor engines have an LED labeled SYSTEM that indicates alarms. The alarm can be on a supervisor engine, module, power supply, fan tray, or backplane.

In non-VSS mode, the SYSTEM LED indicates the following:

- With one supervisor engine, the SYSTEM LED on the active supervisor engine indicates the alarm status for the chassis.
- With two supervisor engines, the SYSTEM LED on the standby supervisor engine indicates the alarm status of the standby supervisor engine.

In VSS mode, the SYSTEM LED indicates the following:

- With one supervisor engine in each chassis, the SYSTEM LED on the active chassis supervisor engine indicates the alarm status for both chassis. The SYSTEM LED on the standby chassis supervisor engine indicates the alarm status of the standby chassis.
- With two supervisor engines in each chassis, the SYSTEM LED on the active-chassis, in-chassis active supervisor engine indicates system status for both chassis.
- With two supervisor engines in each chassis, the SYSTEM LED on the standby-chassis in-chassis active supervisor engine indicates the alarm status of the standby chassis.
- With two supervisor engines in each chassis, the SYSTEM LEDs on in-chassis standby supervisor engines are turned off.

The SYSTEM LED can indicate two alarm types: major and minor. Major alarms indicate a critical problem that could lead to the system being shut down. Minor alarms are for informational purposes only, alerting you to a problem that could turn critical if corrective action is not taken.

Temperature sensors monitor key components. The temperature sensors are polled every 30 seconds. If the sensed temperature falls below the alarm threshold, the alarm is immediately cancelled.

For major alarms, there is a delay before any automatic actions occur to protect the switch. The delay is 5 minutes for a major alarm from a board sensor, and a 1 minute for a major alarm from an ASIC sensor.

Table 10-2 lists the environmental indicators for the supervisor engine and switching modules.

Note See the *Catalyst 6500 Series Switch Supervisor Engine Guide* for additional information on LEDs, including the supervisor engine STATUS LED.

Table 10-2 Environmental Monitoring for Supervisor Engine and Switching Modules

Component	Alarm Type	SYSTEM LED Color	Action
VSS mode supervisor engine temperature sensor exceeds major threshold	Major	Red	Generates syslog message and an SNMP trap. After the time delay, these actions happen: • With a redundant supervisor engine in the same chassis, the peer chassis becomes active and the previously active supervisor engine drops to ROMMON. With Release 12.2(33)SXI4 and later releases, if the fan tray is faulty or absent, the chassis with fault shuts down. The peer chassis continues operating. • With a single supervisor engine in the chassis, the supervisor engine drops to ROMMON and the peer chassis becomes active. With Release 12.2(33)SXI4 and later releases, if the fan tray is faulty or absent, the faulty chassis shuts down.
Non-VSS mode supervisor engine temperature sensor exceeds major threshold	Major	Red	Generates syslog message and an SNMP trap. After the time delay, these actions happen: • With a redundant supervisor engine in the same chassis, the redundant supervisor engine becomes active and the previously active supervisor engine drops to ROMMON. With Release 12.2(33)SXI4 and later releases, if the fan tray is faulty or absent, the chassis shuts down. • With a single supervisor engine, the supervisor engine drops to ROMMON. With Release 12.2(33)SXI4 and later releases, if the fan tray is faulty or absent, the chassis shuts down.
Supervisor engine temperature sensor exceeds minor threshold	Minor	Orange	Generates syslog message and an SNMP trap. Monitors the condition.
Redundant supervisor engine temperature sensor exceeds major or minor threshold	Major	Red	Generates syslog message and an SNMP trap. If a major alarm is generated and the overtemperature condition is not corrected, the system shuts down after 5 minutes.
	Minor	Orange	Monitors the condition if a minor alarm is generated.
Switching module temperature sensor exceeds major threshold	Major	Red	Generates syslog message and SNMP. Powers down the module (see the "Understanding Power Management" section on page 10-1).
Switching module temperature sensor exceeds minor threshold	Minor	Orange	Generates syslog message and an SNMP trap. Monitors the condition.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring EnergyWise

EnergyWise is a Cisco energy management architecture that provides a common approach to configuring, reporting and managing the power consumed by Cisco switches and attached devices. With Cisco EnergyWise, power can be managed on a network, sub-network or network element level.

Release	Feature Modification
12.2(33)SXI4	EnergyWise Phase2 introduced on Catalyst 6500 Series

EnergyWise configuration procedures for the Catalyst 6500 Series are documented in the "*Cisco EnergyWise Configuration Guide, EnergyWise Phase 2*" located at the following URL:

http://www.cisco.com/en/US/docs/switches/lan/energywise/phase2/ios/configuration/guide/ew_v2.html

Additional Cisco EnergyWise documents, such as White Papers, Data Sheets, FAQs, andare located at the following URL:

http://www.cisco.com/en/US/products/ps10195/

Cisco EnergyWise Orchestrator configuration guides are located at the following URL:

http://www.cisco.com/en/US/products/ps10195/products_installation_and_configuration_guides_list.html

Configuring Online Diagnostics

This chapter describes how to configure the online diagnostics in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Online Diagnostics

With online diagnostics, you can test and verify the hardware functionality of the switch while the switch is connected to a live network.

The online diagnostics contain packet switching tests that check different hardware components and verify the data path and control signals. Disruptive online diagnostic tests, such as the built-in self-test (BIST) and the disruptive loopback test, and nondisruptive online diagnostic tests, such as packet switching, run during bootup, module online insertion and removal (OIR), and system reset. The nondisruptive online diagnostic tests run as part of background health monitoring. Either disruptive or nondisruptive tests can be run at the user's request (on-demand).

The online diagnostics detect problems in the following areas:

- Hardware components

- Interfaces (GBICs, Ethernet ports, and so forth)
- Connectors (loose connectors, bent pins, and so forth)
- Solder joints
- Memory (failure over time)

Online diagnostics is one of the requirements for the high availability feature. High availability is a set of quality standards that seek to limit the impact of equipment failures on the network. A key part of high availability is detecting hardware failures and taking corrective action while the switch runs in a live network. Online diagnostics in high availability detect hardware failures and provide feedback to high availability software components to make switchover decisions.

Online diagnostics are categorized as bootup, on-demand, schedule, or health-monitoring diagnostics. Bootup diagnostics run during bootup; on-demand diagnostics run from the CLI; schedule diagnostics run at user-designated intervals or specified times when the switch is connected to a live network; and health-monitoring runs in the background.

Configuring Online Diagnostics

These sections describe how to configure online diagnostics:

- Setting Bootup Online Diagnostics Level, page 12-2
- Configuring On-Demand Online Diagnostics, page 12-3
- Scheduling Online Diagnostics, page 12-4

Setting Bootup Online Diagnostics Level

You can set the bootup diagnostics level as minimal or complete or you can bypass the bootup diagnostics entirely. Enter the **complete** keyword to run all diagnostic tests; enter the **minimal** keyword to run only EARL tests and loopback tests for all ports in the switch. Enter the **no** form of the command to bypass all diagnostic tests. The default bootup diagnositcs level is minimal.

To set the bootup diagnostic level, perform this task:

Command	Purpose
Router(config)# **diagnostic bootup level {minimal | complete}**	Sets the bootup diagnostic level.

This example shows how to set the bootup online diagnostic level:

```
Router(config)# diagnostic bootup level complete
Router(config)#
```

This example shows how to display the bootup online diagnostic level:

```
Router(config)# do show diagnostic bootup level
Router(config)#
```

Configuring On-Demand Online Diagnostics

You can run the on-demand online diagnostic tests from the CLI. You can set the execution action to either stop or continue the test when a failure is detected or to stop the test after a specific number of failures occur by using the failure count setting. You can configure a test to run multiple times using the iteration setting.

You should run packet-switching tests before memory tests.

Note Do not use the **diagnostic start all** command until all of the following steps are completed.

Because some on-demand online diagnostic tests can affect the outcome of other tests, you should perform the tests in the following order:

1. Run the nondisruptive tests.
2. Run all tests in the relevant functional area.
3. Run the TestTrafficStress test.
4. Run the TestEobcStressPing test.
5. Run the exhaustive-memory tests.

To run on-demand online diagnostic tests, perform this task:

Step 1 Run the nondisruptive tests.

To display the available tests and their attributes, and determine which commands are in the nondisruptive category, enter the **show diagnostic content** command.

Step 2 Run all tests in the relevant functional area.

Packet-switching tests fall into specific functional areas. When a problem is suspected in a particular functional area, run all tests in that functional area. If you are unsure about which functional area you need to test, or if you want to run all available tests, enter the **complete** keyword.

Step 3 Run the TestTrafficStress test.

This is a disruptive packet-switching test. This test switches packets between pairs of ports at line rate for the purpose of stress testing. During this test all of the ports are shut down, and you may see link flaps. The link flaps will recover after the test is complete. The test takes several minutes to complete.

Disable all health-monitoring tests f before running this test by using the **no diagnostic monitor module** *number* **test all** command.

Step 4 Run the TestEobcStressPing test.

This is a disruptive test and tests the Ethernet over backplane channel (EOBC) connection for the module. The test takes several minutes to complete. You cannot run any of the packet-switching tests described in previous steps after running this test. However, you can run tests described in subsequent steps after running this test.

Disable all health-monitoring tests before running this test by using the **no diagnostic monitor module** *number* **test all** command. The EOBC connection is disrupted during this test and will cause the health-monitoring tests to fail and take recovery action.

Step 5 Run the exhaustive-memory tests.

Before running the exhaustive-memory tests, all health-monitoring tests should be disabled because the tests will fail with health monitoring enabled and the switch will take recovery action. Disable the health-monitoring diagnostic tests by using the **no diagnostic monitor module** *number* **test all** command.

Perform the exhaustive-memory tests in the following order:

1. TestFibTcamSSRAM
2. TestAclQosTcam
3. TestNetFlowTcam
4. TestAsicMemory
5. TestAsicMemory

You must reboot the after running the exhaustive-memory tests before it is operational again. You cannot run any other tests on the switch after running the exhaustive-memory tests. Do not save the configuration when rebooting as it will have changed during the tests. After the reboot, reenable the health-monitoring tests using the **diagnostic monitor module** *number* **test all** command.

To set the bootup diagnostic level, perform this task:

Command	Purpose
Router# **diagnostic ondemand** {**iteration** *iteration_count*} | {**action-on-error {continue** | **stop**}[*error_count*]}	Configures on-demand diagnostic tests to run, how many times to run (iterations), and what action to take when errors are found.

This example shows how to set the on-demand testing iteration count:

```
Router# diagnostic ondemand iteration 3
Router#
```

This example shows how to set the execution action when an error is detected:

```
Router# diagnostic ondemand action-on-error continue 2
Router#
```

Scheduling Online Diagnostics

You can schedule online diagnostics to run at a designated time of day or on a daily, weekly, or monthly basis. You can schedule tests to run only once or to repeat at an interval. Use the **no** form of this command to remove the scheduling.

To schedule online diagnostics, perform this task:

Command	Purpose
Router(config)# **diagnostic schedule module** *number* **test** {*test_id* | *test_id_range* | **all**} [**port** {*num* | *num_range* | **all**}] {**on** *mm dd yyyy hh*:*mm*} | {**daily** *hh*:*mm*} | {**weekly** *day_of_week hh*:*mm*}	Schedules on-demand diagnostic tests on the specified module for a specific date and time, how many times to run (iterations), and what action to take when errors are found.

This example shows how to schedule diagnostic testing on a specific date and time for a specific port on module 1:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 on january 3 2003 23:32
Router(config)#
```

This example shows how to schedule diagnostic testing to occur daily at a certain time for a specific port:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 daily 12:34
Router(config)#
```

This example shows how to schedule diagnostic testing to occur weekly on a certain day for a specific port:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 weekly friday 09:23
Router(config)#
```

Configuring Health-Monitoring Diagnostics

You can configure health-monitoring diagnostic testing while the switch is connected to a live network. You can configure the execution interval for each health-monitoring test, the generation of a system message upon test failure, or the enabling or disabling an individual test. Use the **no** form of this command to disable testing.

To configure health-monitoring diagnostic testing, perform this task:

	Command	Purpose
Step 1	Router(config)# **diagnostic monitor interval module** *number* **test** {*test_id* | *test_id_range* | **all**} [**hour** *hh*] [**min** *mm*] [**second** *ss*] [**millisec** *ms*] [**day** *day*]	Configures the health-monitoring interval of the specified tests. The **no** form of this command will change the interval to the default interval, or zero.
Step 2	Router(config)#[**no**] **diagnostic monitor module** *number* **test** {*test_id* | *test_id_range* | **all**}	Enables or disables health-monitoring diagnostic tests.
Step 3	Router# **show diagnostic health**	Displays the output for the health checks performed.

This example shows how to configure the specified test to run every two minutes on module 1:

```
Router(config)# diagnostic monitor interval module 1 test 1 min 2
Router(config)#
```

This example shows how to run the test if health monitoring has not previously been enabled:

```
Router(config)# diagnostic monitor module 1 test 1
```

This example shows how to enable the generation of a syslog message when any health-monitoring test fails:

```
Router(config)# diagnostic monitor syslog
Router(config)#
```

Running Online Diagnostic Tests

After you configure online diagnostics, you can start or stop diagnostic tests or display the test results. You can also see which tests are configured and what diagnostic tests have already run.

These sections describe how to run online diagnostic tests after they have been configured:

- Starting and Stopping Online Diagnostic Tests, page 12-6

- Running All Online Diagnostic Tests, page 12-7
- Displaying Online Diagnostic Tests and Test Results, page 12-7

Note
- We recommend that before you enable any online diagnostics tests that you enable the logging console/monitor to see all warning messages.
- We recommend that when you are running disruptive tests that you only run the tests when connected through the console. When disruptive tests are complete, a warning message on the console recommends that you reload the system to return to normal operation. Strictly follow this warning.
- While tests are running, all ports are shut down because a stress test is being performed with ports configured to loop internally; external traffic might alter the test results. The switch must be rebooted to bring the switch to normal operation. When you issue the command to reload the switch, the system will ask you if the configuration should be saved. Do not save the configuration.
- If you are running the tests on a supervisor engine, after the test is initiated and complete, you must reload or power down and then power up the entire system.
- If you are running the tests on a switching module, rather than the supervisor engine, after the test is initiated and complete, you must reset the switching module.

Starting and Stopping Online Diagnostic Tests

After you configure diagnostic tests to run, you can use the **start** and **stop** to begin or end a diagnostic test.

To start or stop an online diagnostic command, perform one of these tasks:

Command	Purpose
Router# **diagnostic start module** *number* **test** {*test_id* | *test_id_range* | **minimal** | **complete** | **basic** | **per-port** | **non-disruptive** | **all**} **[port** {*num* | *port#_range* | **all**}]	Starts a diagnostic test on a port or range of ports on the specified module.
Router# **diagnostic stop module** *number*	Stops a diagnostic test on the specified module.

This example shows how to start a diagnostic test on module 1:

```
Router# diagnostic start module 1 test 5
Module 1:Running test(s) 5 may disrupt normal system operation
Do you want to run disruptive tests? [no]yes
00:48:14:Running OnDemand Diagnostics [Iteration #1] ...
00:48:14:%DIAG-SP-6-TEST_RUNNING:Module 1:Running TestNewLearn{ID=5} ...
00:48:14:%DIAG-SP-6-TEST_OK:Module 1:TestNewLearn{ID=5} has completed successfully
00:48:14:Running OnDemand Diagnostics [Iteration #2] ...
00:48:14:%DIAG-SP-6-TEST_RUNNING:Module 1:Running TestNewLearn{ID=5} ...
00:48:14:%DIAG-SP-6-TEST_OK:Module 1:TestNewLearn{ID=5} has completed successfully
Router#
```

This example shows how to stop a diagnostic test:

```
Router# diagnostic stop module 1
Router#
```

Running All Online Diagnostic Tests

You can run all diagnostic tests, disruptive and nondisruptive, at once with a single command. In this case, all test dependencies will be handled automatically.

Note
- Running all online diagnostic tests will disrupt normal system operation. Reset the system after the **diagnostic start system test all** command has completed.
- Do not insert, remove, or power down modules or the supervisor while the system test is running.
- Do not issue any diagnostic command other than the **diagnostic stop system test all** command while the system test is running.
- Make sure no traffic is running in background.

To start or stop all online diagnostic tests, perform one of these tasks:

Command	Purpose
`Router# diagnostic start system test all`	Executes all online diagnostic tests.
`Router# diagnostic stop system test all`	Stops the execution of all online diagnostic tests.

This example shows how to start all online diagnostic tests:

```
Router# diagnostic start system test all
*************************************************************************
* WARNING:                                                              *
*   'diagnostic start system test all' will disrupt normal system       *
*   operation. The system requires RESET after the command              *
*   'diagnostic start system test all' has completed prior to           *
*   normal use.                                                         *
*                                                                       *
* IMPORTANT:                                                            *
*   1. DO NOT INSERT, OIR, or POWER DOWN Linecards or                   *
*      Supervisor while system test is running.                         *
*                                                                       *
*   2. DO NOT ISSUE ANY DIAGNOSTIC COMMAND except                       *
*      "diagnostic stop system test all" while system test              *
*      is running.                                                      *
*                                                                       *
*   3. PLEASE MAKE SURE no traffic is running in background.            *
*************************************************************************
Do you want to continue? [no]:
```

Displaying Online Diagnostic Tests and Test Results

You can display the online diagnostic tests that are configured and check the results of the tests using the following **show** commands:

- **show diagnostic content**
- **show diagnostic health**

To display the diagnostic tests that are configured, perform this task:

Command	Purpose
show diagnostic {**bootup level** | **content** [**module** *num*] | **events** [**module** *num*] [**event-type** *event-type*] | **health** | **ondemand settings** | **result** [**module** *num*] [**detail**] | **schedule** [**module** *num*]}	Displays the test results of online diagnostics and lists supported test suites.

This example shows how to display the online diagnostics that are configured on module 1:

```
Router# show diagnostic content module 1

Module 1: Supervisor Engine 32 8GE (Active)

  Diagnostics test suite attributes:
    M/C/* - Minimal bootup level test / Complete bootup level test / NA
      B/* - Basic ondemand test / NA
    P/V/* - Per port test / Per device test / NA
    D/N/* - Disruptive test / Non-disruptive test / NA
      S/* - Only applicable to standby unit / NA
      X/* - Not a health monitoring test / NA
      F/* - Fixed monitoring interval test / NA
      E/* - Always enabled monitoring test / NA
      A/I - Monitoring is active / Monitoring is inactive
      R/* - Power-down line cards and need reload supervisor / NA
      K/* - Require resetting the line card after the test has completed / NA
      T/* - Shut down all ports and need reload supervisor / NA

                                                          Test Interval   Thre-
  ID   Test Name                          Attributes      day hh:mm:ss.ms shold
  ==== ================================== ============    =============== =====
    1) TestScratchRegister -------------> ***N****A***    000 00:00:30.00 5
    2) TestSPRPInbandPing --------------> ***N****A***    000 00:00:15.00 10
    3) TestTransceiverIntegrity --------> **PD****I***    not configured  n/a
    4) TestActiveToStandbyLoopback -----> M*PDSX**I***    not configured  n/a
    5) TestLoopback --------------------> M*PD*X**I***    not configured  n/a
    6) TestTxPathMonitoring ------------> M**N****A***    000 00:00:02.00 10
    7) TestNewIndexLearn ---------------> M**N****I***    000 00:00:15.00 10
    8) TestDontConditionalLearn --------> M**N****I***    000 00:00:15.00 10
    9) TestBadBpduTrap -----------------> M**D*X**I***    not configured  n/a
   10) TestMatchCapture ----------------> M**D*X**I***    not configured  n/a
   11) TestProtocolMatchChannel --------> M**D*X**I***    not configured  n/a
   12) TestFibDevices ------------------> M**N****I***    000 00:00:15.00 10
   13) TestIPv4FibShortcut -------------> M**N****I***    000 00:00:15.00 10
   14) TestL3Capture2 ------------------> M**N****I***    000 00:00:15.00 10
   15) TestIPv6FibShortcut -------------> M**N****I***    000 00:00:15.00 10
   16) TestMPLSFibShortcut -------------> M**N****I***    000 00:00:15.00 10
   17) TestNATFibShortcut --------------> M**N****I***    000 00:00:15.00 10
   18) TestAclPermit -------------------> M**N****I***    000 00:00:15.00 10
   19) TestAclDeny ---------------------> M**D*X**I***    not configured  n/a
   20) TestQoSTcam ---------------------> M**D*X**I***    not configured  n/a
   21) TestL3VlanMet -------------------> M**N****I***    000 00:00:15.00 10
   22) TestIngressSpan -----------------> M**N****I***    000 00:00:15.00 10
   23) TestEgressSpan ------------------> M**D*X**I***    not configured  n/a
   24) TestNetflowInlineRewrite --------> C*PD*X**I***    not configured  n/a
   25) TestTrafficStress ---------------> ***D*X**I**T    not configured  n/a
   26) TestFibTcamSSRAM ----------------> ***D*X**IR**    not configured  n/a
   27) TestAsicMemory ------------------> ***D*X**IR**    not configured  n/a
   28) TestAclQosTcam ------------------> ***D*X**IR**    not configured  n/a
   29) TestNetflowTcam -----------------> ***D*X**IR**    not configured  n/a
   30) ScheduleSwitchover --------------> ***D*X**I***    not configured  n/a
```

```
  31) TestFirmwareDiagStatus ----------> M**N****I***    000 00:00:15.00 10
  32) TestAsicSync --------------------> ***N****A***    000 00:00:15.00 10
  33) TestUnusedPortLoopback ----------> **PN****A***    000 00:01:00.00 10
  34) TestErrorCounterMonitor ---------> ***N****A***    000 00:00:30.00 10
  35) TestPortTxMonitoring ------------> **PN****A***    000 00:01:15.00 5
  36) TestL3HealthMonitoring ----------> ***N**FEA***    000 00:00:05.00 10
  37) TestCFRW ------------------------> M*VN*X**I***    not configured  n/a

Router#
```

This example shows how to display the online diagnostic results for module 1:

```
Router# show diagnostic result module 1
Current bootup diagnostic level: bypass

Module 1: Cisco ME 6524 Ethernet Switch  SerialNo : CAT103956WS

  Overall Diagnostic Result for Module 1 : MINOR ERROR
  Diagnostic level at card bootup: bypass

  Test results: (. = Pass, F = Fail, U = Untested)

    1) TestSPRPInbandPing --------------> .
    2) TestTransceiverIntegrity:

   Port  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
   ----------------------------------------------------------------------------
         U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U

   Port 25 26 27 28 29 30 31 32
   ----------------------------
         .  .  U  U  U  U  U  .


    3) TestScratchRegister -------------> .
    4) TestNonDisruptiveLoopback:

   Port  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
   ----------------------------------------------------------------------------
         .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

   Port 25 26 27 28 29 30 31 32
   ----------------------------
         U  U  U  U  U  U  U  U


    5) TestLoopback:

   Port  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
   ----------------------------------------------------------------------------
         .  F  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

   Port 25 26 27 28 29 30 31 32
   ----------------------------
         .  .  .  .  .  .  .  .


    6) TestNewIndexLearn ---------------> .
    7) TestDontConditionalLearn --------> .
    8) TestBadBpduTrap -----------------> .
    9) TestMatchCapture ----------------> .
   10) TestProtocolMatchChannel --------> .
   11) TestFibDevices ------------------> .
   12) TestIPv4FibShortcut -------------> .
```

```
   13) TestL3Capture2 ------------------> .
   14) TestIPv6FibShortcut -------------> .
   15) TestMPLSFibShortcut -------------> .
   16) TestNATFibShortcut --------------> .
   17) TestAclPermit -------------------> .
   18) TestAclDeny ---------------------> .
   19) TestQoSTcam ---------------------> .
   20) TestL3VlanMet -------------------> .
   21) TestIngressSpan -----------------> .
   22) TestEgressSpan ------------------> .
   23) TestNetflowInlineRewrite:

   Port  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
   ----------------------------------------------------------------------------
         U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U  U

   Port 25 26 27 28 29 30 31 32
   ----------------------------
         U  U  U  U  U  U  U  U


   24) TestTrafficStress ---------------> U
   25) TestFibTcamSSRAM ----------------> U
   26) TestAsicMemory ------------------> U
   27) TestLtlFpoeMemoryConsistency ----> .
   28) TestAclQosTcam ------------------> U
   29) TestNetflowTcam -----------------> U
   30) FirmwareDiagStatus --------------> .
   31) TestL3HealthMonitoring ----------> .
   32) TestCFRW:

      Device  1
      ---------
              .


   33) TestRwEngineOverSubscription ----> U


Router#
```

This example shows how to display the detailed online diagnostic results for module 1:

```
Router# show diagnostic result module 1 detail
Current bootup diagnostic level:minimal

Module 1:

  Overall Diagnostic Result for Module 1 :PASS
  Diagnostic level at card bootup:minimal

  Test results:(. = Pass, F = Fail, U = Untested)

  ___________________________________________________________________________

    1) TestScratchRegister -------------> .

          Error code ------------------> 0 (DIAG_SUCCESS)
          Total run count -------------> 330
          Last test execution time ----> May 12 2003 14:49:36
          First test failure time -----> n/a
          Last test failure time ------> n/a
          Last test pass time ---------> May 12 2003 14:49:36
          Total failure count ---------> 0
```

```
        Consecutive failure count ---> 0
  ___________________________________________________________________________

  2) TestSPRPInbandPing --------------> .

        Error code ------------------> 0 (DIAG_SUCCESS)
        Total run count -------------> 660
        Last test execution time ----> May 12 2003 14:49:38
        First test failure time -----> n/a
        Last test failure time ------> n/a
        Last test pass time ---------> May 12 2003 14:49:38
        Total failure count ---------> 0
        Consecutive failure count ---> 0
  ___________________________________________________________________________

  3) TestGBICIntegrity:

    Port  1  2
    ----------
          U  U


        Error code ------------------> 0 (DIAG_SUCCESS)
        Total run count -------------> 0
        Last test execution time ----> n/a
        First test failure time -----> n/a
        Last test failure time ------> n/a
        Last test pass time ---------> n/a
        Total failure count ---------> 0
        Consecutive failure count ---> 0
  ________________________________________________________________________
.
.
.

Router#
```

This example shows how to display the output for the health checks performed:

```
Router# show diagnostic health
CPU utilization for the past 5 mins is greater than 70%
five minutes: 81%
EARL reset history:
Module 5: WS-SUP720-BASE EARL patch log -
Num. of times patch applied : 0
Num. of times patch requested : 0
Non-zero port counters for 1/8 -
13. linkChange = 159810
Non-zero port counters for 1/9 -
0. rxCRCAlignErrors = 5
3. rxFragmentPkts = 9
6. ifInErrors = 46
13. linkChange = 1
Current bootup diagnostic level: minimal
Test results: (. = Pass, F = Fail, U = Untested)
36) TestErrorCounterMonitor ---------> F
Error code ------------------> 1 (DIAG_FAILURE)
Total run count -------------> 29
Last test execution time ----> Mar 16 2007 19:04:02
First test failure time -----> Mar 16 2007 19:03:21
Last test failure time ------> Mar 16 2007 19:04:02
Last test pass time ---------> Mar 16 2007 19:03:19
Total failure count ---------> 4
Consecutive failure count ---> 4
```

```
Error Records as following.
ID -- Asic Identification
IN -- Asic Instance
PO -- Asic Port Number
RE -- Register Identification
RM -- Register Identification More
EG -- Error Group
DV -- Delta Value
CF -- Consecutive Failure
TF -- Total Failure
ID IN PO RE RM DV EG CF TF
---------------------------------------------------------------
26 0 0 338 255 256 2 13 13
26 0 0 344 255 256 2 13 13
26 0 0 358 255 256 2 13 13
System Memory: 524288K total, 353225K used, 171063K free, 1000K kernel reserved
Lowest(b) : 171020288
Process kernel, type POSIX, PID = 1
0K total, 0K text, 0K data, 0K stack, 0K dynamic
Process sbin/chkptd.proc, type POSIX, PID = 16386
2296K total, 1988K text, 120K data, 12K stack, 176K dynamic
65536 heapsize, 55356 allocated, 8084 free
Router#
```

Performing Memory Tests

Most online diagnostic tests do not need any special setup or configuration. However, the memory tests, which include the TestFibTcamSSRAM and TestLinecardMemory tests, have some required tasks and some recommended tasks that you should complete before running them.

Before you run any of the online diagnostic memory tests, perform the following tasks:

- Required tasks
  - Isolate network traffic by disabling all connected ports.
  - Do not send test packets during a memory test.
  - Reset the system before returning the system to normal operating mode.
- Turn off all background health-monitoring tests using the **no diagnostic monitor module** *number* **test all** command.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Onboard Failure Logging

This chapter describes how to configure Onboard Failure Logging (OBFL) in Cisco IOS Release 12.2SX. Release 12.2(33)SXH and later releases support OBFL.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding OBFL

These sections describe OBFL:

Overview of OBFL

The Onboard Failure Logging (OBFL) feature collects data such as operating temperatures, hardware uptime, interrupts, and other important events and messages from system hardware installed in a Cisco router or switch. The data is stored in nonvolatile memory and helps technical personnel diagnose hardware problems.

Data Collected by OBFL

The OBFL feature records operating temperatures, hardware uptime, interrupts, and other important events and messages that can assist with diagnosing problems with hardware cards (or *modules*) installed in a Cisco router or switch. Data is logged to files stored in nonvolatile memory. When the onboard hardware is started up, a first record is made for each area monitored and becomes a base value for subsequent records. The OBFL feature provides a circular updating scheme for collecting continuous records and archiving older (historical) records, ensuring accurate data about the system. Data is recorded in one of two formats: continuous information that displays a snapshot of measurements and samples in a continuous file, and summary information that provides details about the data being collected. The data is displayed using the **show logging onboard** command. The message “No historical data to display” is seen when historical data is not available.

The following sections describe the type of data collected:

- Temperature, page 13-2
- Operational Uptime, page 13-4
- Interrupts, page 13-7
- Message Logging, page 13-8

Temperature

Temperatures surrounding hardware modules can exceed recommended safe operating ranges and cause system problems such as packet drops. Higher than recommended operating temperatures can also accelerate component degradation and affect device reliability. Monitoring temperatures is important for maintaining environmental control and system reliability. Once a temperature sample is logged, the sample becomes the base value for the next record. From that point on, temperatures are recorded either when there are changes from the previous record or if the maximum storage time is exceeded. Temperatures are measured and recorded in degrees Celsius.

Temperature Example

```
--------------------------------------------------------------------------------
TEMPERATURE SUMMARY INFORMATION
--------------------------------------------------------------------------------
Number of sensors          : 12
Sampling frequency         : 5 minutes
Maximum time of storage    : 120 minutes
--------------------------------------------------------------------------------
Sensor                                 |   ID   | Maximum Temperature 0C
--------------------------------------------------------------------------------
MB-Out                                  980201     43
MB-In                                   980202     28
MB                                      980203     29
MB                                      980204     38
EARL-Out                                910201     0
EARL-In                                 910202     0
```

```
SSA 1                              980301     38
SSA 2                              980302     36
JANUS 1                            980303     36
JANUS 2                            980304     35
GEMINI 1                           980305     0
GEMINI 2                           980306     0
---------------------------------------------------------------
Temp                          Sensor ID
0C     1    2    3    4    5    6    7    8    9   10   11   12
---------------------------------------------------------------
No historical data to display
---------------------------------------------------------------
--------------------------------------------------------------------------------
TEMPERATURE CONTINUOUS INFORMATION
--------------------------------------------------------------------------------
Sensor                            |   ID  |
--------------------------------------------------------------------------------
MB-Out                             980201
MB-In                              980202
MB                                 980203
MB                                 980204
EARL-Out                           910201
EARL-In                            910202
SSA 1                              980301
SSA 2                              980302
JANUS 1                            980303
JANUS 2                            980304
GEMINI 1                           980305
GEMINI 2                           980306

-------------------------------------------------------------------------------
       Time Stamp    |Sensor Temperature 0C
MM/DD/YYYY HH:MM:SS |  1    2    3    4    5    6    7    8    9   10   11   12
-------------------------------------------------------------------------------
03/06/2007 22:32:51   31   26   27   27   NA   NA   33   32   30   29   NA   NA
03/06/2007 22:37:51   43   28   29   38   NA   NA   38   36   36   35   NA   NA
-------------------------------------------------------------------------------
```

To interpret this data:

- Number of sensors is the total number of temperature sensors that will be recorded. A column for each sensor is displayed with temperatures listed under the number of each sensor, as available.
- Sampling frequency is the time between measurements.
- Maximum time of storage determines the maximum amount of time, in minutes, that can pass when the temperature remains unchanged and the data is not saved to storage media. After this time, a temperature record will be saved even if the temperature has not changed.
- The Sensor column lists the name of the sensor.
- The ID column lists an assigned identifier for the sensor.
- Maximum Temperature 0C shows the highest recorded temperature per sensor.
- Temp indicates a recorded temperature in degrees Celsius in the historical record. Columns following show the total time each sensor has recorded that temperature.
- Sensor ID is an assigned number, so that temperatures for the same sensor can be stored together.

Operational Uptime

The operational uptime tracking begins when the module is powered on, and information is retained for the life of the module.

Operational Uptime Example

```
--------------------------------------------------------------------------------
UPTIME SUMMARY INFORMATION
--------------------------------------------------------------------------------
First customer power on : 03/06/2007 22:32:51
Total uptime            :   0 years   0 weeks   2 days  18 hours  10 minutes
Total downtime          :   0 years   0 weeks   0 days   8 hours   7 minutes
Number of resets        : 130
Number of slot changes  : 16
Current reset reason    : 0xA1
Current reset timestamp : 03/07/2007 13:29:07
Current slot            : 2
Current uptime          :   0 years   0 weeks   1 days   7 hours   0 minutes
--------------------------------------------------------------------------------
Reset  |        |
Reason | Count  |
--------------------------------------------------------------------------------
0x5         64
0x6         62
0xA1         4
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
UPTIME CONTINUOUS INFORMATION
--------------------------------------------------------------------------------
Time Stamp          | Reset  | Uptime
MM/DD/YYYY HH:MM:SS | Reason | years weeks days hours  minutes
--------------------------------------------------------------------------------
03/06/2007 22:32:51   0xA1      0     0     0     0     0
--------------------------------------------------------------------------------
```

The operational uptime application tracks the following events:

- Date and time the customer first powered on a component.
- Total uptime and downtime for the component in years, weeks, days, hours, and minutes.
- Total number of component resets.
- Total number of slot (module) changes.
- Current reset timestamp to include the date and time.
- Current slot (module) number of the component.
- Current uptime in years, weeks, days, hours, and minutes.
- Reset reason; see Table 13-1 to translate the numbers displayed.
- Count is the number of resets that have occurred for each reset reason.

Table 13-1 Reset Reason Codes and Explanations

Reset Reason Code (in hex)	Component/Explanation
0x01	Chassis on
0x02	Line card hot plug in

Table 13-1 Reset Reason Codes and Explanations

Reset Reason Code (in hex)	Component/Explanation
0x03	Supervisor requests line card off or on
0x04	Supervisor requests hard reset on line card
0x05	Line card requests Supervisor off or on
0x06	Line card requests hard reset on Supervisor
0x07	Line card self reset using the internal system register
0x08	—
0x09	—
0x0A	Momentary power interruption on the line card
0x0B	—
0x0C	—
0x0D	—
0x0E	—
0x0F	—
0x10	—
0x11	Off or on after Supervisor non-maskable interrupts (NMI)
0x12	Hard reset after Supervisor NMI
0x13	Soft reset after Supervisor NMI
0x14	—
0x15	Off or on after line card asks Supervisor NMI
0x16	Hard reset after line card asks Supervisor NMI
0x17	Soft reset after line card asks Supervisor NMI
0x18	—
0x19	Off or on after line card self NMI
0x1A	Hard reset after line card self NMI
0x1B	Soft reset after line card self NMI
0x21	Off or on after spurious NMI
0x22	Hard reset after spurious NMI
0x23	Soft reset after spurious NMI
0x24	—
0x25	Off or on after watchdog NMI
0x26	Hard reset after watchdog NMI
0x27	Soft reset after watchdog NMI
0x28	—
0x29	Off or on after parity NMI
0x2A	Hard reset after parity NMI

Table 13-1 Reset Reason Codes and Explanations

Reset Reason Code (in hex)	Component/Explanation
0x2B	Soft reset after parity NMI
0x31	Off or on after system fatal interrupt
0x32	Hard reset after system fatal interrupt
0x33	Soft reset after system fatal interrupt
0x34	—
0x35	Off or on after application-specific integrated circuit (ASIC) interrupt
0x36	Hard reset after ASIC interrupt
0x37	Soft reset after ASIC interrupt
0x38	—
0x39	Off or on after unknown interrupt
0x3A	Hard reset after unknown interrupt
0x3B	Soft reset after unknown interrupt
0x41	Off or on after CPU exception
0x42	Hard reset after CPU exception
0x43	Soft reset after CPU exception
0xA1	Reset data converted to generic data

Interrupts

Interrupts are generated by system components that require attention from the CPU such as ASICs and NMIs. Interrupts are generally related to hardware limit conditions or errors that need to be corrected.

The continuous format records each time a component is interrupted, and this record is stored and used as base information for subsequent records. Each time the list is saved, a timestamp is added. Time differences from the previous interrupt are counted, so that technical personnel can gain a complete record of the component's operational history when an error occurs.

Interrupts Example

```
--------------------------------------------------------------------------------
INTERRUPT SUMMARY INFORMATION
--------------------------------------------------------------------------------
Name                                                  |  ID | Offset | Bit |  Count
--------------------------------------------------------------------------------
No historical data to display
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
CONTINUOUS INTERRUPT INFORMATION
--------------------------------------------------------------------------------
MM/DD/YYYY HH:MM:SS mmm | Name                                 |  ID | Offset | Bit
--------------------------------------------------------------------------------
03/06/2007 22:33:06 450   Port-ASIC #2                             9   0x00E7     6
--------------------------------------------------------------------------------
```

To interpret this data:

- Name is a description of the component including its position in the device.
- ID is an assigned field for data storage.
- Offset is the register offset from a component register's base address.
- Bit is the interrupt bit number recorded from the component's internal register.
- The timestamp shows the date and time that an interrupt occurred down to the millisecond.

Message Logging

The OBFL feature logs standard system messages. Instead of displaying the message to a terminal, the message is written to and stored in a file, so the message can be accessed and read at a later time. System messages range from level 1 alerts to level 7 debug messages, and these levels can be specified in the **hw module logging onboard** command.

Error Message Log Example

```
--------------------------------------------------------------------------------
ERROR MESSAGE SUMMARY INFORMATION
--------------------------------------------------------------------------------
Facility-Sev-Name       | Count | Persistence Flag
MM/DD/YYYY HH:MM:SS
--------------------------------------------------------------------------------
No historical data to display
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
ERROR MESSAGE CONTINUOUS INFORMATION
--------------------------------------------------------------------------------
MM/DD/YYYY HH:MM:SS Facility-Sev-Name
--------------------------------------------------------------------------------
03/06/2007 22:33:35  %GOLD_OBFL-3-GOLD : Diagnostic OBFL: Diagnostic OBFL testing
```

To interpret this data:

- A timestamp shows the date and time the message was logged.
- Facility-Sev-Name is a coded naming scheme for a system message, as follows:
  - The Facility code consists of two or more uppercase letters that indicate the hardware device (facility) to which the message refers.
  - Sev is a single-digit code from 1 to 7 that reflects the severity of the message.
  - Name is one or two code names separated by a hyphen that describe the part of the system from where the message is coming.
- The error message follows the Facility-Sev-Name codes. For more information about system messages, see the *Cisco IOS System and Error Messages* guide.
- Count indicates the number of instances of this message that is allowed in the history file. Once that number of instances has been recorded, the oldest instance will be removed from the history file to make room for new ones.
- The Persistence Flag gives a message priority over others that do not have the flag set.

Restrictions for OBFL

- Software Restrictions—If a device (router or switch) intends to use *linear* flash memory as its OBFL storage media, Cisco IOS software must reserve a minimum of two physical sectors (or physical blocks) for the OBFL feature. Because an erase operation for a linear flash device is done on per-sector (or per-block) basis, one extra physical sector is needed. Otherwise, the minimum amount of space reserved for the OBFL feature on any device must be at least 8 KB.
- Firmware Restrictions—If a line card or port adapter runs an operating system or firmware that is different from the Cisco IOS operating system, the line card or port adapter must provide device driver level support or an interprocess communications (IPC) layer that allows the OBFL file system to communicate to the line card or port adapter. This requirement is enforced to allow OBFL data to be recorded on a storage device attached to the line card or port adapter.
- Hardware Restrictions—To support the OBFL feature, a device must have at least 8 KB of nonvolatile memory space reserved for OBFL data logging.

Enabling OBFL

Note The OBFL feature is enabled by default. Because of the valuable information this feature offers technical personnel, it should not be disabled. If you find the feature has been disabled, use the following steps to reenable it.

To enable OBFL, perform this task:

	Command or Action	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`hw-module switch`** *`switch-number`* **`module`** *`module-number`* **`logging onboard`** [**`message level`** {**`1`**-**`7`**}]	Enables OBFL on the specified hardware module. **Note** By default, all system messages sent to a device are logged by the OBFL feature. You can define a specific message level (only level 1 messages, as an example) to be logged using the **message level** keywords.
Step 4	`Router(config)#` **`end`**	Ends global configuration mode.

Configuration Examples for OBFL

The important OBFL feature is the information that is displayed by the **show logging onboard module** privileged EXEC command. This section provides the following examples of how to enable and display OBFL records.

- Enabling OBFL Message Logging: Example
- OBFL Message Log: Example
- OBFL Component Uptime Report: Example
- OBFL Report for a Specific Time: Example

Enabling OBFL Message Logging: Example

The following example shows how to configure OBFL message logging at level 3:

```
Router(config)# hw-module switch 2 module 1 logging onboard message level 3
```

OBFL Message Log: Example

The following example shows how to display the system messages that are being logged for module 2:

```
Router# show logging onboard module 2 message continuous

--------------------------------------------------------------------------------
ERROR MESSAGE CONTINUOUS INFORMATION
--------------------------------------------------------------------------------
MM/DD/YYYY HH:MM:SS Facility-Sev-Name
--------------------------------------------------------------------------------
03/06/2007 22:33:35 %SWITCH_IF-3-CAMERR : [chars], for VCI [dec] VPI [dec] in stdby data
path check, status: [dec]
--------------------------------------------------------------------------------
```

OBFL Component Uptime Report: Example

The following example shows how to display a summary report for component uptimes for module 2:

```
Router# show logging onboard module 2 uptime

--------------------------------------------------------------------------------
UPTIME SUMMARY INFORMATION
--------------------------------------------------------------------------------
First customer power on : 03/06/2007 22:32:51
Total uptime            :   0 years    0 weeks    0 days    0 hours   35 minutes
Total downtime          :   0 years    0 weeks    0 days    0 hours    0 minutes
Number of resets        : 1
Number of slot changes  : 0
Current reset reason    : 0xA1
Current reset timestamp : 03/06/2007 22:31:34
Current slot            : 2
Current uptime          :   0 years    0 weeks    0 days    0 hours   35 minutes
--------------------------------------------------------------------------------
Reset  |         |
Reason | Count   |
--------------------------------------------------------------------------------
No historical data to display
--------------------------------------------------------------------------------
```

OBFL Report for a Specific Time: Example

The following example shows how to display continuous reports for all components during a specific time period:

```
Router# show logging onboard module 3 continuous start 15:01:57 1 Mar 2007 end 15:04:57 3
Mar 2007

PID: WS-X6748-GE-TX    , VID:    , SN: SAL09063B85

--------------------------------------------------------------------------------
UPTIME CONTINUOUS INFORMATION
--------------------------------------------------------------------------------
Time Stamp          | Reset  | Uptime
MM/DD/YYYY HH:MM:SS | Reason | years weeks days hours  minutes
--------------------------------------------------------------------------------
03/01/2007 15:01:57   0xA1      0     0     0    10     0
03/03/2007 02:29:29   0xA1      0     0     0     5     0
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
TEMPERATURE CONTINUOUS INFORMATION
--------------------------------------------------------------------------------
Sensor                          |   ID  |
--------------------------------------------------------------------------------
MB-Out                            930201
MB-In                             930202
MB                                930203
MB                                930204
EARL-Out                          910201
EARL-In                           910202
SSA 1                             930301
SSA 2                             930302
JANUS 1                           930303
JANUS 2                           930304
GEMINI 1                          930305
GEMINI 2                          930306
-------------------------------------------------------------------------------
       Time Stamp    |Sensor Temperature 0C
MM/DD/YYYY HH:MM:SS |  1    2    3    4    5    6    7    8    9   10   11   12
-------------------------------------------------------------------------------
03/01/2007 15:01:57   26   26   NA   NA   NA   NA    0    0    0    0    0    0
03/01/2007 15:06:57   39   27   NA   NA   NA   NA   39   37   36   29   32   32
03/01/2007 15:11:02   40   27   NA   NA   NA   NA   40   38   37   30   32   32
03/01/2007 17:06:06   40   27   NA   NA   NA   NA   40   38   37   30   32   32
03/01/2007 19:01:09   40   27   NA   NA   NA   NA   40   38   37   30   32   32
03/03/2007 02:29:30   25   26   NA   NA   NA   NA    0    0    0    0    0    0
03/03/2007 02:34:30   38   26   NA   NA   NA   NA   39   37   36   29   31   31
03/03/2007 04:29:33   40   27   NA   NA   NA   NA   40   38   36   30   32   32
03/03/2007 06:24:37   40   27   NA   NA   NA   NA   40   38   36   29   32   32
03/03/2007 08:19:40   40   27   NA   NA   NA   NA   40   38   36   29   32   32
03/03/2007 10:14:44   40   27   NA   NA   NA   NA   40   38   36   30   32   32
03/03/2007 12:09:47   40   27   NA   NA   NA   NA   40   38   36   30   32   32
03/03/2007 14:04:51   40   27   NA   NA   NA   NA   40   38   36   30   32   32
-------------------------------------------------------------------------------
--------------------------------------------------------------------------------
CONTINUOUS INTERRUPT INFORMATION
--------------------------------------------------------------------------------
MM/DD/YYYY HH:MM:SS mmm | Name                          | ID | Offset | Bit
--------------------------------------------------------------------------------
03/01/2007 15:01:59 350   Port-ASIC #0                     7   0x00E7     6
03/03/2007 02:29:34 650   Port-ASIC #0                     7   0x00E7     6
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
ERROR MESSAGE CONTINUOUS INFORMATION
--------------------------------------------------------------------------------
MM/DD/YYYY HH:MM:SS Facility-Sev-Name
--------------------------------------------------------------------------------
03/01/2007 15:02:15  %GOLD_OBFL-3-GOLD : Diagnostic OBFL: Diagnostic OBFL testing
03/03/2007 02:29:51  %GOLD_OBFL-3-GOLD : Diagnostic OBFL: Diagnostic OBFL testing
--------------------------------------------------------------------------------
```

Configuring and Monitoring the Switch Fabric Functionality

This chapter describes how to configure the switching mode and monitor the switch fabric functionality that is included on the Supervisor Engine 720-10GE and the Supervisor Engine 720:

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Understanding the Switch Fabric Functionality

These sections describe how the switch fabric functionality works:

Switch Fabric Functionality Overview

The switch fabric functionality is built into the supervisor engine and creates a dedicated connection between fabric-enabled modules and provides uninterrupted transmission of frames between these modules. In addition to the direct connection between fabric-enabled modules provided by the switch fabric funtionality, fabric-enabled modules also have a direct connection to the 32-Gbps forwarding bus.

Forwarding Decisions for Layer 3-Switched Traffic

Either a PFC3 or a Distributed Feature Card 3 (DFC3) makes the forwarding decision for Layer 3-switched traffic as follows:

- A PFC3 makes all forwarding decisions for each packet that enters the switch through a module without a DFC3.
- A DFC3 makes all forwarding decisions for each packet that enters the switch on a DFC3-enabled module in these situations:
  - If the egress port is on the same module as the ingress port, the DFC3 forwards the packet locally (the packet never leaves the module).
  - If the egress port is on a different fabric-enabled module, the DFC3 sends the packet to the egress module, which sends it out the egress port.
  - If the egress port is on a different nonfabric-enabled module, the DFC3 sends the packet to the supervisor engine. The supervisor engine fabric interface transfers the packet to the 32-Gbps switching bus where it is received by the egress module and is sent out the egress port.

Switching Modes

With a Supervisor Engine 720-10GE or a Supervisor Engine 720, traffic is forwarded to and from modules in one of the following modes:

- Compact mode—The switch uses this mode for all traffic when only fabric-enabled modules are installed. In this mode, a compact version of the DBus header is forwarded over the switch fabric channel, which provides the best possible performance.
- Truncated mode—The switch uses this mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.
- Bus mode (also called flow-through mode)—The switch uses this mode for traffic between nonfabric-enabled modules and for traffic between a nonfabric-enabled module and a fabric-enabled module. In this mode, all traffic passes between the local bus and the supervisor engine bus.

Table 14-1 shows the switching modes used with fabric-enabled and nonfabric-enabled modules installed.

Table 14-1 Switch Fabric Functionality Switching Modes

Modules	Switching Modes
Between fabric-enabled modules (when no nonfabric-enabled modules are installed)	Compact[1]
Between fabric-enabled modules (when nonfabric-enabled modules are also installed)	Truncated[2]
Between fabric-enabled and nonfabric-enabled modules	Bus
Between non-fabric-enabled modules	Bus

1. In **show** commands, displayed as dcef mode for fabric-enabled modules with DFC3 installed; displayed as fabric mode for other fabric-enabled modules.
2. Displayed as fabric mode in **show** commands.

Configuring the Switch Fabric Functionality

To configure the switching mode, perform this task:

Command	Purpose
`Router(config)# [no] fabric switching-mode allow {bus-mode	{truncated [{threshold [number]}]}`	Configures the switching mode.

When configuring the switching mode, note the following information:

- To allow use of nonfabric-enabled modules or to allow fabric-enabled modules to use bus mode, enter the **fabric switching-mode allow bus-mode** command.
- To prevent use of nonfabric-enabled modules or to prevent fabric-enabled modules from using bus mode, enter the **no fabric switching-mode allow bus-mode** command.

Caution When you enter the **no fabric switching-mode allow bus-mode** command, power is removed from any nonfabric-enabled modules installed in the switch.

- To allow fabric-enabled modules to use truncated mode, enter the **fabric switching-mode allow truncated** command.
- To prevent fabric-enabled modules from using truncated mode, enter the **no fabric switching-mode allow truncated** command.
- To configure how many fabric-enabled modules must be installed before they use truncated mode instead of bus mode, enter the **fabric switching-mode allow truncated threshold** *number* command.
- To return to the default truncated-mode threshold, enter the **no fabric switching-mode allow truncated threshold** command.

Monitoring the Switch Fabric Functionality

The switch fabric functionality supports a number of **show** commands for monitoring purposes. A fully automated startup sequence brings the module online and runs the connectivity diagnostics on the ports.

These sections describe how to monitor the switch fabric functionality:

Displaying the Switch Fabric Redundancy Status

To display the switch fabric redundancy status, perform this task:

Command	Purpose
Router# **show fabric active**	Displays switch fabric redundancy status.

```
Router# show fabric active
Active fabric card in slot 5
No backup fabric card in the system
Router#
```

Displaying Fabric Channel Switching Modes

To display the fabric channel switching mode of one or all modules, perform this task:

Command	Purpose
Router# **show fabric switching-mode** [**module** {*slot_number* | **all**]	Displays fabric channel switching mode of one or all modules.

This example shows how to display the fabric channel switching mode of all modules:

```
Router# show fabric switching-mode module all
Module Slot     Switching Mode
    3                     Bus
    5                     Bus
Router#
```

Displaying the Fabric Status

To display the fabric status of one or all switching modules, perform this task:

Command	Purpose
Router# **show fabric status** [*slot_number* | **all**]	Displays fabric status.

This example shows how to display the fabric status of all modules:

```
Router# show fabric status
slot   channel speed module   fabric   hotStandby  Standby  Standby
                    status   status      support  module   fabric
    1        0   20G     OK       OK          N/A
    1        1   20G     OK       OK          N/A
    2        0   20G     OK       OK   Y(not-hot)
    2        1   20G     OK       OK   Y(not-hot)
    3        0   20G     OK       OK   Y(not-hot)
    4        0   20G     OK       OK   Y(not-hot)
    4        1   20G     OK       OK   Y(not-hot)
Router#
```

Displaying the Fabric Utilization

To display the fabric utilization of one or all modules, perform this task:

Command	Purpose
Router# **show fabric utilization** [*slot_number* | **all**]	Displays fabric utilization.

This example shows how to display the fabric utilization of all modules:

```
Router# show fabric utilization all
slot    channel      speed    Ingress %     Egress %
    1          0        20G            0            0
    1          1        20G            0            0
    2          0        20G            0           24
    2          1        20G            0           24
    3          0        20G           48            0
    4          0        20G            0            0
    4          1        20G            0            0
Router#
```

Displaying Fabric Errors

To display fabric errors of one or all modules, perform this task:

Command	Purpose
Router# **show fabric errors** [*slot_number* | **all**]	Displays fabric errors.

This example shows how to display fabric errors on all modules:

```
Router# show fabric errors

Module errors:
 slot    channel        crc      hbeat        sync   DDR sync
    1          0          0          0           0          0
    8          0          0          0           0          0
    8          1          0          0           0          0
    9          0          0          0           0          0

Fabric errors:
 slot    channel       sync     buffer     timeout
    1          0          0          0           0
    8          0          0          0           0
    8          1          0          0           0
    9          0          0          0           0
Router#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Cisco IP Phone Support

This chapter describes how to configure support for Cisco IP phones on the Catalyst 6500 series switches. This information may be helpful in configuring support for non-Cisco IP phones, but we recommend that you see the manufacturer's documentation for those devices.

Note

- Cisco ME 6500 Series Ethernet switches are not typically used to support Cisco IP phones.
- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Cisco IP Phone Support

These sections describe Cisco IP phone support:

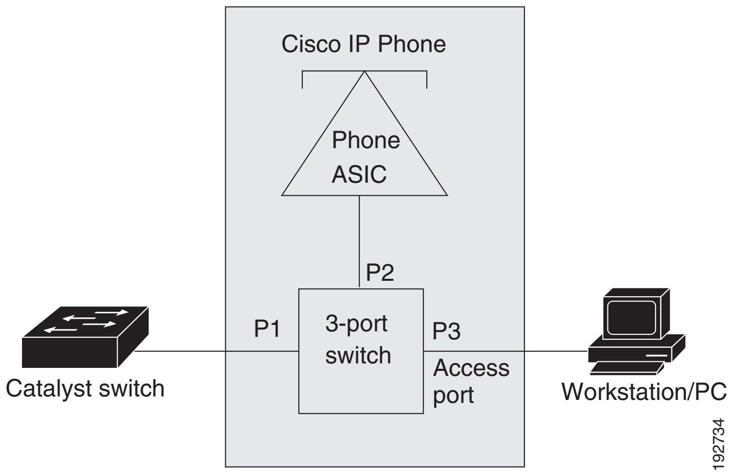
Cisco IP Phone Connections

The Cisco IP phone contains an integrated 3-port 10/100 switch. The ports are dedicated connections to these devices:

- Port 1 connects to the switch.
- Port 2 is an internal 10/100 interface that carries the Cisco IP phone traffic.
- Port 3 connects to a PC or other device.

Figure 15-1 shows a Cisco IP phone connected between a switch and a PC.

Figure 15-1 Cisco IP Phone Connected to a Switch



Cisco IP Phone Voice Traffic

The Cisco IP phone transmits voice traffic with Layer 3 IP precedence and Layer 2 CoS values, which are both set to 5 by default. The sound quality of a Cisco IP phone call can deteriorate if the voice traffic is transmitted unevenly.

You can configure Layer 2 access ports on the switch to send Cisco Discovery Protocol (CDP) packets that configure an attached Cisco IP phone to transmit voice traffic to the switch in any of the following ways:

- In the voice VLAN, tagged with a Layer 2 CoS priority value
- In the access VLAN, tagged with a Layer 2 CoS priority value
- In the access VLAN, untagged (no Layer 2 CoS priority value)

Note In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5 for voice traffic and 3 for voice control traffic).

To provide more predictable voice traffic flow, you can configure QoS on the switch to trust the Layer 3 IP precedence or Layer 2 CoS value in the received traffic (see Chapter 43, "Configuring PFC QoS").

Release 12.2(33)SXI1 and later releases support the trusted boundary device verification feature, which can configure ports on the switch to apply configured QoS port trust commands only when the Cisco Discovery Protocol (CDP) verifies that the device attached to the port is a Cisco IP phone. See the "Configuring Trusted Boundary with Cisco Device Verification" section on page 43-94.

You can configure a Layer 2 access port with an attached Cisco IP phone to use one VLAN for voice traffic and another VLAN for data traffic from a device attached to the Cisco IP phone.

Cisco IP Phone Data Traffic

Note

- The ability to either trust or mark tagged data traffic from the device attached to the access port on the Cisco IP phone is called the "trusted boundary (extended trust for CDP devices)" feature.
- You cannot use Cisco IOS software commands to configure the frame type used by data traffic sent from a device attached to the access port on the Cisco IP phone.
- Untagged data traffic from the device attached to the Cisco IP phone passes through the Cisco IP phone unchanged, regardless of the trust state of the access port on the Cisco IP phone.

To process tagged data traffic (traffic in 802.1Q or 802.1p frame types) from the device attached to the access port on the Cisco IP phone (see Figure 15-1), you can configure Layer 2 access ports on the switch to send CDP packets that instruct an attached Cisco IP phone to configure the access port on the Cisco IP phone to either of these two modes:

- Trusted mode—All traffic received through the access port on the Cisco IP phone passes through the Cisco IP phone unchanged.
- Untrusted mode—All traffic in 802.1Q or 802.1p frames received through the access port on the Cisco IP phone is marked with a configured Layer 2 CoS value. The default Layer 2 CoS value is 0. Untrusted mode is the default.

Most IP phones have no ability to notify the switch of link state changes on the IP phone's access port. When a device attached to the access port is disconnected or disabled administratively, the switch is unaware of the change. Some Cisco IP phones can send a CDP message containing a host presence type length value (TLV) indicating the changed state of the access port link. To recognize the host presence TLV, the switch must be running Cisco IOS Release 12.2(33)SXI or a later release.

Other Cisco IP Phone Features

The Catalyst 6500 series switch provides support for authentication, authorization, and accounting (AAA) for Cisco IP phones, as described in Chapter 60, "Configuring IEEE 802.1X Port-Based Authentication."

The Catalyst 6500 series switch also supports automatic tracking for Cisco Emergency Responder (Cisco ER) to help you manage emergency calls in your telephony network. For further information, see this URL:

http://www.cisco.com/en/US/products/sw/voicesw/ps842/tsd_products_support_series_home.html

Default Cisco IP Phone Support Configuration

Cisco IP phone support is disabled by default.

When the voice VLAN feature is enabled, all untagged traffic is sent with the default CoS priority of the port.

The CoS is not trusted for 802.1P or 802.1Q tagged traffic.

Cisco IP Phone Support Configuration Guidelines and Restrictions

The following guidelines and restrictions apply when configuring Cisco IP phone support:

- You must enable the Cisco Discovery Protocol (CDP) on the port connected to the Cisco IP phone to send configuration information to the Cisco IP phone.
- You can configure a voice VLAN only on a Layer 2 LAN port.
- You can configure the ports on WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules to trust received Layer 2 CoS values (QoS port architecture 1p1q0t/1p3q1t). The WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules cannot supply power to Cisco IP phones.
- You cannot configure 10/100 Mbps ports with QoS port architecture 1p4t/2q2t to trust received Layer 2 CoS values. Configure policies to trust the Layer 3 IP precedence value on switching modules with QoS port architecture 1p4t/2q2t.
- The following conditions indicate that the Cisco IP phone and a device attached to the Cisco IP phone are in the same VLAN and must be in the same IP subnet:
  - If they both use 802.1p or untagged frames
  - If the Cisco IP phone uses 802.1p frames and the device uses untagged frames
  - If the Cisco IP phone uses untagged frames and the device uses 802.1p frames
  - If the Cisco IP phone uses 802.1Q frames and the voice VLAN is the same as the access VLAN
- The Cisco IP phone and a device attached to the Cisco IP phone cannot communicate if they are in the same VLAN and subnet but use different frame types, because traffic between devices in the same subnet is not routed (routing would eliminate the frame type difference).
- You cannot use Cisco IOS software commands to configure the frame type used by traffic sent from a device attached to the access port on the Cisco IP phone.
- If you enable port security on a port configured with a voice VLAN and if there is a PC connected to the Cisco IP phone, set the maximum allowed secure addresses on the port to at least 2.
- You cannot configure static secure MAC addresses in the voice VLAN.
- Ports configured with a voice VLAN can be secure ports (see Chapter 62, "Configuring Port Security").
- In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5 for voice traffic and 3 for voice control traffic).

Configuring Cisco IP Phone Support

These sections describe how to configure Cisco IP phone support:

- Configuring Voice Traffic Support, page 15-5
- Configuring Data Traffic Support, page 15-6

Note Voice VLANs are referred to as *auxiliary VLANs* in the Catalyst software publications.

Configuring Voice Traffic Support

To configure the way in which the Cisco IP phone transmits voice traffic, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface fastethernet`** *`slot/port`*	Selects the port to configure.
Step 2	`Router(config-if)#` **`switchport`**	Configures the LAN port for Layer 2 switching. **Note** You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 port before you can enter additional **switchport** commands with keywords.
Step 3	`Router(config-if)#` **`switchport voice vlan`** `{`*`voice_vlan_ID`* `|` **`dot1p`** `|` **`none`** `|` **`untagged`**`}`	Configures the way in which the Cisco IP phone transmits voice traffic.
Step 4	`Router(config)#` **`end`**	Exits configuration mode.
Step 5	`Router#` **`show interfaces fastethernet`** *`slot/port`* **`switchport`** `Router#` **`show running-config interface fastethernet`** *`slot/port`*	Verifies the configuration.

When configuring the way in which the Cisco IP phone transmits voice traffic, note the following information:

- Enter a voice VLAN ID to send CDP packets that configure the Cisco IP phone to transmit voice traffic in 802.1Q frames, tagged with the voice VLAN ID and a Layer 2 CoS value (the default is 5). Valid VLAN IDs are from 1 to 4094. The switch puts the 802.1Q voice traffic into the voice VLAN.
- Enter the **dot1p** keyword to send CDP packets that configure the Cisco IP phone to transmit voice traffic in 802.1p frames, tagged with VLAN ID 0 and a Layer 2 CoS value (the default is 5 for voice traffic and 3 for voice control traffic). The switch puts the 802.1p voice traffic into the access VLAN.
- Enter the **untagged** keyword to send CDP packets that configure the Cisco IP phone to transmit untagged voice traffic. The switch puts the untagged voice traffic into the access VLAN.
- Enter the **none** keyword to allow the Cisco IP phone to use its own configuration and transmit untagged voice traffic. The switch puts the untagged voice traffic into the access VLAN.
- In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5).
- See Chapter 43, "Configuring PFC QoS," for information about how to configure QoS.

- See the "Configuring a LAN Interface as a Layer 2 Access Port" section on page 17-16 for information about how to configure the port as a Layer 2 access port and configure the access VLAN.

This example shows how to configure Fast Ethernet port 5/1 to send CDP packets that tell the Cisco IP phone to use VLAN 101 as the voice VLAN:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# switchport voice vlan 101
Router(config-if)# exit
```

This example shows how to verify the configuration of Fast Ethernet port 5/1:

```
Router# show interfaces fastethernet 5/1 switchport
Name: Fa5/1
Switchport: Enabled
Administrative Mode: access
Operational Mode: access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: off
Access Mode VLAN: 100
Voice VLAN: 101
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: none
Administrative private-vlan mapping: 900 ((Inactive)) 901 ((Inactive))
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
```

Configuring Data Traffic Support

Note The trusted boundary feature is implemented with the **mls qos trust extend** command.

To configure the way in which an attached Cisco IP phone transmits data traffic, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface fastethernet** *slot/port*	Selects the port to configure.
Step 2	Router(config-if)# **mls qos trust extend** [**cos** *cos_value*]	Configures the way in which an attached Cisco IP phone transmits data traffic.
Step 3	Router(config)# **end**	Exits configuration mode.
Step 4	Router# **show interfaces fastethernet** *slot/port* **switchport** Router# **show running-config interface fastethernet** *slot/port*	Verifies the configuration.

When configuring the way in which an attached Cisco IP phone transmits data traffic, note the following information:

- To send CDP packets that configure an attached Cisco IP phone to trust tagged traffic received from a device connected to the access port on the Cisco IP phone, do not enter the **cos** keyword and CoS value.
- To send CDP packets that configure an attached Cisco IP phone to mark tagged ingress traffic received from a device connected to the access port on the Cisco IP phone, enter the **cos** keyword and CoS value (valid values are 0 through 7).
- You cannot use Cisco IOS software commands to configure whether or not traffic sent from a device attached to the access port on the Cisco IP phone is tagged.

This example shows how to configure Fast Ethernet port 5/1 to send CDP packets that tell the Cisco IP phone to configure its access port as untrusted and to mark all tagged traffic received from a device connected to the access port on the Cisco IP phone with CoS 3:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# mls qos trust extend cos 3
```

This example shows how to configure Fast Ethernet port 5/1 to send CDP packets that tell the Cisco IP phone to configure its access port as trusted:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# mls qos trust extend
```

This example shows how to verify the configuration on Fast Ethernet port 5/1:

```
Router# show queueing interface fastethernet 5/1 | include Extend
  Extend trust state: trusted
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Power over Ethernet Support

This chapter describes how to configure power over Ethernet (PoE) support on the Catalyst 6500 series switches.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Note For information about switching modules that support PoE, see the *Release Notes for Cisco IOS Release 12.2SX* publication at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/release/notes/ol_14271.html

Understanding PoE

These sections describe PoE:

Device Roles

- Power sourcing equipment (PSE)—A device that provides power through a twisted-pair Ethernet connection. The switch, through switching modules equipped with Power over Ethernet (PoE) daughtercards, functions in the PSE role.
- Powered device (PD)—A device powered by a PSE (for example, IP phones, IP cameras, and wireless access points).

Note Not all PoE-capable devices are powered from the switch. There are two sources of local power for PoE-capable devices:

- A power supply connected to the device.
- A power supply through a patch panel over the Ethernet connection to the device.

When a locally powered PoE-capable device is present on a switching module port, the switching module itself cannot detect its presence. If the device supports CDP, the supervisor engine can discover a locally powered PoE-capable device through CDP messaging with the device. If a locally powered PoE-capable device loses local power, the switching module can discover and supply power to the IP phone if the inline power mode is set to **auto**.

PoE Overview

Cisco PoE daughtercards support one or more PoE implementation:

- IEEE 802.3at standard, shown in Cisco Feature Navigator as “PoE Plus (PoE+, PoEP) support”.
  - Supported only with the PoE daughtercard on the WS-X6148E-GE-45AT switching module.
  - With Release 12.2(33)SXJ2 and later releases, these features are supported for IEEE 802.3at-compliant class 4 PDs:
    - Class 4: 30.00 W at the PSE (12.95 W to 25.50 W at the PD).
    - Optionally, LLDP Inline Power Negotiation for PoE+.
  - With releases earlier than Release 12.2(33)SXJ2, maximum 16.8 W at the PSE (ePoE for 45 ports maximum).
- IEEE 802.3af standard.
  - Supported with the WS-F6K-48-AF PoE daughtercard and the PoE daughtercard on the WS-X6148E-GE-45AT switching module.
  - With Release 12.2(33)SXH2 and later releases, maximum 16.80 W at the PSE.
  - With releases earlier than Release 12.2(33)SXH2, maximum 15.40 W at the PSE.
  - The IEEE 802.3af PoE standard defines a method to sense a PD and to immediately classify the power requirement of the PD into these per port power ranges at the PSE:
    - Class 0: Up to 15.4 W (0.44–12.95 W at the PD; default classification)
    - Class 1: Up to 4 W (0.44–3.84 W at the PD)
    - Class 2: Up to 7 W (3.84–6.49 W at the PD)
    - Class 3: Up to 15.4 W (6.49–12.95 W at the PD)
- Cisco prestandard inline power—10 W at the PSE.

With a PoE daughtercard installed, a switching module can automatically detect and provision a PoE-capable device that adheres to a PoE implementation supported by the PoE daughtercard. The switching module can supply power to devices supporting other PoE implementations only through manual configuration.

Only a PD connected directly to the switch port can be powered from the switch. If a second PD is daisy-chained from the PD that is connected to the switch port, the second PD cannot be powered by the switch.

Each PD requires power to be allocated from the chassis power budget. Because each PD can have unique power requirements, more devices can be supported if the system's power management software can intelligently allocate the necessary power on a per-port basis.

You can configure ports to allocate power at a level based on the following:

- If a PD is detected, with auto mode configured:
  - Information sensed from the device
  - A default level
  - A configured maximum level
- Whether or not a PD is present on the port, with static mode configured:
  - A default level
  - A configured level

CPD-Based PoE Management

When a switching module port detects an unpowered PD, the default-allocated power is provided to the port. When the correct amount of power is determined through CDP messaging with the PD, the supervisor engine reduces or increases the allocated power, up to the hardware limit of the installed PoE daughtercard.

Caution When a PD cable is plugged into a port and the power is turned on, the supervisor engine has a 4-second timeout waiting for the link to go up on the line. During those 4 seconds, if the IP phone cable is unplugged and a network device is plugged in, the network device could be damaged. We recommend that you wait at least 10 seconds between unplugging a network device and plugging in another network device.

Inline Power IEEE Power Classification Override

The IEEE 802.3af standard contains no provision for adjustment of the power allocation. 802.3af-compliant PDs that support CDP can use CDP to override the IEEE 802.3af power classification.

With the WS-F6K-48-AF PoE daughtercard or the PoE daughtercard on the WS-X6148E-GE-45AT switching module, Release 12.2(33)SXH and later releases support these inline power IEEE 802.3af power classification override features:

- Power use measurement—The ability to accurately measure the power provided by the port to the powered device.
- Power policing—The ability to monitor power usage on a port.

With power measurement and policing, you can safely override the IEEE 802.3af power classification of a device that requires a power level at the lower end of its IEEE power classification range.

PoE monitoring and policing compares the power consumption on ports with the administrative maximum value (either a configured maximum value or the port's default value). If the power consumption on a monitored port exceeds the administrative maximum value, the following actions occur:

- A syslog message is issued.
- The monitored port is shut down and error-disabled.
- The allocated power is freed.

LLDP Inline Power Negotiation for PoE+ (IEEE 802.3at)

With the PoE daughtercard on the WS-X6148E-GE-45AT switching module, Release 12.2(33)SXJ2 and later releases support IEEE 802.3at-compliant LLDP PoE power negotiation, which supports additional negotiation that can reduce power usage.

Note
- Enabled by default.
- The LLDP TLV used is DTE Power-via-MDI TLV.
- When a PD that performs power negotiation using multiple protocols (CDP and LLDP 802.3at) is connected to a switch, the switch locks to the first protocol packet (CDP or LLDP) that contains the power negotiation TLV. If you need to use any single protocol for power negotiation each time, you must administratively disable the other power negotiation protocols on the switch interface.
- See this publication for other the Link Layer Discovery Protocol (LLDP) configuration procedures:

  http://www.cisco.com/en/US/docs/ios/cether/configuration/guide/ce_lldp-med.html

Configuring PoE Support

Note The switch supports PoE only on Layer 2 switchports.

Displaying PoE Status

This example shows how to display the PoE status on switch:

```
Router# show power auxiliary
system auxiliary power mode = on
system auxiliary power redundancy operationally = redundant
system primary connector power limit =   7266.00 Watts (173.00 Amps @ 42V)
system auxiliary connector power limit = 10500.00 Watts (250.00 Amps @ 42V)
system primary power used =              1407.00 Watts (33.50 Amps @ 42V)
system auxiliary power used =              22.68 Watts ( 0.54 Amps @ 42V)
                        Inline         Inline-Pwr     Inline-Pwr     VDB
                        Pwr-Limit      Used-Thru-Pri  Used-Thru-Aux  Aux-Pwr
Slot Card-Type          Watts   A @42V Watts   A @42V Watts   A @42V Capable
---- ------------------ ------- ------ ------- ------ ------- ------ -------
2    WS-F6K-48-AT       1600.20 38.10    23.10  0.55    11.34  0.27  Yes
4    WS-F6K-48-AT       1600.20 38.10    23.10  0.55    11.34  0.27  Yes
---- ------------------ ------- ------ ------- ------ ------- ------ -------
Totals:                                  46.20  1.10    22.68  0.54
```

Configuring Per-Port PoE Support

To configure per-port PoE support, perform this task:

	Command	Purpose
Step 1	`Router(config-if)#` **power inline** {**auto** | **static** | **never**}[**max** *milliwatts*]	Configures per-port PoE support and optionally specifies a maximum inline power level in milliwatts for the port.
Step 2	`Router#` **show power inline** {*type slot/port* | **module** *slot*}[**detail**]	Verifies the configuration.

When configuring inline power support with the **power inline** command, note the following information:

- To configure auto-detection of a PD and PoE auto-allocation, enter the **auto** keyword.
- To configure auto-detection of a PD but reserve a fixed PoE allocation, enter the **static** keyword.
- To specify the maximum power to allocate to a port, enter either the **auto** or **static** keyword followed by the **max** keyword and the power level in milliwatts.
- When the **auto** keyword is entered and CDP is enabled on the port, a PD that supports CDP can negotiate a different power level.
- To disable auto-detection of a PD, enter the **never** keyword.

- With a WS-F6K-GE48-AF, WS-F6K-48-AF, or the PoE daughtercard on the WS-X6148E-GE-45AT switching module:
  - In Cisco IOS Release 12.2(33)SXH2 and later releases, the configurable range of maximum power using the **max** keyword is 4000 to 16800 milliwatts. For earlier releases, the configurable range for maximum power is 4000 to 15400 milliwatts. For all releases, if no maximum power level is configured, the default maximum power is 15400 milliwatts.

Note To support a large number of inline-powered ports using power levels above 15400 milliwatts on an inline power card, we recommend using the **static** keyword so that the power budget is deterministic.

  - In Cisco IOS Release 12.2(33)SXH2 and later releases, when the **auto** keyword is entered and CDP is enabled on the port, an inline-powered device that supports CDP can negotiate a power level up to 16800 milliwatts unless a lower maximum power level is configured. For earlier releases, the inline-powered device can negotiate a power level up to 15400 milliwatts or the configured maximum power level, if lower.

This example shows how to disable inline power on GigabitEthernet port 2/10:

```
Router# configure terminal
Router(config)# interface gigabitethernet 2/10
Router(config-if)# power inline never
```

This example shows how to enable inline power on GigabitEthernet port 2/10:

```
Router# configure terminal
Router(config)# interface gigabitethernet 2/10
Router(config-if)# power inline auto
```

For Release 12.2(33)SXJ2 and later releases, this example shows how to verify the inline power configuration on GigabitEthernet port 2/10:

```
Router# show power inline gigabitethernet 2/10
Interface Admin  Priority     Oper      Power(Watts)    Device             Class
                 (enabled )            From PS To PD
---------------- ---------- ---------- ------- ------- ------------------- -----

Gi2/10    auto   low        on            14.5    13.1 Cisco IP Phone 9971     4

Interface AdminPowerMax Police ActConsumption
             (Watts)
--------- ------------- ------ --------------

Gi2/10             30.0 on                6.7
```

For releases earlier than Release 12.2(33)SXJ2, this example shows how to verify the inline power configuration on GigabitEthernet port 2/10:

```
Router# show power inline gigabitethernet 2/10
Interface  Admin    Oper     Power        Device
                            (Watts)
---------- ----- ---------- ------- -------------------
Fa5/1      auto  on           6.3   cisco phone device
Router#
```

Configuring PoE Power Priority

With Release 12.2(33)SXJ2 and later releases, you can configure how the switch responds if a power shortage occurs by setting the priority of ports providing PoE. The priority determines the order in which PoE is removed from ports if a power shortage occurs: low-priority, then high-priority, with power maintained for critical-priority ports as long as possible. These sections describe how to configure PoE power priority:

- Setting the PoE Power Priority Global Enable State, page 16-7
- Configuring PoE Port Power Priority, page 16-7

Setting the PoE Power Priority Global Enable State

To disable PoE power priority globally, perform this task:

	Command	Purpose
Step 1	Router(config)# **no power inline priority enable**	Disables PoE power priority globally (default: enabled).
Step 2	Router# **show power inline**	Verifies the configuration.

This example shows how to disable PoE power priority globally:

```
Router(config)# no power inline priority enable
```

The column heading of any **show power inline** command displays the PoE power priority global state ("disabled" in this example):

```
Router# show power inline
Interface Admin  Priority     Oper      Power(Watts)    Device              Class
                 (disabled)            From PS  To PD
---------------- ---------- ---------- ------- ------- ------------------- -----
...
```

Configuring PoE Port Power Priority

To configure PoE port power priority, perform this task:

	Command	Purpose
Step 1	Router(config-if)# **power inline auto priority** {**critical** | **high** | **low**}	Enables PoE port power priority (default: low priority when power priority is enabled globally). If a power shortage occurs, PoE is removed from ports in the following order: • Low priority ports • High priority ports PoE is maintained for critical priority ports as long as possible.
Step 2	Router# **show power inline** *type slot/port* [**detail**]	Verifies the configuration.

This example shows how to configure the PoE port power priority of GigabitEthernet port 2/10 as high:

```
Router# configure terminal
Router(config)# interface gigabitethernet 2/10
Router(config-if)# power inline auto priority high
```

This example shows how to verify the PoE port power priority configuration of GigabitEthernet port 2/10:

```
Router# show power inline gigabitethernet 2/10 detail | include Priority
  Priority: high
```

Configuring PoE Monitoring and Policing

With the WS-F6K-48-AF PoE daughtercard or the PoE daughtercard on the WS-X6148E-GE-45AT switching module, to configure PoE monitoring and policing, perform this task:

	Command	Purpose
Step 1	`Router(config-if)#` **`power inline police`**	Enables PoE monitoring and policing.
Step 2	`Router#` **`show power inline`** `{`*`type slot/port`* `|` **`module`** *`slot`*`}[`**`detail`**`]`	Verifies the configuration.

This example shows how to enable monitoring and policing on GigabitEthernet port 1/9:

```
Router# configure terminal
Router(config)# interface gigabitethernet 2/10
Router(config-if)# power inline police
```

These examples shows how to verify the power monitoring and policing configuration on GigabitEthernet port 2/10:

```
Router# show power inline gigabitethernet 2/10 detail | include Police
Police: on
Router#
Router# show power inline gigabitethernet 2/10
Interface  Admin  Oper  Power (Watts)     Device   Class
                        From PS To Device
--------   -----  ---- ------- ---------  -------  -----
Gi2/10      auto   on    17.3    15.4      Ieee PD   3

Interface  AdminPowerMax (Watts) Police   ActualConsumption
---------  --------------------- ------   -----------------
 Gi2/10           15.4            on      5.7
Router#
```

Disabling LLDP Power Negotiation (IEEE 802.3at)

With the WS-X6148E-GE-45AT switching module and Release 12.2(33)SXJ2 and later releases, LLDP power negotiation is enabled by default. To disable LLDP power negotiation, perform this task:

Command	Purpose
`Router(config-if)#` **`no lldp tlv-select power-management`**	Disables LLDP power negotiation (default: enabled).

This example shows how to display the LLDP power negotiation configuration on interface GigabitEthernet 3/1 when LLDP power negotiation is enabled:

```
Router# show power inline gigabitethernet 2/10 detail | begin LLDP
 LLDP Power Classification    -- Sent to PD --  -- Rcvd from PD --
   Power Type :               type 2 PSE        type 2 PD
   Power Source :             primary           PSE
   Power Priority :           low               high
   Requested Power (watts):   11.2              11.2
   Allocated Power (watts):   11.2              11.2
   Power class :              4                 4

 LLDP Legacy MDI TLV          -- Rcvd from PD --
   MDI power support :        0
   pse power pair :           0
   MDI power class :          0
```

This example shows how to disable LLDP power negotiation on interface GigabitEthernet 2/10:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 2/10
Router(config-if)# no lldp tlv-select power-management
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 2

LAN Switching

Configuring LAN Ports for Layer 2 Switching

This chapter describes how to use the command-line interface (CLI) to configure Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet LAN ports for Layer 2 switching in Cisco IOS Release 12.2SX. The configuration tasks in this chapter apply to LAN ports on switching modules and to the LAN ports on the supervisor engine and Cisco ME 6500 Series Ethernet switches.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- To configure Layer 3 interfaces, see Chapter 30, "Configuring Layer 3 Interfaces."

This chapter consists of these sections:

Understanding Layer 2 Switching

These sections describe how Layer 2 switching works in Cisco IOS Release 12.2SX:

Understanding Layer 2 Ethernet Switching

These sections describe Layer 2 Ethernet switching:

Layer 2 Ethernet Switching Overview

Layer 2 Ethernet ports on Cisco switches support simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

Cisco switches that support Layer 2 Ethernet ports solve congestion problems caused by high-bandwidth devices and by a large number of users by assigning each device (for example, a server) to its own 10-, 100-, or 1000-Mbps collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a properly configured switched environment achieve full access to the bandwidth.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. Normally, Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles.

Switching Frames Between Segments

Each Layer 2 Ethernet port can connect to a single workstation or server, or to a hub through which workstations or servers connect to the network.

On a typical Ethernet hub, all ports connect to a common backplane within the hub, and the bandwidth of the network is shared by all devices attached to the hub. If two stations establish a session that uses a significant level of bandwidth, the network performance of all other stations attached to the hub is degraded.

To reduce degradation, the switch considers each LAN port to be an individual segment. When stations connected to different LAN ports need to communicate, the switch forwards frames from one LAN port to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between LAN ports efficiently, the switch maintains an address table. When a frame enters the switch, it associates the MAC address of the sending network device with the LAN port on which it was received.

Building the MAC Address Table

The MAC address table is built by using the source MAC address of the frames received. When the switch receives a frame for a destination MAC address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the switch adds its relevant source MAC address and port ID to the address table. The switch then forwards subsequent frames to a single LAN port without flooding to all LAN ports.

In PFC3C and PFC3CXL mode, the MAC address table can store at 96,000 address entries (for other PFC3 modes, 64,000 address entries) without flooding any entries. The switch uses an aging mechanism, defined by a configurable aging timer, so if a MAC address remains inactive for a specified number of seconds, it is removed from the address table.

Synchronization and Sharing of the MAC Address Table

With distributed switching, each DFC-equipped switching module learns MAC addresses, maintains a MAC address table, and ages table entries. By enabling MAC address table synchronization over the Ethernet Out of Band Channel (EOBC), you can configure the switch to allow sharing and synchronization of the address tables among all DFCs and supervisor engines in the switch, eliminating the need for flooding by a DFC for an address that is active on another DFC. In VSS mode or when a

WS-6708-10G switching module is present in the system, MAC synchronization is automatically enabled. Otherwise, MAC synchronization must be enabled manually by entering the **mac-address-table synchronize** command.

Notification of Address Table Changes

You can configure the switch to maintain a history of dynamic additions and removals of address table entries associated with a particular LAN port. The change history can be sent as an SNMP trap notification or it can be read manually from the SNMP MIB.

Understanding VLAN Trunks

These sections describe VLAN trunks in Cisco IOS Release 12.2SX:

- Trunking Overview, page 17-3
- Encapsulation Types, page 17-4

Trunking Overview

Note For information about VLANs, see Chapter 23, “Configuring VLANs.”

A trunk is a point-to-point link between the switch and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

Two trunking encapsulations are available on all Ethernet ports:

- Inter-Switch Link (ISL)—ISL is a Cisco-proprietary trunking encapsulation.

  Note The following switching modules do not support ISL encapsulation:

  • WS-X6502-10GE
  • WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6548-GE-45AF
  • WS-X6148-GE-TX, WS-X6148V-GE-TX, WS-X6148-GE-45AF

- 802.1Q—802.1Q is an industry-standard trunking encapsulation.

You can configure a trunk on a single Ethernet port or on an EtherChannel. For more information about EtherChannel, see Chapter 19, “Configuring EtherChannels.”

Ethernet trunk ports support several trunking modes (see Table 17-2 on page 17-4). You can specify whether the trunk uses ISL or 802.1Q encapsulation, and if the encapsulation type is autonegotiated.

Note You can configure LAN ports to negotiate the encapsulation type. You cannot configure WAN interfaces to negotiate the encapsulation type.

The Dynamic Trunking Protocol (DTP) manages trunk autonegotiation on LAN ports. DTP supports autonegotiation of both ISL and 802.1Q trunks.

To autonegotiate trunking, the LAN ports must be in the same VTP domain. Use the **trunk** or **nonegotiate** keywords to force LAN ports in different domains to trunk. For more information on VTP domains, see Chapter 22, “Configuring VTP.”

Encapsulation Types

Table 17-1 lists the Ethernet trunk encapsulation types.

Table 17-1 Ethernet Trunk Encapsulation Types

Encapsulation	Function
switchport trunk encapsulation isl	Specifies ISL encapsulation on the trunk link. **Note** Some modules do not support ISL encapsulation (see the "Trunking Overview" section on page 17-3).
switchport trunk encapsulation dot1q	Specifies 802.1Q encapsulation on the trunk link.
switchport trunk encapsulation negotiate	Specifies that the LAN port negotiate with the neighboring LAN port to become an ISL (preferred) or 802.1Q trunk, depending on the configuration and capabilities of the neighboring LAN port.

The trunking mode, the trunk encapsulation type, and the hardware capabilities of the two connected LAN ports determine whether a link becomes an ISL or 802.1Q trunk.

Layer 2 LAN Port Modes

Table 17-2 lists the Layer 2 LAN port modes and describes how they function on LAN ports.

Table 17-2 Layer 2 LAN Port Modes

Mode	Function
switchport mode access	Puts the LAN port into permanent nontrunking mode and negotiates to convert the link into a nontrunk link. The LAN port becomes a nontrunk port even if the neighboring LAN port does not agree to the change.
switchport mode dynamic desirable	Makes the LAN port actively attempt to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to **trunk**, **desirable**, or **auto** mode. This is the default mode for all LAN ports.
switchport mode dynamic auto	Makes the LAN port willing to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to **trunk** or **desirable** mode.
switchport mode trunk	Puts the LAN port into permanent trunking mode and negotiates to convert the link into a trunk link. The LAN port becomes a trunk port even if the neighboring port does not agree to the change.
switchport nonegotiate	Puts the LAN port into permanent trunking mode but prevents the port from generating DTP frames. You must configure the neighboring port manually as a trunk port to establish a trunk link.

Note DTP is a point-to-point protocol. However, some internetworking devices might forward DTP frames improperly. To avoid this problem, ensure that LAN ports connected to devices that do not support DTP are configured with the **access** keyword if you do not intend to trunk across those links. To enable trunking to a device that does not support DTP, use the **nonegotiate** keyword to cause the LAN port to become a trunk but not generate DTP frames.

Default Layer 2 LAN Interface Configuration

Table 17-3 shows the Layer 2 LAN port default configuration.

Table 17-3 Layer 2 LAN Interface Default Configuration

Feature	Default
Interface mode: • Before entering the **switchport** command • After entering the **switchport** command	 Layer 3 (unconfigured) **switchport mode dynamic desirable**
Trunk encapsulation	**switchport trunk encapsulation negotiate**
Allowed VLAN range	VLANs 1 to 4094, except reserved VLANs (see Table 23-1 on page 23-2)
VLAN range eligible for pruning	VLANs 2 to 1001
Default access VLAN	VLAN 1
Native VLAN (for 802.1Q trunks)	VLAN 1
Spanning Tree Protocol (STP)	Enabled for all VLANs
STP port priority	128
STP port cost	• 100 for 10-Mbps Ethernet LAN ports • 19 for 10/100-Mbps Fast Ethernet LAN ports • 19 for 100-Mbps Fast Ethernet LAN ports • 4 for 1,000-Mbps Gigabit Ethernet LAN ports • 2 for 10,000-Mbps 10-Gigabit Ethernet LAN ports

Layer 2 LAN Interface Configuration Guidelines and Restrictions

When configuring Layer 2 LAN ports, follow these guidelines and restrictions:

- The following switching modules do not support ISL encapsulation:
  - WS-X6502-10GE
  - WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6548-GE-45AF
  - WS-X6148-GE-TX, WS-X6148V-GE-TX, WS-X6148-GE-45AF

- The following configuration guidelines and restrictions apply when using 802.1Q trunks and impose some limitations on the trunking strategy for a network. Note these restrictions when using 802.1Q trunks:
  - When connecting Cisco switches through an 802.1q trunk, make sure the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.
  - Disabling spanning tree on the native VLAN of an 802.1Q trunk without disabling spanning tree on every VLAN in the network can cause spanning tree loops. We recommend that you leave spanning tree enabled on the native VLAN of an 802.1Q trunk. If this is not possible, disable spanning tree on every VLAN in the network. Make sure your network is free of physical loops before disabling spanning tree.
  - When you connect two Cisco switches through 802.1Q trunks, the switches exchange spanning tree BPDUs on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved IEEE 802.1d spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all other VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).
  - Non-Cisco 802.1Q switches maintain only a single instance of spanning tree (the Mono Spanning Tree, or MST) that defines the spanning tree topology for all VLANs. When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the MST of the non-Cisco switch and the native VLAN spanning tree of the Cisco switch combine to form a single spanning tree topology known as the Common Spanning Tree (CST).
  - Because Cisco switches transmit BPDUs to the SSTP multicast MAC address on VLANs other than the native VLAN of the trunk, non-Cisco switches do not recognize these frames as BPDUs and flood them on all ports in the corresponding VLAN. Other Cisco switches connected to the non-Cisco 802.1q cloud receive these flooded BPDUs. This allows Cisco switches to maintain a per-VLAN spanning tree topology across a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud separating the Cisco switches is treated as a single broadcast segment between all switches connected to the non-Cisco 802.1q cloud through 802.1q trunks.
  - Make certain that the native VLAN is the same on all of the 802.1q trunks connecting the Cisco switches to the non-Cisco 802.1q cloud.
  - If you are connecting multiple Cisco switches to a non-Cisco 802.1q cloud, all of the connections must be through 802.1q trunks. You cannot connect Cisco switches to a non-Cisco 802.1q cloud through ISL trunks or through access ports. Doing so causes the switch to place the ISL trunk port or access port into the spanning tree "port inconsistent" state and no traffic will pass through the port.

Configuring LAN Interfaces for Layer 2 Switching

These sections describe how to configure Layer 2 switching in Cisco IOS Release 12.2SX:

Note Use the **default interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* command to revert an interface to its default configuration.

Configuring a LAN Port for Layer 2 Switching

To configure a LAN port for Layer 2 switching, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`shutdown`**	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	`Router(config-if)#` **`switchport`**	Configures the LAN port for Layer 2 switching. **Note** You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 port before you can enter additional **switchport** commands with keywords.
Step 4	`Router(config-if)#` **`no shutdown`**	Activates the interface. (Required only if you shut down the interface.)
Step 5	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 6	`Router#` **`show running-config interface`** [*`type`*[1] *`slot/port`*]	Displays the running configuration of the interface.
Step 7	`Router#` **`show interfaces`** [*`type`*[1] *`slot/port`*] **`switchport`**	Displays the switch port configuration of the interface.
Step 8	`Router#` **`show interfaces`** [*`type`*[1] *`slot/port`*] **`trunk`**	Displays the trunk configuration of the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

After you enter the **switchport** command, the default mode is **switchport mode dynamic desirable**. If the neighboring port supports trunking and is configured to allow trunking, the link becomes a Layer 2 trunk when you enter the **switchport** command. By default, LAN trunk ports negotiate encapsulation. If the neighboring port supports ISL and 802.1Q encapsulation and both ports are set to negotiate the encapsulation type, the trunk uses ISL encapsulation (10-Gigabit Ethernet ports do not support ISL encapsulation).

Note When using the **switchport** command, if a port configured for Layer 3 is now configured for Layer 2, the configuration for Layer 3 is retained in the memory but not in the running configuration and is applied to the port whenever the port switches back to Layer 3. Also, if a port configured for Layer 2 is now configured for Layer 3, the configuration for Layer 2 is retained in the memory but not in the running configuration and is applied to the port whenever the port switches back to Layer 2. To restore the default configuration of the port in the memory and in the running configuration, use the **default interface** command. To avoid potential issues while changing the role of a port using the **switchport** command, shut down the interface before applying the **switchport** command.

Enabling Out-of-Band MAC Address Table Synchronization

With Release 12.2(33)SXF and later releases, to enable the out-of-band MAC address table synchronization feature, perform this task:

	Command	Purpose
Step 1	`Router(config)# [no] mac-address-table synchronize [activity-time seconds]`	Enables out-of-band synchronization of MAC address tables among DFC-equipped switching modules. • **activity-time** *seconds*—(Optional) Specifies the activity timer interval.
Step 2	`Router# show mac-address-table synchronize statistics`	Displays MAC address table synchronization statistics.

When configuring out-of-band MAC address table synchronization, note the following information:

- By default, out-of-band MAC address table synchronization is disabled.
- Out-of-band MAC address table synchronization is enabled automatically if any of the following conditions are met:
  - A WS-6708-10G switching module is installed in the switch.
  - The switch is part of a virtual switch system (VSS) running Cisco IOS Release 12.2(33)SXI4 or a later release.
- The activity timer interval can be configured as 160, 320, and 640 seconds. The default is 160 seconds.

This example shows how to enable out-of-band MAC address table synchronization:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mac-address-table synchronize activity-time 320
```

Configuring MAC Address Table Notification

Note

- Complete the steps in the "Configuring a LAN Port for Layer 2 Switching" section on page 17-7 before performing the tasks in this section.
- To send SNMP trap notifications using this feature, you must also enable the global MAC trap flag, using the **snmp-server enable mac-notification change** command.

With Release 12.2(33)SXH and later releases, to configure the MAC address table notification feature, perform this task:

	Command	Purpose
Step 1	Router(config)# **mac-address-table notification change** [**interval** *value*] [**history** *size*]	Enables sending notification of dynamic changes to MAC address table. (Optional) Sets the minimum change-sending interval in seconds. (Optional) Sets the number of entries in the history buffer. **Note** The **no** form of the command reverts to the default without sending any change information.
Step 2	Router(config)# **interface** *type*[1] *slot/port*	Selects the LAN port to configure.
Step 3	Router(config-if)# **snmp trap mac-notification change** [**added** | **removed**]	For MAC addresses that are associated with this LAN port, enable SNMP trap notification when MAC addresses are added to or removed from the address table. (Optional) To notify only when a MAC address is added to the table, use the **added** option. To notify only when a MAC address is removed, use the **removed** option.
Step 4	Router(config-if)# **end**	Exits interface configuration mode.
Step 5	Router# **show mac-address-table notification**	Displays whether this feature is enabled, the notification interval, and the history table maximum size. Displays history table contents.
Step 6	Router# **show mac-address-table notification** [*type slot/port*]	Displays the interface-specific flags for the specified interface. If slot and port are not specified, the flags for all interfaces will be displayed.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring the notification parameters, note the following information:

- The **interval** *value* parameter can be configured from 0 seconds (immediate) to 2,147,483,647 seconds. The default is 1 second.
- The **history** *size* parameter can be configured from 0 entries to 500 entries. The default is 1 entry.

This example shows how to configure the SNMP notification of dynamic additions to the MAC address table of addresses on the Fast Ethernet ports 5/7 and 5/8. Notifications of changes will be sent no more frequently than 5 seconds, and up to 25 changes can be stored and sent in that interval:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mac-address-table notification change interval 5 history 25
Router(config)# interface fastethernet 5/7
Router(config-if)# snmp trap mac-notification change added
Router(config-if)# end
Router(config)# interface fastethernet 5/8
Router(config-if)# snmp trap mac-notification change added
Router(config-if)# end
Router# exit
```

Configuring a Layer 2 Switching Port as a Trunk

These sections describe configuring a Layer 2 switching port as a trunk:

Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk

Note

- Complete the steps in the "Configuring a LAN Port for Layer 2 Switching" section on page 17-7 before performing the tasks in this section.
- When you enter the **switchport** command with no other keywords (Step 3 in the previous section), the default mode is **switchport mode dynamic desirable** and **switchport trunk encapsulation negotiate**.

To configure the Layer 2 switching port as an ISL or 802.1Q trunk, perform this task:

Command	Purpose
Router(config-if)# **switchport trunk encapsulation** {**isl** | **dot1q** | **negotiate**}	(Optional) Configures the encapsulation, which configures the Layer 2 switching port as either an ISL or 802.1Q trunk.

When configuring the Layer 2 switching port as an ISL or 802.1Q trunk, note the following information:

- The **switchport mode trunk** command (see the "Configuring the Layer 2 Trunk Not to Use DTP" section on page 17-11) is not compatible with the **switchport trunk encapsulation negotiate** command.
- To support the **switchport mode trunk** command, you must configure the encapsulation as either ISL or 802.1Q.
- The following switching modules do not support ISL encapsulation:
  - WS-X6502-10GE
  - WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6548-GE-45AF
  - WS-X6148-GE-TX, WS-X6148V-GE-TX, WS-X6148-GE-45AF

Note Complete the steps in the "Completing Trunk Configuration" section on page 17-15 after performing the tasks in this section.

Configuring the Layer 2 Trunk to Use DTP

Note Complete the steps in the "Configuring a LAN Port for Layer 2 Switching" section on page 17-7 before performing the tasks in this section.

To configure the Layer 2 trunk to use DTP, perform this task:

Command	Purpose
`Router(config-if)# switchport mode dynamic {auto | desirable}`	(Optional) Configures the trunk to use DTP. **Note** The **no** form of the command reverts to the default trunk trunking mode (**switchport mode dynamic desirable**).

When configuring the Layer 2 trunk to use DTP, note the following information:

- Required only if the interface is a Layer 2 access port or to specify the trunking mode.
- See Table 17-2 on page 17-4 for information about trunking modes.

Note Complete the steps in the "Completing Trunk Configuration" section on page 17-15 after performing the tasks in this section.

Configuring the Layer 2 Trunk Not to Use DTP

Note Complete the steps in the "Configuring a LAN Port for Layer 2 Switching" section on page 17-7 before performing the tasks in this section.

To configure the Layer 2 trunk not to use DTP, perform this task:

	Command	Purpose
Step 1	`Router(config-if)# switchport mode trunk`	(Optional) Configures the port to trunk unconditionally.
Step 2	`Router(config-if)# switchport nonegotiate`	(Optional) Configures the trunk not to use DTP. **Note** The **no** form of the command enables DTP on the port.

When configuring the Layer 2 trunk not to use DTP, note the following information:

- Before entering the **switchport mode trunk** command, you must configure the encapsulation (see the "Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk" section on page 17-10).
- To support the **switchport nonegotiate** command, you must enter the **switchport mode trunk** command.
- Enter the **switchport mode dynamic trunk** command. See Table 17-2 on page 17-4 for information about trunking modes.
- Before entering the **switchport nonegotiate** command, you must configure the encapsulation (see the "Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk" section on page 17-10) and configure the port to trunk unconditionally with the **switchport mode trunk** command (see the "Configuring the Layer 2 Trunk to Use DTP" section on page 17-11).

Note Complete the steps in the "Completing Trunk Configuration" section on page 17-15 after performing the tasks in this section.

Configuring the Access VLAN

Note Complete the steps in the "Configuring a LAN Port for Layer 2 Switching" section on page 17-7 before performing the tasks in this section.

To configure the access VLAN, perform this task:

Command	Purpose
`Router(config-if)# switchport access vlan vlan_ID`	(Optional) Configures the access VLAN, which is used if the interface stops trunking. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2). **Note** • If VLAN locking is enabled, enter the VLAN name instead of the VLAN number. For more information, see the "VLAN Locking" section on page 23-4. • The **no** form of the command reverts to the default VLAN (VLAN 1).

Note Complete the steps in the "Completing Trunk Configuration" section on page 17-15 after performing the tasks in this section.

Configuring the 802.1Q Native VLAN

Note Complete the steps in the "Configuring a LAN Port for Layer 2 Switching" section on page 17-7 before performing the tasks in this section.

To configure the 802.1Q native VLAN, perform this task:

Command	Purpose
Router(config-if)# **switchport trunk native vlan** *vlan_ID*	(Optional) Configures the 802.1Q native VLAN. **Note** If VLAN locking is enabled, enter the VLAN name instead of the VLAN number. For more information, see the "VLAN Locking" section on page 23-4.

When configuring the native VLAN, note the following information:

- The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2).
- The access VLAN is not automatically used as the native VLAN.

Note Complete the steps in the "Completing Trunk Configuration" section on page 17-15 after performing the tasks in this section.

Configuring the List of VLANs Allowed on a Trunk

Note Complete the steps in the "Configuring a LAN Port for Layer 2 Switching" section on page 17-7 before performing the tasks in this section.

To configure the list of VLANs allowed on a trunk, perform this task:

Command	Purpose
Router(config-if)# **switchport trunk allowed vlan** {**add** | **except** | **none** | **remove**} *vlan* [,*vlan*[,*vlan*[,...]]	(Optional) Configures the list of VLANs allowed on the trunk. **Note** • If VLAN locking is enabled, enter VLAN names instead of VLAN numbers. For more information, see the "VLAN Locking" section on page 23-4. • The **no** form of the command reverts to the default value (all VLANs allowed).

When configuring the list of VLANs allowed on a trunk, note the following information:

- The *vlan* parameter is either a single VLAN number from 1 through 4094, or a range of VLANs described by two VLAN numbers, the lesser one first, separated by a dash. Do not enter any spaces between comma-separated *vlan* parameters or in dash-specified ranges.
- If VLAN locking is enabled, enter VLAN names instead of VLAN numbers. When entering a range of VLAN names, you must leave spaces between the VLAN names and the dash.
- All VLANs are allowed by default.
- You can remove VLAN 1. If you remove VLAN 1 from a trunk, the trunk interface continues to send and receive management traffic, for example, Cisco Discovery Protocol (CDP), VLAN Trunking Protocol (VTP), Port Aggregation Protocol (PAgP), and DTP in VLAN 1.

Note Complete the steps in the "Completing Trunk Configuration" section on page 17-15 after performing the tasks in this section.

Configuring the List of Prune-Eligible VLANs

Note Complete the steps in the "Configuring a LAN Port for Layer 2 Switching" section on page 17-7 before performing the tasks in this section.

To configure the list of prune-eligible VLANs on the Layer 2 trunk, perform this task:

Command	Purpose
Router(config-if)# **switchport trunk pruning vlan** {**none** |{{**add** | **except** | **remove**} *vlan*[,*vlan*[,*vlan*[,...]]]}}	(Optional) Configures the list of prune-eligible VLANs on the trunk (see the "Understanding VTP Pruning" section on page 22-6). **Note** The **no** form of the command reverts to the default value (all VLANs prune-eligible).

When configuring the list of prune-eligible VLANs on a trunk, note the following information:

- The *vlan* parameter is either a single VLAN number from 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2), or a range of VLANs described by two VLAN numbers, the lesser one first, separated by a dash. Do not enter any spaces between comma-separated *vlan* parameters or in dash-specified ranges.
- The default list of VLANs allowed to be pruned contains all VLANs.
- Network devices in VTP transparent mode do not send VTP Join messages. On trunk connections to network devices in VTP transparent mode, configure the VLANs used by the transparent-mode network devices or that need to be carried across the transparent-mode network devices as pruning ineligible.

Note Complete the steps in the "Completing Trunk Configuration" section on page 17-15 after performing the tasks in this section.

Completing Trunk Configuration

To complete Layer 2 trunk configuration, perform this task:

	Command	Purpose
Step 1	`Router(config-if)# no shutdown`	Activates the interface. (Required only if you shut down the interface.)
Step 2	`Router(config-if)# end`	Exits configuration mode.

Verifying Layer 2 Trunk Configuration

To verify Layer 2 trunk configuration, perform this task:

	Command	Purpose
Step 1	Router# **show running-config interface** *type*[1] *slot/port*	Displays the running configuration of the interface.
Step 2	Router# **show interfaces** [*type*[1] *slot/port*] **switchport**	Displays the switch port configuration of the interface.
Step 3	Router# **show interfaces** [*type*[1] *slot/port*] **trunk**	Displays the trunk configuration of the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Configuration and Verification Examples

This example shows how to configure the Fast Ethernet port 5/8 as an 802.1Q trunk. This example assumes that the neighbor port is configured to support 802.1Q trunking:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/8
Router(config-if)# shutdown
Router(config-if)# switchport
Router(config-if)# switchport mode dynamic desirable
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# no shutdown
Router(config-if)# end
Router# exit
```

This example shows how to verify the configuration:

```
Router# show running-config interface fastethernet 5/8
Building configuration...
Current configuration:
!
interface FastEthernet5/8
 no ip address
 switchport
 switchport trunk encapsulation dot1q
end

Router# show interfaces fastethernet 5/8 switchport
Name: Fa5/8
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
```

```
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: ALL

Router# show interfaces fastethernet 5/8 trunk

Port      Mode         Encapsulation  Status        Native vlan
Fa5/8     desirable    n-802.1q       trunking      1

Port      Vlans allowed on trunk
Fa5/8 1-1005

Port      Vlans allowed and active in management domain
Fa5/8 1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-8
02,850,917,999,1002-1005

Port      Vlans in spanning tree forwarding state and not pruned
Fa5/8 1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-8
02,850,917,999,1002-1005

Router#
```

Configuring a LAN Interface as a Layer 2 Access Port

Note If you assign a LAN port to a VLAN that does not exist, the port is shut down until you create the VLAN in the VLAN database (see the "Creating or Modifying an Ethernet VLAN" section on page 23-5).

To configure a LAN port as a Layer 2 access port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`shutdown`**	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	`Router(config-if)#` **`switchport`**	Configures the LAN port for Layer 2 switching. **Note** You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 port before you can enter additional **switchport** commands with keywords.
Step 4	`Router(config-if)#` **`switchport mode access`**	Configures the LAN port as a Layer 2 access port.
Step 5	`Router(config-if)#` **`switchport access vlan`** *`vlan_ID`*	Places the LAN port in a VLAN. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2). **Note** If VLAN locking is enabled, enter the VLAN name instead of the VLAN number. For more information, see the "VLAN Locking" section on page 23-4.

	Command	Purpose
Step 6	Router(config-if)# **no shutdown**	Activates the interface. (Required only if you shut down the interface.)
Step 7	Router(config-if)# **end**	Exits configuration mode.
Step 8	Router# **show running-config interface** [*type*[1] *slot/port*]	Displays the running configuration of the interface.
Step 9	Router# **show interfaces** [*type*[1] *slot/port*] **switchport**	Displays the switch port configuration of the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure the Fast Ethernet port 5/6 as an access port in VLAN 200:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/6
Router(config-if)# shutdown
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# switchport access vlan 200
Router(config-if)# no shutdown
Router(config-if)# end
Router# exit
```

This example shows how to verify the configuration:

```
Router# show running-config interface fastethernet 5/6
Building configuration...
!
Current configuration:
interface FastEthernet5/6
 no ip address
 switchport access vlan 200
 switchport mode access
end

Router# show interfaces fastethernet 5/6 switchport
Name: Fa5/6
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Enabled
Access Mode VLAN: 200 (VLAN0200)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: ALL

Router#
```

Configuring an IEEE 802.1Q Custom EtherType Field Value

You can configure a custom EtherType field value on a port to support network devices that do not use the standard 0x8100 EtherType field value on 802.1Q-tagged or 802.1p-tagged frames.

To configure a custom value for the EtherType field, perform this task:

Command	Purpose
`Router(config-if)# switchport dot1q ethertype value`	Configures the 802.1Q EtherType field value for the port.

When configuring a custom EtherType field value, note the following information:

- To use a custom EtherType field value, all network devices in the traffic path across the network must support the custom EtherType field value.
- You can configure a custom EtherType field value on trunk ports, access ports, and tunnel ports.
- You can configure a custom EtherType field value on the member ports of an EtherChannel.
- You cannot configure a custom EtherType field value on a port-channel interface.
- Each port supports only one EtherType field value. A port that is configured with a custom EtherType field value does not recognize frames that have any other EtherType field value as tagged frames. For example, a trunk port that is configured with a custom EtherType field value does not recognize the standard 0x8100 EtherType field value on 802.1Q-tagged frames and cannot put the frames into the VLAN to which they belong.

Caution A port that is configured with a custom EtherType field value considers frames that have any other EtherType field value to be untagged frames. A trunk port with a custom EtherType field value places frames with any other EtherType field value into the native VLAN. An access port or tunnel port with a custom EtherType field value places frames that are tagged with any other EtherType field value into the access VLAN. If you misconfigure a custom EtherType field value, frames might be placed into the wrong VLAN.

- See the *Release Notes for Cisco IOS Release 12.2SX* for a list of the modules that support custom IEEE 802.1Q EtherType field values.

This example shows how to configure the EtherType field value to 0x1234:

```
Router (config-if)# switchport dot1q ethertype 1234
Router (config-if)#
```

Configuring Flex Links

This chapter describes how to configure Flex Links in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

The chapter consists of these sections:

Understanding Flex Links

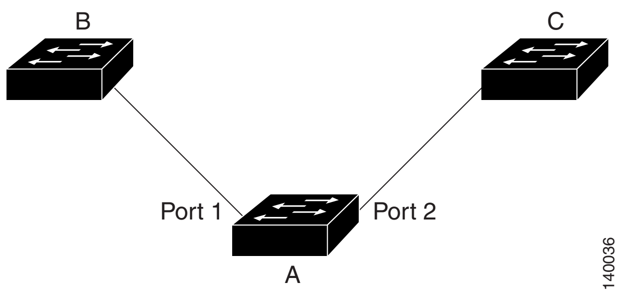
Flex Links are a pair of a Layer 2 interfaces (ports or port channels), where one interface is configured to act as a backup to the other. Flex Links are typically configured in service-provider or enterprise networks where customers do not want to run STP. Flex Links provide link-level redundancy that is an alternative to Spanning Tree Protocol (STP). STP is automatically disabled on Flex Links interfaces.

Release 12.2SX supports a maximum of 16 Flex Links. Flex Links are supported only on Layer 2 ports and port channels, not on VLANs or on Layer 3 ports.

To configure the Flex Links feature, you configure one Layer 2 interface as the standby link for the link that you want to be primary. With Flex Links configured for a pair of interfaces, only one of the interfaces is in the linkup state and is forwarding traffic. If the primary link shuts down, the standby link starts forwarding traffic. When the inactive link comes back up, it goes into standby mode.

In Figure 18-1, ports 1 and 2 on switch A are connected to uplink switches B and C. Because they are configured as Flex Links, only one of the interfaces is forwarding traffic and the other one is in standby mode. If port 1 is the active link, it begins forwarding traffic between port 1 and switch B; the link between port 2 (the backup link) and switch C is not forwarding traffic. If port 1 goes down, port 2 comes up and starts forwarding traffic to switch C. When port 1 comes back up, it goes into standby mode and does not forward traffic; port 2 continues to forward traffic.

Figure 18-1 Flex Links Configuration Example



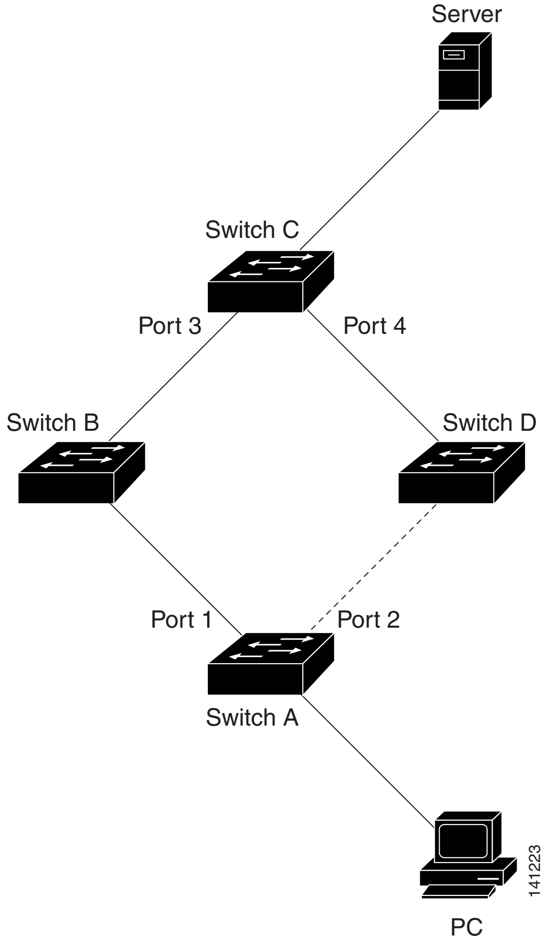
If a primary (forwarding) link goes down, a trap notifies the network management stations. If the standby link goes down, a trap notifies the users. When a primary link fails, the feature takes these actions:

- Detects the failure.
- Moves any dynamic unicast MAC addresses that are learned on the primary link to the standby link.
- Moves the standby link to a forwarding state.
- Transmits dummy multicast packets over the new active interface. The dummy multicast packet format is:
  - Destination: 01:00:0c:cd:cd:cd
  - Source: MAC address of the hosts or ports on the newly active Flex Link port.

  In Figure 18-2, ports 1 and 2 on switch A are connected to switches B and D through a Flex Link pair. Port 1 is forwarding traffic, and port 2 is in the blocking state. Traffic from the PC to the server is forwarded from port 1 to port 3. The MAC address of the PC has been learned on port 3 of switch C. Traffic from the server to the PC is forwarded from port 3 to port 1.

  If port 1 shuts down, port 2 starts forwarding traffic. If there is no traffic from the PC to the server after failover to port 2, switch C does not learn the MAC address of the PC on port 4, and because of that, switch C keeps forwarding traffic from the server to the PC out of port 3. There is traffic loss from the server to the PC because port 1 is down. To alleviate this problem, the feature sends out a dummy multicast packet with the source MAC address of the PC over port 2. Switch C learns the PC MAC address on port 4 and start forwarding traffic from the server to the PC out of port 4. One dummy multicast packet is sent out for every MAC address.

Figure 18-2 Flexlink Dummy Multicast Packets Example



Note

- Local administrative shut down or a link that starts forwarding again due to preemption is not considered a link failure. In those cases, the feature flushes the dynamic hosts and and does not move them.
- Static MAC addresses configured on a Flex Link port are restored when it starts forwarding again.

Configuring Flex Links

These sections contain this configuration information:

- Flex Links Default Configuration, page 18-4
- Flex Links Configuration Guidelines and Restrictions, page 18-4
- Configuring Flex Links, page 18-4

Flex Links Default Configuration

There is no default Flex Links configuration.

Flex Links Configuration Guidelines and Restrictions

When configuring Flex Links, follow these guidelines and restrictions:

- You can configure only one Flex Links backup link for any active link, and it must be a different interface from the active interface.
- An interface can belong to only one Flex Links pair. An interface can be a backup link for only one active link. An active link cannot belong to another Flex Links pair.
- Neither of the links can be a port that belongs to an EtherChannel. However, you can configure two port channels (EtherChannel logical interfaces) as Flex Links, and you can configure a port channel and a physical interface as Flex Links, with either the port channel or the physical interface as the active link.
- A backup link does not have to be the same type as the active link (Fast Ethernet, Gigabit Ethernet, or port channel). However, you should configure both Flex Links with similar characteristics so that there are no loops or changes in operation if the standby link becomes active.
- STP is disabled on Flex Links ports. If STP is disabled on the switch, be sure that there are no Layer 2 loops in the network topology.
- Do not configure the following STP features on Flex Links ports or the ports to which the links connect:
  - Bridge Assurance
  - UplinkFast
  - BackboneFast
  - EtherChannel Guard
  - Root Guard
  - Loop Guard
  - PVST Simulation
- Flex Links and port security are not compatible with each other.
- Static unicast MAC addresses that are configured on the primary link are not moved to the standby link.

Configuring Flex Links

To configure Flex Links, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(conf)#` **`interface`** `{{`*`type`*[1] *`slot/port`*`} |` `{`**`port-channel`** *`number`*`}}`	Specifies a Layer 2 interface.

	Command	Purpose
Step 3	`Router(conf-if)#` **`switchport backup interface`** `{{`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`number`*`}}`	Configures the interface as part of a Flex Links pair.
Step 4	`Router(conf-if)#` **`exit`**	Exits configuration mode.
Step 5	`Router#` **`show interface`** `[{`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`number`*`}]` **`switchport backup`**	Verifies the configuration.
Step 6	`Router#` **`copy running-config startup config`**	(Optional) Saves your entries in the switch startup configuration file.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure an interface with a backup interface and how to verify the configuration:

```
Router# configure terminal
Router(conf)# interface fastethernet1/1
Router(conf-if)# switchport backup interface fastethernet1/2
Router(conf-if)# exit
Router# show interface switchport backup
Router Backup Interface Pairs:

Active Interface        Backup Interface        State
------------------------------------------------------------------------------------------
FastEthernet1/1         FastEthernet1/2       Active Up/Backup Standby
FastEthernet1/3         FastEthernet2/4       Active Up/Backup Standby
Port-channel1           GigabitEthernet7/1    Active Up/Backup Standby
```

Monitoring Flex Links

To monitor the Flex Links configuration, perform this task:

Command	Purpose
`show interface` `[{`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`number`*`}]` **`switchport backup`**	Displays the Flex Links backup interface configured for an interface, or displays all Flex Links configured on the switch and the state of each active and backup interface (up or standby mode).

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring EtherChannels

This chapter describes how to configure EtherChannels on Layer 2 or Layer 3 LAN ports in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding EtherChannels

These sections describe how EtherChannels work:

EtherChannel Feature Overview

An EtherChannel bundles individual Ethernet links into a single logical link that provides the aggregate bandwidth of up to eight physical links.

Cisco IOS Release 12.2SX supports a maximum of 128 EtherChannels. You can form an EtherChannel with up to eight compatibly configured LAN ports on any switching module. All LAN ports in each EtherChannel must be the same speed and must all be configured as either Layer 2 or Layer 3 LAN ports.

Note The network device to which a switch is connected may impose its own limits on the number of ports in an EtherChannel.

If a segment within an EtherChannel fails, traffic previously carried over the failed link switches to the remaining segments within the EtherChannel. When a failure occurs, the EtherChannel feature sends a trap that identifies the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one segment in an EtherChannel are blocked from returning on any other segment of the EtherChannel.

Understanding EtherChannel Configuration

These sections describe how EtherChannels are configured:

- EtherChannel Configuration Overview, page 19-2
- Understanding Manual EtherChannel Configuration, page 19-3
- Understanding PAgP EtherChannel Configuration, page 19-3
- Understanding IEEE 802.3ad LACP EtherChannel Configuration, page 19-4

EtherChannel Configuration Overview

You can configure EtherChannels manually or you can use the Port Aggregation Control Protocol (PAgP) or the Link Aggregation Control Protocol (LACP) to form EtherChannels. The EtherChannel protocols allow ports with similar characteristics to form an EtherChannel through dynamic negotiation with connected network devices. PAgP is a Cisco-proprietary protocol and LACP is defined in IEEE 802.3ad.

PAgP and LACP do not interoperate with each other. Ports configured to use PAgP cannot form EtherChannels with ports configured to use LACP. Ports configured to use LACP cannot form EtherChannels with ports configured to use PAgP. Neither interoperates with ports configured manually.

Table 19-1 lists the user-configurable EtherChannel modes.

Table 19-2 lists the EtherChannel member port states.

Table 19-1 EtherChannel Modes

Mode	Description
on	Mode that forces the LAN port to channel unconditionally. In the **on** mode, a usable EtherChannel exists only when a LAN port group in the **on** mode is connected to another LAN port group in the **on** mode. Because ports configured in the **on** mode do not negotiate, there is no negotiation traffic between the ports. You cannot configure the **on** mode with an EtherChannel protocol. If one end uses the **on** mode, the other end must also.
auto	PAgP mode that places a LAN port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not initiate PAgP negotiation. (Default)
desirable	PAgP mode that places a LAN port into an active negotiating state, in which the port initiates negotiations with other LAN ports by sending PAgP packets.
passive	LACP mode that places a port into a passive negotiating state, in which the port responds to LACP packets it receives but does not initiate LACP negotiation. (Default)
active	LACP mode that places a port into an active negotiating state, in which the port initiates negotiations with other ports by sending LACP packets.

Table 19-2 EtherChannel Member Port States

Port States	Description
bundled	The port is part of an EtherChannel and can send and receive BPDUs and data traffic.
suspended	The port is not part of an EtherChannel. The port can receive BPDUs but cannot send them. Data traffic is blocked.
standalone	The port is not bundled in an EtherChannel. The port functions as a standalone data port. The port can send and receive BPDUs and data traffic. **Note** When one end of an EtherChannel has more members than the other, the unmatched ports enter the standalone state. In a topology that is not protected from Layer 2 loops by the spanning tree protocol (STP), a port in the standalone state can cause significant network errors. For LACP EtherChannels with Release 12.2(33)SXI3 and later releases, you can enter the **port-channel standalone-disable** interface configuration mode command to put ports into the suspended state instead of the standalone state. See the "Configuring LACP Port-Channel Standalone Disable" section on page 19-17.

Understanding Manual EtherChannel Configuration

Manually configured EtherChannel ports do not exchange EtherChannel protocol packets. A manually configured EtherChannel forms only when you configure all ports in the EtherChannel compatibly.

Understanding PAgP EtherChannel Configuration

PAgP supports the automatic creation of EtherChannels by exchanging PAgP packets between LAN ports. PAgP packets are exchanged only between ports in **auto** and **desirable** modes.

The protocol learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once PAgP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

Both the **auto** and **desirable** modes allow PAgP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. Layer 2 EtherChannels also use VLAN numbers.

LAN ports can form an EtherChannel when they are in different PAgP modes if the modes are compatible. For example:

- A LAN port in **desirable** mode can form an EtherChannel successfully with another LAN port that is in **desirable** mode.
- A LAN port in **desirable** mode can form an EtherChannel with another LAN port in **auto** mode.
- A LAN port in **auto** mode cannot form an EtherChannel with another LAN port that is also in **auto** mode, because neither port will initiate negotiation.

Understanding IEEE 802.3ad LACP EtherChannel Configuration

LACP supports the automatic creation of EtherChannels by exchanging LACP packets between LAN ports. LACP packets are exchanged only between ports in **passive** and **active** modes.

The protocol learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

Both the **passive** and **active** modes allow LACP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. Layer 2 EtherChannels also use VLAN numbers.

LAN ports can form an EtherChannel when they are in different LACP modes as long as the modes are compatible. For example:

- A LAN port in **active** mode can form an EtherChannel successfully with another LAN port that is in **active** mode.
- A LAN port in **active** mode can form an EtherChannel with another LAN port in **passive** mode.
- A LAN port in **passive** mode cannot form an EtherChannel with another LAN port that is also in **passive** mode, because neither port will initiate negotiation.

LACP uses the following parameters:

- LACP system priority—You must configure an LACP system priority on each switch running LACP. The system priority can be configured automatically or through the CLI (see the "Configuring the LACP System Priority and System ID" section on page 19-11). LACP uses the system priority with the switch MAC address to form the system ID and also during negotiation with other systems.

  Note The LACP system ID is the combination of the LACP system priority value and the MAC address of the switch.

- LACP port priority—You must configure an LACP port priority on each port configured to use LACP. The port priority can be configured automatically or through the CLI (see the "Configuring Channel Groups" section on page 19-9). LACP uses the port priority with the port number to form the port identifier. LACP uses the port priority to decide which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating.
- LACP auto interleaved port priority—You can configure LACP auto interleaved port priority to create an effective distribution of bundled and hot standby ports across different slots that are part of the same port channel, either Distributed EtherChannel (DEC) or Multichassis EtherChannel (MEC). To configure auto interleaved port priority use the **lacp active-port distribution automatic**

command. The bundled port distribution can be configured through the CLI (see the "Configuring Channel Groups" section on page 19-9). Once the auto interleaved port priority feature is enabled, it automatically distributes bundled ports based on the position of when a port link comes up and is effective only if you configure it on the system that has the higher LACP system priority. You need to perform a shutdown and no shutdown on the interface port channel to enable the auto interleaved port priority feature on all ports.

- LACP administrative key—LACP automatically configures an administrative key value equal to the channel group identification number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port's ability to aggregate with other ports is determined by these factors:
  - Port physical characteristics, such as data rate, duplex capability, and point-to-point or shared medium
  - Configuration restrictions that you establish

On ports configured to use LACP, LACP tries to configure the maximum number of compatible ports in an EtherChannel, up to the maximum allowed by the hardware (eight ports). If LACP cannot aggregate all the ports that are compatible (for example, the remote system might have more restrictive hardware limitations), then all the ports that cannot be actively included in the channel are put in hot standby state and are used only if one of the channeled ports fails. You can configure an additional 8 standby ports (total of 16 ports associated with the EtherChannel).

Understanding LACP 1:1 Redundancy

With Release 12.2(33)SXH and later releases, the LACP 1:1 redundancy feature provides an EtherChannel configuration with one active link and fast switchover to a hot standby link.

To use LACP 1:1 redundancy, you configure an LACP EtherChannel with two ports (one active and one standby). If the active link goes down, the EtherChannel stays up and the system performs fast switchover to the hot standby link. When the failed link becomes operational again, the EtherChannel performs another fast switchover to revert to the original active link.

For the LACP 1:1 redundancy feature to work correctly, especially the fast switchover capability, the feature needs to be enabled at both ends of the link.

Understanding Port Channel Interfaces

Each EtherChannel has a numbered port channel interface. You can configure a maximum of 128 port-channel interfaces, numbered from 1 to 256. The configuration that you apply to the port channel interface affects all LAN ports assigned to the port channel interface.

After you configure an EtherChannel, the configuration that you apply to the port channel interface affects the EtherChannel; the configuration that you apply to the LAN ports affects only the LAN port where you apply the configuration. To change the parameters of all ports in an EtherChannel, apply the configuration commands to the port channel interface, for example, Spanning Tree Protocol (STP) commands or commands to configure a Layer 2 EtherChannel as a trunk.

Understanding Load Balancing

An EtherChannel balances the traffic load across the links in an EtherChannel by reducing part of the binary information in the frame header to a numerical value that selects one of the links in the channel.

You can configure which header information will be used to derive the numerical value for link selection. The EtherChannel load balancing method can use MAC addresses, IP addresses, or Layer 4 port numbers, and you can configure whether the information will be from the source or destination or both source and destination addresses or ports. The selected mode applies to all EtherChannels configured on the switch. EtherChannel load balancing can also use MPLS Layer 2 information. If PFC3C/CXL hardware is installed, EtherChannel load balancing will additionally consider VLAN information by default.

Use the option that provides the balance criteria with the greatest variety in your configuration. For example, if the traffic on an EtherChannel is going only to a single MAC address and you use the destination MAC address as the basis of EtherChannel load balancing, the EtherChannel always chooses the same link in the EtherChannel; using source addresses or IP addresses might result in better load balancing.

EtherChannel Feature Configuration Guidelines and Restrictions

When EtherChannel interfaces are configured improperly, they are disabled automatically to avoid network loops and other problems. To avoid configuration problems, observe these guidelines and restrictions:

- The commands in this chapter can be used on all Layer 2 Ethernet ports, including the ports on the supervisor engine and a redundant supervisor engine.
- The WS-X6148-GE-TX and WS-X6148V-GE-TX switching modules do not support more than 1 Gbps of traffic per EtherChannel.
- When you add a member port that does not support ISL trunking to an EtherChannel, Cisco IOS software automatically adds a **switchport trunk encapsulation dot1q** command to the port-channel interface to prevent configuration of the EtherChannel as an ISL trunk. The **switchport trunk encapsulation dot1q** command is inactive when the EtherChannel is not a trunk.
- All Layer 2 Ethernet ports on all modules, including those on a redundant supervisor engine, support EtherChannels (maximum of eight LAN ports) with no requirement that the LAN ports be physically contiguous or on the same module.
- A Distributed EtherChannel (DEC) is an EtherChannel with member ports that are served by the PFC and one or more DFCs or by multiple DFCs. On switching modules with dual switch-fabric connections, a DEC can also be a single-module EtherChannel. Search the release notes for “Dual switch-fabric connections”.
- Configure all LAN ports in an EtherChannel to use the same EtherChannel protocol; you cannot run two EtherChannel protocols in one EtherChannel.
- Configure all LAN ports in an EtherChannel to operate at the same speed and in the same duplex mode.
- LACP does not support half-duplex. Half-duplex ports in an LACP EtherChannel are put in the suspended state.
- Enable all LAN ports in an EtherChannel. If you shut down a LAN port in an EtherChannel, it is treated as a link failure and its traffic is transferred to one of the remaining ports in the EtherChannel.
- An EtherChannel will not form if one of the LAN ports is a Switched Port Analyzer (SPAN) destination port.
- For Layer 3 EtherChannels, assign Layer 3 addresses to the port channel logical interface, not to the LAN ports in the channel.

- Layer 3 EtherChannels of all types and single-module non-DEC EtherChannels offer the highest throughput.
- CSCti23324 is resolved in Release 12.2(33)SXJ1 and later releases. In releases where CSCti23324 is not resolved, Layer 2 DECs have lower throughput because Layer 2 DEC traffic uses packet recirculation. If possible, configure nondistributed Layer 2 EtherChannels or Layer 3 DECs.
- When the switch is in bus mode (also called flow-through mode), Layer 2 multi-module EtherChannels have lower throughput because Layer 2 multi-module EtherChannel traffic uses packet recirculation. If possible, configure single-module EtherChannels or upgrade the installed hardware so that the switch does not operate in bus mode.
- For Layer 2 EtherChannels:
  - Assign all LAN ports in the EtherChannel to the same VLAN or configure them as trunks.
  - If you configure an EtherChannel from trunking LAN ports, verify that the trunking mode is the same on all the trunks. LAN ports in an EtherChannel with different trunk modes can operate unpredictably.
  - An EtherChannel supports the same allowed range of VLANs on all the LAN ports in a trunking Layer 2 EtherChannel. If the allowed range of VLANs is not the same, the LAN ports do not form an EtherChannel.
  - LAN ports with different STP port path costs can form an EtherChannel as long they are compatibly configured with each other. If you set different STP port path costs, the LAN ports are not incompatible for the formation of an EtherChannel.
  - An EtherChannel will not form if protocol filtering is set differently on the LAN ports.
  - Configure static MAC addresses on the EtherChannel only and not on physical member ports of the EtherChannel.
- After you configure an EtherChannel, the configuration that you apply to the port channel interface affects the EtherChannel. The configuration that you apply to the LAN ports affects only the LAN port where you apply the configuration.
- When QoS is enabled, enter the **no mls qos channel-consistency** port-channel interface command to support EtherChannels that have ports with different queue structures, for example, ports with and without strict-priority queues.

Caution Serious traffic problems can result from mixing manual mode with PAgP or LACP modes, or with a port with no EtherChannel configured. For example, if a port configured in **on** mode is connected to another port configured in **desirable** mode, or to a port not configured for EtherChannel, a bridge loop is created and a broadcast storm can occur. If one end uses the **on** mode, the other end must also.

Serious traffic problems can result if an EtherChannel forms from ports that pass data through the switch in significantly different ways. For example, ports on modules with and without DFCs, or when enabled with the **no mls qos channel-consistency** port-channel interface command, ports that have significantly different QoS port parameters (buffers sizes and queue types). Be prepared to disable such EtherChannels.

Configuring EtherChannels

These sections describe how to configure EtherChannels:

- Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels, page 19-8
- Configuring Channel Groups, page 19-9
- Configuring EtherChannel Load Balancing, page 19-12
- Configuring EtherChannel Hash-Distribution Algorithm, page 19-13
- Configuring the EtherChannel Min-Links Feature, page 19-14
- Configuring LACP 1:1 Redundancy, page 19-15
- Configuring Auto Interleaved Port Priority For LACP Port Channels, page 19-16
- Configuring LACP Port-Channel Standalone Disable, page 19-17

Note Make sure that the LAN ports are configured correctly (see the "EtherChannel Feature Configuration Guidelines and Restrictions" section on page 19-6).

Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels

Note

- When configuring Layer 2 EtherChannels, you cannot put Layer 2 LAN ports into manually created port channel logical interfaces. If you are configuring a Layer 2 EtherChannel, do not perform the procedures in this section (see the "Configuring Channel Groups" section on page 19-9).
- When configuring Layer 3 EtherChannels, you must manually create the port channel logical interface as described in this section, and then put the Layer 3 LAN ports into the channel group (see the "Configuring Channel Groups" section on page 19-9).
- To move an IP address from a Layer 3 LAN port to an EtherChannel, you must delete the IP address from the Layer 3 LAN port before configuring it on the port channel logical interface.

To create a port channel interface for a Layer 3 EtherChannel, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface port-channel group_number`	Creates the port channel interface.
Step 2	`Router(config-if)# ip address ip_address mask`	Assigns an IP address and subnet mask to the EtherChannel.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show running-config interface port-channel group_number`	Verifies the configuration.

The *group_number* can be 1 through 256, up to a maximum of 128 port-channel interfaces. This example shows how to create port channel interface 1:

```
Router# configure terminal
Router(config)# interface port-channel 1
Router(config-if)# ip address 172.32.52.10 255.255.255.0
```

```
Router(config-if)# end
```

This example shows how to verify the configuration of port channel interface 1:

```
Router# show running-config interface port-channel 1
Building configuration...

Current configuration:
!
interface Port-channel1
 ip address 172.32.52.10 255.255.255.0
 no ip directed-broadcast
end
Router#
```

Configuring Channel Groups

Note

- When configuring Layer 3 EtherChannels, you must manually create the port channel logical interface first (see the "Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels" section on page 19-8), and then put the Layer 3 LAN ports into the channel group as described in this section.
- When configuring Layer 2 EtherChannels, configure the LAN ports with the **channel-group** command as described in this section, which automatically creates the port channel logical interface. You cannot put Layer 2 LAN ports into a manually created port channel interface.
- For Cisco IOS to create port channel interfaces for Layer 2 EtherChannels, the Layer 2 LAN ports must be connected and functioning.

To configure channel groups, perform this task for each LAN port:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Selects a LAN port to configure.
Step 2	`Router(config-if)# no ip address`	Ensures that there is no IP address assigned to the LAN port.
Step 3	`Router(config-if)# channel-protocol (lacp | pagp}`	(Optional) On the selected LAN port, restricts the **channel-group** command to the EtherChannel protocol configured with the **channel-protocol** command.
Step 4	`Router(config-if)# channel-group` *group_number* `mode {active | auto | desirable | on | passive}`	Configures the LAN port in a port channel and specifies the mode (see Table 19-1 on page 19-3). PAgP supports only the auto and desirable modes. LACP supports only the active and passive modes.
Step 5	`Router(config-if)# lacp port-priority` *priority_value*	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.
Step 6	`Router(config-if)# shutdown`	Disables the interface.
Step 7	`Router(config-if)# no shutdown`	Enables the interface.

	Command	Purpose
Step 8	`Router(config-if)# end`	Exits configuration mode.
Step 9	`Router# show running-config interface type[1] slot/port` `Router# show interfaces type[1] slot/port etherchannel`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure Fast Ethernet ports 5/6 and 5/7 into port channel 2 with PAgP mode **desirable**:

```
Router# configure terminal
Router(config)# interface range fastethernet 5/6 -7
Router(config-if)# channel-group 2 mode desirable
Router(config-if)# shutdown
Router(config-if)# no shutdown
Router(config-if)# end
```

Note See the "Configuring a Range of Interfaces" section on page 8-4 for information about the **range** keyword.

This example shows how to verify the configuration of port channel interface 2:

```
Router# show running-config interface port-channel 2
Building configuration...

Current configuration:
!
interface Port-channel2
 no ip address
 switchport
 switchport access vlan 10
 switchport mode access
end
Router#
```

This example shows how to verify the configuration of Fast Ethernet port 5/6:

```
Router# show running-config interface fastethernet 5/6
Building configuration...

Current configuration:
!
interface FastEthernet5/6
 no ip address
 switchport
 switchport access vlan 10
 switchport mode access
 channel-group 2 mode desirable
end
Router# show interfaces fastethernet 5/6 etherchannel
Port state    = Down Not-in-Bndl
Channel group = 12          Mode = Desirable-Sl     Gcchange = 0
Port-channel  = null        GC   = 0x00000000          Pseudo port-channel = Po1
2
Port index    = 0           Load = 0x00        Protocol =   PAgP

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        P - Device learns on physical port.
        d - PAgP is down.
```

```
Timers: H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.    I - Interface timer is running.

Local information:
                                Hello    Partner  PAgP     Learning  Group
Port      Flags State   Timers  Interval Count   Priority   Method  Ifindex
Fa5/2     d     U1/S1           1s       0        128        Any      0

Age of the port in the current state: 04d:18h:57m:19s
```

This example shows how to verify the configuration of port channel interface 2 after the LAN ports have been configured:

```
Router# show etherchannel 12 port-channel
                Port-channels in the group:
                ----------------------

Port-channel: Po12
------------

Age of the Port-channel   = 04d:18h:58m:50s
Logical slot/port   = 14/1          Number of ports = 0
GC                  = 0x00000000      HotStandBy port = null
Port state          = Port-channel Ag-Not-Inuse
Protocol            =   PAgP

Router#
```

Configuring the LACP System Priority and System ID

The LACP system ID is the combination of the LACP system priority value and the MAC address of the switch.

To configure the LACP system priority and system ID, perform this task:

	Command	Purpose
Step 1	Router(config)# **lacp system-priority** *priority_value*	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show lacp sys-id**	Verifies the configuration.

This example shows how to configure the LACP system priority:

```
Router# configure terminal
Router(config)# lacp system-priority 23456
Router(config)# end
Router(config)#
```

This example shows how to verify the configuration:

```
Router# show lacp sys-id
23456,0050.3e8d.6400
Router#
```

The system priority is displayed first, followed by the MAC address of the switch.

Configuring EtherChannel Load Balancing

To configure EtherChannel load balancing, perform this task:

	Command	Purpose
Step 1	Router(config)# **port-channel per-module load-balance**	(Optional) Enables the ability to specify the load-balancing method on a per-module basis.
Step 2	Router(config)# **port-channel load-balance** {**src-mac** | **dst-mac** | **src-dst-mac** | **src-ip** | **dst-ip** | **src-dst-ip** | **src-port** | **dst-port** | **src-dst-port**} [**module** *slot*]	Configures the EtherChannel load-balancing method. The method is globally applied to all port channels. Optionally, you can configure the load-balancing method for a specific module. The default method is src-dst-ip.
	Router(config)# **no port-channel load-balance**	Reverts to default EtherChannel load balancing.
Step 3	Router(config)# **end**	Exits configuration mode.
Step 4	Router# **show etherchannel load-balance**	Verifies the configuration.

The load-balancing method keywords indicate the following information:

- **dst-ip**—Destination IP addresses
- **dst-mac**—Destination MAC addresses
- **dst-port**—Destination Layer 4 port
- **mpls**—Load balancing for MPLS packets
- **src-dst-ip**—(Default) Source and destination IP addresses
- **src-dst-mac**—Source and destination MAC addresses
- **src-dst-port**—Source and destination Layer 4 port
- **src-ip**—Source IP addresses
- **src-mac**—Source MAC addresses
- **src-port**—Source Layer 4 port

The optional **module** keyword allows you to specify the load-balancing method for a specific module. This capability is supported only on DFC-equipped switching modules. You must enable per-module load balancing globally before configuring the feature on a module.

This example shows how to configure EtherChannel to use source and destination IP addresses:

```
Router# configure terminal
Router(config)# port-channel load-balance src-dst-ip
Router(config)# end
Router(config)#
```

This example shows how to verify the configuration:

```
Router# show etherchannel load-balance

EtherChannel Load-Balancing Configuration:
        src-dst-ip enhanced
        mpls label-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address
```

```
  MPLS: Label or IP

Router#
```

Note In this example, the **enhanced** keyword indicates that PFC3C/CXL hardware is installed and, as a result, VLAN information will also be included in the load-balancing method.

Configuring EtherChannel Hash-Distribution Algorithm

Releases earlier than 12.2(33)SXH support a load-distribution algorithm called the fixed algorithm. When you add a port to the EtherChannel or delete a port from the EtherChannel, the switch updates the port ASIC for each port in the EtherChannel, which causes a short outage on each port.

Release 12.2(33)SXH and later releases support an additional algorithm called the adaptive algorithm. The adaptive algorithm does not need to update the port ASIC for existing member ports.

The fixed algorithm is the default algorithm. You can configure a global value for the adaptive algorithm. You can also specify the algorithm for individual port channels.

When you change the algorithm, the change is applied at the next member link event (link down, link up, addition, deletion, no shutdown, and shutdown). When you enter the command to change the algorithm, the command console issues a warning that the command does not take effect until the next member link event.

Note Some external devices require the fixed algorithm. For example, the service control engine (SCE) requires incoming and outgoing packets to use the same port.

Note If you change the load-balancing method, EtherChannel ports on DFC-equipped switching modules or on an active supervisor engine in a dual supervisor engine configuration will flap.

Configuring the Hash-Distribution Algorithm Globally

To configure the load-sharing algorithm globally, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`port-channel hash-distribution {adaptive	fixed}`**	Sets the hash distribution algorithm to adaptive or fixed.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.

This example shows how to globally set the hash distribution to adaptive:

```
Router(config)# port-channel hash-distribution adaptive
```

Configuring the Hash-Distribution Algorithm for a Port Channel

To configure the hash-distribution algorithm for a specific port channel, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface port-channel** *channel-num*	Enters interface configuration mode for the port channel.
Step 2	Router(config-if)# **port-channel port hash-distribution** {**adaptive** | **fixed**}	Sets the hash distribution algorithm for this interface
Step 3	Router(config-if)# **end**	Exits interface configuration mode.

This example shows how to set the hash distribution algorithm to adaptive on port channel 10:

```
Router (config)# interface port-channel 10
Router (config-if)# port-channel port hash-distribution adaptive
```

Configuring the EtherChannel Min-Links Feature

The EtherChannel min-links feature is supported on LACP EtherChannels. This feature allows you to configure the minimum number of member ports that must be in the link-up state and bundled in the EtherChannel for the port channel interface to transition to the link-up state. You can use the EtherChannel min-links feature to prevent low-bandwidth LACP EtherChannels from becoming active. This feature also causes LACP EtherChannels to become inactive if they have too few active member ports to supply your required minimum bandwidth. In addition, when LACP max-bundle values are specified in conjunction with min-links, the configuration is verified and an error message is returned if the min-links value is not compatible with (equal to or less than) the max-bundle value.

To configure the EtherChannel min-links feature, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface port-channel** *group_number*	Selects an LACP port channel interface.
Step 2	Router(config-if)# **port-channel min-links** *number*	Configures the minimum number of member ports that must be in the link-up state and bundled in the EtherChannel for the port channel interface to transition to the link-up state.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show running-config interface port-channel** *group_number* Router# **show interfaces** *type*[1] *slot/port* **etherchannel**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note Although the EtherChannel min-links feature works correctly when configured only on one end of an EtherChannel, for best results, configure the same number of minimum links on both ends of the EtherChannel.

This example shows how to configure port channel interface 1 to be inactive if fewer than two member ports are active in the EtherChannel:

```
Router# configure terminal
Router(config)# interface port-channel 1
Router(config-if)# port-channel min-links 2
Router(config-if)# end
```

Configuring LACP 1:1 Redundancy

For the LACP 1:1 redundancy feature, the LACP EtherChannel must contain exactly two links, of which only one is active. The link with the lower port priority number (and therefore a higher priority) will be the active link, and the other link will be in a hot standby state. The LACP max-bundle must be set to 1.

To configure the LACP 1:1 redundancy feature, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface port-channel group_number`	Selects an LACP port channel interface.
Step 1	`Router(config-if)# lacp fast-switchover`	Enables the fast switchover feature for this EtherChannel.
Step 2	`Router(config-if)# lacp max-bundle 1`	Sets the maximum number of active member ports to be one.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show running-config interface port-channel group_number` `Router# show interfaces type[1] slot/port etherchannel`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note For the LACP 1:1 redundancy feature to work correctly, especially the fast switchover capability, the feature needs to be enabled at both ends of the EtherChannel.

This example shows how to configure an LACP EtherChannel with 1:1 redundancy. Because Fast Ethernet port 5/6 is configured with a higher port priority number (and therefore a lower priority) than the default of 32768, it will be the standby port.

```
Router# configure terminal
Router(config)# lacp system-priority 33000
Router(config)# interface range fastethernet 5/6 -7
Router(config-if)# channel-protocol lacp
Router(config-if)# channel-group 1 mode active
Router(config)# interface fastethernet 5/6
Router(config-if)# lacp port-priority 33000
Router(config)# interface port-channel 1
Router(config-if)# lacp fast-switchover
Router(config-if)# lacp max-bundle 1
Router(config-if)# end
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Configuring Auto Interleaved Port Priority For LACP Port Channels

To configure auto interleaved port priority for LACP on a port channel, perform this task on the port channel interface:

	Command	Purpose
Step 1	`Router(config)#` **`interface port-channel`** *`channel-group`*	Selects a port channel interface to configure.
Step 2	`Router(config-if)#` **`lacp active-port distribution automatic`**	Configures the port channel to use interleaved port priority. **Note** You need to perform a shutdown and no shutdown for interleaved port priority to be enabled.
Step 3	`Router(config-if)#` **`shutdown`**	Disables the interface.
Step 4	`Router(config-if)#` **`no shutdown`**	Enables the interface.
Step 5	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 6	`Router#` **`show etherchannel`** *`channel-group`* **`port-channel`** `Router#` **`show etherchannel`** *`channel-group`* **`summary`**	Verifies the configuration.

This example shows how to configure auto interleaved port priority on a port channel:

```
Router(config)# interface port-channel23
Router(config-if)# lacp active-port distribution automatic
Please shut/no shut the port-channel for configuration to take effect immediately.
Router(config-if)# shutdown
Router(config-if)# no shutdown

Router(config-if)# end
```

This example shows how to verify the configuration of port channel interface 23:

```
Router# show running interfaces port-channel23
Building configuration...

Current configuration : 81 bytes
!
interface Port-channel23
 no switchport
 no ip address
 lacp max-bundle 4
 lacp active-port distribution automatic
end
```

This example shows how to verify the configuration of EtherChannel 23:

```
Router# show etherchannel 23 summary

Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator
```

```
        M - not in use, no aggregation due to minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        d - default port

        w - waiting to be aggregated
Number of channel-groups in use: 9
Number of aggregators:           9

Group  Port-channel  Protocol    Ports
------+-------------+-----------+---------------------------------------
23     Po23(RU)        LACP      Gi1/1/21(P)    Gi1/1/22(P)    Gi1/1/23(H)
                                 Gi1/1/24(H)    Gi2/1/17(P)    Gi2/1/18(P)
                                 Gi2/1/19(H)    Gi2/1/20(H)

Last applied Hash Distribution Algorithm: Fixed
```

Note The above example shows that the four bundled ports are distributed 2 per chassis and slot.

Configuring LACP Port-Channel Standalone Disable

To disable the standalone EtherChannel member port state on a port channel (see Table 19-2 on page 19-3), perform this task on the port channel interface:

	Command	Purpose
Step 1	`Router(config)#` **`interface port-channel`** *`channel-group`*	Selects a port channel interface to configure.
Step 2	`Router(config-if)#` **`port-channel standalone-disable`**	Disables the standalone mode on the port-channel interface.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show etherchannel`** *`channel-group`* **`port-channel`** `Router#` **`show etherchannel`** *`channel-group`* **`detail`**	Verifies the configuration.

This example shows how to disable the standalone EtherChannel member port state on port channel 42:

```
Router(config)# interface port-channel channel-group
Router(config-if)# port-channel standalone-disable
```

This example shows how to verify the configuration:

```
Router# show etherchannel 42 port-channel | include Standalone
Standalone Disable  = enabled
Router# show etherchannel 42 detail | include Standalone
Standalone Disable  = enabled
```

Configuring mLACP for Server Access

This chapter describes how to configure the multichassis LACP (mLACP) for server access feature. Release 12.2(33)SXJ and later releases support the mLACP for server access feature. This chapter includes these sections:

Note

- For information about Etherchannels, see Chapter 19, "Configuring EtherChannels."
- For information about the IEEE 802.3ad link aggregation control protocol (LACP), see the "Understanding IEEE 802.3ad LACP EtherChannel Configuration" section on page 19-4.

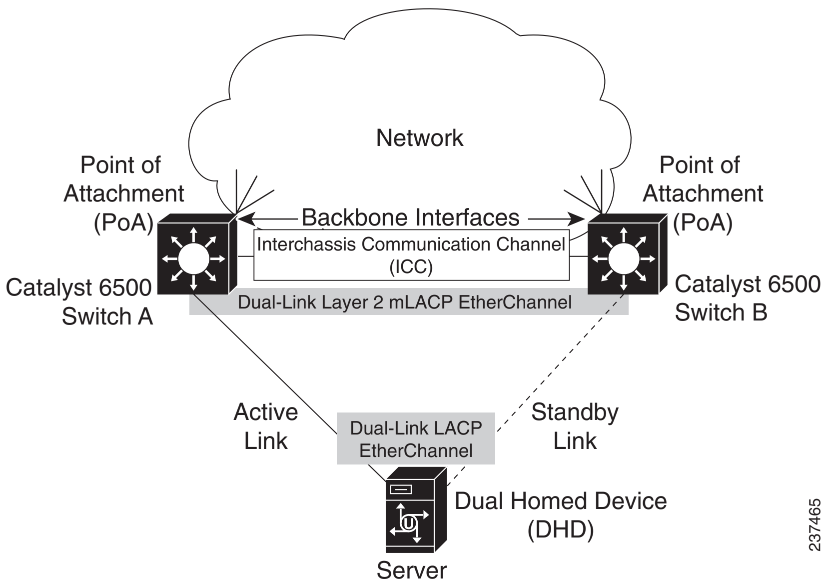
Understanding mLACP for Server Access

These sections describe mLACP for server access:

Overview of mLACP for Server Access

The mLACP for server access feature supports single redundant Layer 2 LACP links from a pair of Catalyst 6500 switches to a server. Figure 20-1 illustrates switches connected to a server.

Figure 20-1 mLACP for Server Access Topology



In the mLACP for server access topology, the server is a dual-homed device connected through two Layer 2 links. On the server, the links attach to ports that are configured as members of a Layer 2 LACP EtherChannel. The links function in active-standby redundancy mode, with only one link active at any time, which prevents Layer 2 loops and imposes no Spanning Tree Protocol (STP) requirements.

Each link connects to a switch that functions as a point-of-attachment (PoA). On the switches, the links attach to ports that are configured as members of a Layer 2 Multichassis EtherChannel (MCEC). An MCEC uses the Interchassis Communication Protocol (ICCP) over the interchassis communication channel (ICC) to synchronize states between the PoAs.

Note
- A switch configured as a PoA cannot form mLACP peer relationships with more than one other switch.
- The switch-to-server connection is a connection between an mLACP port-channel interface on the PoAs and an LACP port-channel interface on the server.

Understanding mLACP Operation

These sections describe mLACP operation:

- mLACP for Server Access Feature Components, page 20-3
- mLACP System ID, page 20-5
- mLACP Port Identifier, page 20-5
- Port-Channel ID, page 20-5

mLACP for Server Access Feature Components

This table describes the feature components that must be configured compatibly on the PoAs.

<table>
<tr><th>Active PoA (Switch A)</th><th>Standby PoA (Switch B)</th></tr>
<tr><td colspan="2">Component:
Interchassis communication channel (ICC)
Definition:
A link that connects the PoAs to carry only ICCP traffic.
Note:
The ICC supports only Interchassis Communication Protocol (ICCP) traffic.</td></tr>
<tr><td>On switch A, the ip address interface command configures an IP address on the switch A end of the ICC link.</td><td>On switch B, the ip address interface command configures an IP address on the switch B end of the ICC link.</td></tr>
<tr><td colspan="2">Component:
mLACP Redundancy Group
Definition:
Two PoAs that form a virtual LACP peer.
Note:
• One mLACP redundancy group can support multiple mLACP port-channel interfaces.
• An mLACP redundancy group has the same mLACP interchassis group ID on both PoAs.
• The mLACP interchassis group ID configured on port-channel interfaces configures them as members of the mLACP redundancy group.</td></tr>
<tr><td>On switch A, the member ip redundancy group command points to the ICC-link IP address on switch B.</td><td>On switch B, the member ip redundancy group command points to the ICC-link IP address on switch A.</td></tr>
<tr><td>On switch A, the mlacp system-priority redundancy group command configures the switch A mLACP system priority value used on this PoA in this redundancy group as part of the mLACP system ID value.</td><td>On switch B, the mlacp system-priority redundancy group command configures the switch B mLACP system priority value used on this PoA in this redundancy group as part of the mLACP system ID value.</td></tr>
<tr><td>On switch A, the mlacp system-mac redundancy group command configures the switch A mLACP system priority value used on this PoA in this redundancy group as part of the mLACP system ID value.</td><td>On switch B, the mlacp system-mac redundancy group command configures the switch B mLACP system priority value used on this PoA in this redundancy group as part of the mLACP system ID value.</td></tr>
<tr><td>On switch A, the mlacp node-id redundancy group command configures the switch A mLACP port number value used on this PoA in this redundancy group as part of the mLACP port identifier value.</td><td>On switch B, the mlacp node-id redundancy group command configures the switch B mLACP port number value used on this PoA in this redundancy group as part of the mLACP port identifier value.</td></tr>
<tr><td>On switch A, the backbone interface redundancy group command configures mLACP link-status monitoring on the switch A physical ports that carry server traffic to and from the network.</td><td>On switch B, the backbone interface redundancy group command configures mLACP link-status monitoring on the switch B physical ports that carry server traffic to and from the network.</td></tr>
</table>

Active PoA (Switch A)	Standby PoA (Switch B)
Component: Port-channel interfaces **Definition:** Port-channel interface commands that configure DHD connection links. **Note:** • Each port-channel interface supports one Layer 2 link to a server. • Port-channel interfaces must be configured with matching **interface port-channel** port-channel interface command ID values on both PoAs. • Use the mLACP interchassis group ID as the **mlacp interchassis group** port-channel interface command ID value on both PoAs.	
On switch A, the **mlacp lag-priority** port-channel interface command configures the switch A mLACP port priority value used on this PoA in this redundancy group as part of the mLACP port identifier value.	On switch B, the **mlacp lag-priority** port-channel interface command configures the switch B mLACP port priority value used on this PoA in this redundancy group as part of the mLACP port identifier value.
Component: Port interfaces **Definition:** Interface commands that configure Layer 2 DHD connection links as members of the mLACP port-channel interfaces. **Note:** The channel-group interface command configured with the mLACP port-channel ID value makes the port a member of an mLACP port-channel interface.	

mLACP System ID

In each mLACP redundancy group, the PoA with the lowest mLACP system ID value is the link selection PoA. The link selection PoA controls selection of the link that will be active. Comparisons of mLACP system IDs are numeric comparisons of the unsigned integer values of the mLACP system ID values.

mLACP uses the two-byte mLACP system priority value as the most significant two octets and the configured mLACP system MAC address value as the least significant octets of the 8-byte mLACP system ID value.

Note The mLACP for server access feature does not support DHD control of active link selection. Configure the LACP instance on the DHD to have a numerically higher LACP system ID value than the PoA mLACP system ID values.

mLACP System Priority

The **mlacp system-priority** redundancy group command sets the mLACP system priority value. A lower value contributes to selection of the PoA as the link selection PoA.

mLACP System MAC Address

The **mlacp system-mac** redundancy group command sets the mLACP system MAC address. A lower value contributes to selection of the PoA as the link selection PoA.

Note The mLACP system MAC value is only used in the LACP PDUs sent between the PoAs and DHD.

mLACP Port Identifier

In each redundancy group, for each port-channel interface in the redundancy group, the link selection PoA selects the link with the lowest mLACP port identifier value to be active. Comparisons of mLACP port identifiers are numeric comparisons of the unsigned integer values of the mLACP port identifier values.

mLACP uses the two-byte mLACP port priority value as the most significant two octets and the configured mLACP port number as the least significant two octets of the four-byte mLACP port identifier value.

mLACP Port Priority

The **mlacp lag-priority** port-channel interface command configures the mLACP port priority value. A lower value contributes to selection of a link to be active.

mLACP Port Number

The **mlacp node-id** redundancy group command configures the mLACP port number. A lower value contributes to selection of a link to be active.

Port-Channel ID

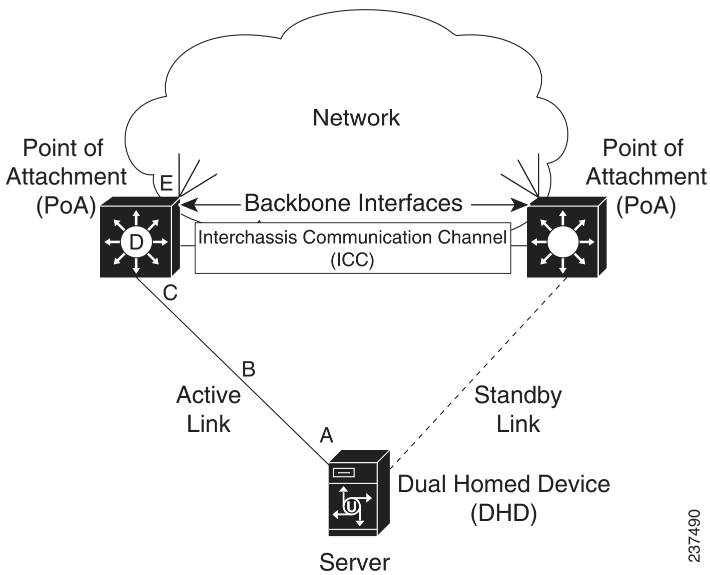
You create the port-channel ID with the **port-channel** *id_number* command and associate physical ports to it with the **channel-group** *id_number* interface configuration mode command. The port-channel ID on the two PoAs in an mLACP redundancy group must match. The port-channel ID on the DHD can be different from the value configured on the PoAs.

Failure Protection Scenarios

The mLACP for server access feature provides network resiliency by protecting against port, link, and PoA failures. These failures can be categorized into five types. Figure 20-2 shows the failure points in a network, denoted by the letters A through E.

- A—Failure of the port on the server
- B—Failure of the server-connection link
- C—Failure of the server-connection port on the active PoA
- D—Failure of the active PoA
- E—Failure of the backbone interfaces

Figure 20-2 mLACP for Server Access Protected Failure Points



When any of these faults occur, mLACP triggers a switchover from the active PoA to the standby PoA.

mLACP Failover

These sections describe mLACP failover:

- Overview, page 20-7
- Dynamic Port Priority, page 20-7
- Revertive and Nonrevertive Modes, page 20-7
- Peer Monitoring with Interchassis Redundancy Manager, page 20-7

Overview

mLACP forces failover in these situations:

- If the active PoA loses communication with the server (failure points A, B, or C) or if all backbone interfaces on the active PoA fail (failure point E), mLACP fails over to the link on the standby PoA. (PoA failover does not occur.)
- If ICRM notifies the standby PoA that the active PoA has failed, the standby PoA becomes active.

Note The DHD does not participate in failover determination.

Dynamic Port Priority

The default failover method uses dynamic port priority changes on the local member links to force the LACP selection logic to move the required standby link to the SELECTED and Collecting_Distributing state. This state change occurs when the LACP actor port priority values for all affected member links on the currently active PoA are changed to a higher numeric value than the standby PoA's port priority (which gives the standby PoA ports a higher claim to bundle links). Changing the actor port priority triggers the transmission of an mLACP Port Config Type-Length-Value (TLV) message to all peers in the redundancy group. These messages also serve as notification to the standby PoA that the currently active PoA is attempting to relinquish its role. The LACP then transitions the standby link to the SELECTED state and moves all the currently active links to STANDBY.

Dynamic port priority changes are not automatically written back to the running configuration or to the NVRAM configuration. If you want the system to use the current priorities when the it reloads, configure the **mlacp lag-priority** command and save the configuration.

Revertive and Nonrevertive Modes

The mLACP feature uses the dynamic port priority functionality for both revertive mode and nonrevertive mode. The default operation is revertive, which is the default in single chassis LACP. Nonrevertive mode can be enabled on a per port-channel basis by using the **lacp failover non-revertive** command in interface configuration mode, which is supported only for mLACP.

Nonrevertive mode limits failover and possible traffic loss. Dynamic port priority changes are utilized to ensure that the newly activated PoA remains active after the failed PoA recovers.

Revertive mode operation forces the configured primary PoA to return to active state after it recovers from a failure. Dynamic port priority changes are uses when necessary to allow the recovering PoA to resume its active role.

Peer Monitoring with Interchassis Redundancy Manager

The interchassis redundancy manager (ICRM) can monitor a peer with these methods:

- Routewatch (RW)—This method is the default.
- Bidirectional Forwarding Detection (BFD)—You must configure the redundancy group with the **monitor peer bfd** command.

Note For stateful switchover (SSO) deployments with redundant supervisor engines, configure BFD monitoring and a static route for the ICCP connection to prevent both PoAs from being active after an SSO failover. Routewatch does not support SSO.

There is a monitoring adjacency for each peer (designated by a member IP address) in each redundancy group. If there are two peers with the same IP address, the adjacency is shared regardless of the monitoring mode. For example, if redundancy groups 1 and 2 have a peer relationship with member IP address 10.10.10.10, there is only one adjacency to 10.10.10.10, which is shared in both redundancy groups. Redundancy groups can use different monitoring methods.

Note BFD is completely dependent on routewatch; there must be a route to the peer for ICRM to initiate BFD monitoring. BFD implies routewatch. Sometimes the status of the adjacency might seem misleading but is accurately representing the state. Also, if the route to the peer PoA is not through the directly connected (back-to-back) link between the systems, BFD can give misleading results.

This is an example of output from the **show redundancy interface** command:

```
Router# show redundancy interface

Redundancy Group 1 (0x1)
  Applications connected: mLACP
  Monitor mode: Route-watch
  member ip: 201.0.0.1 'mlacp-201', CONNECTED
    Route-watch for 201.0.0.1 is UP
    mLACP state: CONNECTED

ICRM fast-failure detection neighbor table
  IP Address       Status Type Next-hop IP      Interface
  ==========       ====== ==== ===========      =========
  201.0.0.1        UP     RW
```

Table 20-1 explains the adjacency status displayed by the **show redundancy interchassis** command.

Table 20-1 Status Information from the show redundancy interchassis command

Adjacency Type	Adjacency Status	Meaning
RW	DOWN	Routewatch or BFD is configured, but there is no route for the given IP address.
RW	UP	Routewatch or BFD is configured. Routewatch is up, which indicates that there is a valid route to the peer. If BFD is configured and the adjacency status is UP, BFD is probably not configured on the interface of the route's adjacency.
BFD	DOWN	BFD is configured. A route exists and the route's adjacency is to an interface that has BFD enabled. BFD is started but the peer is down. The DOWN status can be because the peer is not present or BFD is not configured on the peer's interface.
BFD	UP	BFD is configured and operational.
Note If the adjacency type is BFD, routewatch is UP regardless of the BFD status.		

mLACP for Server Access Guidelines and Restrictions

When configuring mLACP for Server Access, follow these guidelines and restrictions:

- PFC3A mode does not support the mLACP for server access feature.
- VSS mode does not support the mLACP for server access feature.
- No more than 100 VLANs can be active on a switch configured as a PoA.
- Switches configured with the mLACP for server access feature cannot support the Wireless Services Module (WiSM; WS-SVC-WISM-1-K9) or Wireless Services Module 2 (WiSM2; WS-SVC-WISM2-K9). Do not install WiSM modules in switches configured with the mLACP for server access feature. Do not configure the mLACP for server access feature in switches where any WiSM modules are installed. (CSCtn90999)
- The mLACP for server access feature supports the following:
  - Pairs of Catalyst 6500 switches with Supervisor Engine 720 or with Supervisor Engine 720-10GE configured as points of attachment (PoAs).

    Note A switch configured as a PoA cannot form an mLACP peer relationship with more than one other switch.

  - Servers with fully compliant IEEE 802.3ad LACP support, configured as dual-homed devices (DHDs).

    Note The CLI does not enforce this restriction, but servers that support IEEE 802.3ad LACP are the only tested and supported DHDs.

  - One Layer 2 access link from each PoA to each DHD.

    Note The CLI does not enforce this restriction, but one Layer 2 access link from each PoA to each DHD is the only tested and supported configuration.

  - mLACP Layer 2 port-channel interfaces on a pair of switches with one Layer 2 access port per mLACP port-channel interface on each PoA.

    Note The CLI does not enforce this restriction, but one Layer 2 access port per mLACP port-channel interface on each PoA is the only tested and supported configuration.

- The mLACP for server access feature has an mLACP extended mode.
  - The mLACP extended mode is disabled by default.
  - A reload is required to enable the mLACP extended mode after you enter the **port-channel mode mlacp-extended** command.
  - When the mLACP extended mode is not enabled, the switch supports 128 PaGP, LACP, or mLACP port-channel interfaces, numbered between 1 and 256. These port channel interfaces support QoS and ACLs.

- When the mLACP extended mode is enabled, the switch supports the following:
  - —128 PaGP, LACP, or mLACP port-channel interfaces, numbered 1 through 256. These port-channel interfaces support QoS and ACLs.
  - —An additional 128 mLACP port-channel interfaces, numbered 257 through 512. These port-channel interfaces do not support QoS and ACLs.

- Configure PoA network access so that each PoA can fully support all of the server traffic. Do not include one PoA in the network access path of the other PoA.
- Ensure that there is no server traffic on any links between the PoAs.
- Configure the Interchassis Communication Channel (ICC) as a point-to-point connection between the PoAs.
- Configure the ICC so that it carries only ICCP traffic.

Note Traffic volume on the ICC will be relatively low.

- mLACP operation is only supported when the ICC is functioning correctly. If possible, configure the ICC as a redundant connection. For example, you can configure the ICC as a two-link EtherChannel if there is available port-channel capacity.
- mLACP does not support half-duplex links.
- mLACP does not support multiple neighbors.
- Converting a port channel to mLACP can cause a service disruption.
- The DHD system priority must be lower (higher numerically) than the PoA system priority.

Configuring mLACP for Server Access

These sections describe how to configure mLACP for server access:

Summary of mLACP PoA Configuration Values

Table 20-2 provides a list of the values that need to be coordinated on the PoA switches.

Table 20-2 Coordinated mLACP Configuration Values

Active PoA (Switch A)	Standby PoA (Switch B)
interface *type slot/port* description connected to switch **B** ip address 10.0.0.**1** 255.255.255.255	interface *type slot/port* description connected to switch **A** ip address 10.0.0.**2** 255.255.255.255
ip route 200.0.0.1 255.255.255.255 *icc_port_A*	ip route 100.0.0.1 255.255.255.255 *icc_port_B*
interface loopback 0 description Supports routing to switch B ip address 100.0.0.1 255.255.255.255	interface loopback 0 description Supports routing to switch A ip address 200.0.0.1 255.255.255.255
mpls ldp router-id loopback 0 force	mpls ldp router-id loopback 0 force
redundancy interchassis group **10** member ip 10.0.0.**2** mlacp system-priority **1** mlacp system-mac **0001.0001.0001** mlacp node-id **1**	redundancy interchassis group **10** member ip 10.0.0.**1** mlacp system-priority **2** mlacp system-mac **0002.0002.0002** mlacp node-id **2**
interface port-channel **50** mlacp interchassis group **10** mlacp lag-priority **1**	interface port-channel **50** mlacp interchassis group **10** mlacp lag-priority **2**
interface *type slot/port* description connected to DHD channel-group **50** mode passive	interface type *slot/port* description connected to DHD channel-group **50** mode passive

Note This summary section does not list all of the commands required to configure the mLACP for server access feature. See the following sections for complete configuration procedures.

Configuring mLACP Global Options

To configure mLACP global options, perform this task:

	Command or Action	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **port-channel mode mlacp-extended**	(Optional) Enables the mLACP extended mode, which supports configuration of an additional 128 mLACP port-channel interfaces. **Note** A reload is required to enable the mLACP extended mode after you enter the **port-channel mode mlacp-extended** command.

Step 4	`Router(config)# errdisable recovery cause mlacp-minlink`	Enables automatic recovery from a failover state of the port channel.
Step 5	`Router(config)# end`	Returns to privileged EXEC mode.

Configuring the Interchassis Communication Channel

These sections describe how to configure the Interchassis Communication Channel (ICC):

- Configuring the ICC Port, page 20-12
- Configuring ICCP Routing, page 20-13

Configuring the ICC Port

To configure the ICC port, perform this task:

	Command or Action	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config-if)# **interface** *type*[1] *slot/port*	Selects the ICC port.
Step 4	Router(config-if)# **description** *peer_description*	Describes the ICC port that connects to the other PoA.
Step 5	Router(config-if)# **ip address** *ip_address mask*	Configures an IP address on the ICC port. **Note** This address is used as the member IP address on the other PoA.
Step 6	Router(config-if)# **mpls ip**	Enables MPLS on the interface.
Step 7	Router(config-if)# **bfd interval 600 min_rx 600 multiplier 6**	Configures BFD to support Interchassis Redundancy Manager (ICRM) traffic.
Step 8	Router(config-if)# **shutdown**	Disables the interface.
Step 9	Router(config-if)# **no shutdown**	Enables the interface.
Step 10	Router(config-if)# **exit**	Exits interface configuration mode.
Step 11	Router(config)# **end**	Returns to privileged EXEC mode.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure the port that connects to the other PoA switch:

```
Router> enable
Router# configure terminal
Router(config-if)# interface fastethernet 1/1
Router(config-if)# description Connected to switch B
Router(config-if)# ip address 10.0.0.1 255.255.255.255
Router(config-if)# bfd interval 600 min_rx 600 multiplier 6
Router(config-if)# shutdown
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# end
```

Configuring ICCP Routing

To configure ICCP routing, perform this task:

	Command or Action	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`ip route`** *`ip_address mask icc_port`*	Configures a static route that points to the IP address of the loopback interface on the other PoA through the ICC port.
Step 4	`Router(config)#` **`interface`** *`loopback_interface`*	Configures a loopback interface to support ICCP routing. (You can use an existing loopback interface.)
Step 5	`Router(config-if)#` **`description`** *`loopback_description`*	Describes the loopback interface.
Step 6	`Router(config-if)#` **`ip address`** *`ip_address mask`*	Configures an IP address on the ICC port.
Step 7	`Router(config-if)#` **`exit`**	Exits interface configuration mode.
Step 8	`Router(config)#` **`mpls ldp router-id`** *`loopback_interface`* **`force`**	Configures MPLS LDP to use the loopback interface created in Step 4.
Step 9	`Router(config)#` **`end`**	Returns to privileged EXEC mode.

This example shows how to configure the port that connects to the other PoA switch:

```
Router> enable
Router# configure terminal
Router(config)# ip route 200.0.0.2 255.255.255.255 fastethernet 1/1
Router(config-if)# interface loopback 0
Router(config-if)# description Supports routing to switch B
Router(config-if)# ip address 100.0.0.1 255.255.255.255
Router(config-if)# exit
Router(config)# mpls ldp router-id 0 force
Router(config)# end
```

Configuring Interchassis Redundancy Groups

These sections describe how to configure interchassis redundancy groups:

- Configuring an Interchassis Redundancy Group, page 20-14
- Configuring an mLACP Port-Channel Interface, page 20-15
- Configuring the mLACP Member Port, page 20-17

Configuring an Interchassis Redundancy Group

To configure an interchassis redundancy group, perform this task:

	Command or Action	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`redundancy`**	Enters redundancy configuration mode.
Step 4	`Router(config-red)#` **`interchassis group`** *`group_id`*	Creates an interchassis group and enters interchassis redundancy mode. **Note** Use the same interchassis group ID on the other PoA for the other link in the redundancy group.
Step 5	`Router(config-r-ic)#` **`member ip`** *`ip_address`*	Configures the IP address of the mLACP peer member group. Use the IP address configured on ICC port on the other PoA ("Configuring the Interchassis Communication Channel," Step 5).
Step 6	`Router(config-r-ic)#` **`mlacp system-mac`** `{`**`0001.0001.0001`** `|` **`0002.0002.0002`**`}`	Defines the mLACP system MAC address value that is part of the mLACP system ID value that selects the PoA that selects the active link. • Configure the active PoA with 0001.0001.0001. • Configure the standby PoA with 0002.0002.0002.
Step 7	`Router(config-r-ic)#` **`mlacp system-priority`** `{`**`1`** `|` **`2`**`}`	Defines the mLACP system priority value that is part of the mLACP system ID value that selects the PoA that selects the active link. • Configure the active PoA with priority 1. • Configure the standby PoA with priority 2.
Step 8	`Router(config-r-ic)#` **`mlacp node-id`** `{`**`1`** `|` **`2`**`}`	Defines the mLACP port number value used as part of the mLACP port identifier value that is used to select the active link. • Configure node ID 1 on the PoA that will have the active link. • Configure node ID 2 on the PoA that will have the standby link.
Step 9	`Router(config-r-ic)#` **`backbone interface`** *`type slot/port`*	Configures mLACP link-status monitoring on the physical ports that carry server traffic to and from the network. **Note** Enter a **backbone interface** command for each port that carries server traffic between the PoA and the network.
Step 10	`Router(config-r-ic)#` **`monitor peer bfd`**	Configures the BFD option to monitor the state of the peer. The default option is route-watch.

This example shows how to configure an interchassis redundancy group that configures a switch as the active PoA:

```
Router> enable
Router# configure terminal
Router(config)# redundancy
Router(config-red)# interchassis group 10
Router(config-r-ic)# member ip 10.0.0.2
Router(config-r-ic)# mlacp node-id 1
Router(config-r-ic)# mlacp system-mac 0001.0001.0001
Router(config-r-ic)# mlacp system-priority 1
Router(config-r-ic)# monitor peer bfd
```

Configuring an mLACP Port-Channel Interface

To configure an mLACP port-channel interface, perform this task:

	Command or Action	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# interface port-channel port_channel_id`	Configures the port channel and enters interface configuration mode. • The *port_channel_id* value can be from 1 through 256. • In mLACP extended mode, the value can be from 1 through 512. – You can configure 128 PaGP, LACP, or mLACP port-channel interfaces, numbered 1 through 256. These port channel interfaces support QoS and ACLs. – You can configure an additional 128 mLACP port-channel interfaces, numbered 257 through 512. These port channel interfaces do not support QoS and ACLs. • You must configure the same port-channel ID on the other PoA for the port-channel interface of other link in the mLACP redundancy group.
Step 4	`Router(config-if)# switchport`	Configures the port-channel interface for Layer 2 switching.
Step 5	`Router(config-if)# switchport access`	Configures the port-channel interface as an access port.
Step 6	`Router(config-if)# no shutdown`	Enables the interface.
Step 7	`Router(config-if)# mlacp interchassis group group_id`	Associates the port-channel interface with the mLACP redundancy group. Use the *group_id* configured in "Configuring Interchassis Redundancy Groups" section on page 20-13, Step 4.

	Command or Action	Purpose
Step 8	`Router(config-r-ic)#` **`mlacp node-id`** `{`**`1`**`|`**`2`**`}`	Defines the the mLACP port number value used as part of the mLACP port identifier value that is used to select the active link. • Configure the PoA with the active link with priority 1. • Configure the PoA with the standby link with priority 2.
Step 9	`Router(config-if)#` **`lacp max-bundle 1`**	Sets maximum number of active member ports. **Note** The CLI does not enforce 1 as the only value, but it is the only tested and supported value.
Step 10	`Router(config-if)#` **`port-channel min-links 1`**	Sets the minimum number of member ports. **Note** • The CLI does not enforce 1 as the only value, but it is the only tested and supported value. • The other link in the EtherChannel is on the other PoA.
Step 11	`Router(config-if)#` **`lacp failover non-revertive`**	(Optional) Sets the mLACP switchover to nonrevertive. The revertive mode is the default, with a 180-second delay. **Note** Although present in the CLI, the **lacp failover brute-force** command is not supported.

This example shows how to configure an mLACP port-channel interface:

```
Router> enable
Router# configure terminal
Router(config)# interface port-channel 50
Router(config-if)# switchport
Router(config-if)# switchport access
Router(config-if)# no shutdown
Router(config-if)# mlacp lag-priority 1
Router(config-if)# mlacp interchassis group 10
Router(config-if)# lacp max-bundle 1
Router(config-if)# port-channel min-links 1
```

Configuring the mLACP Member Port

To configure the mLACP member port, perform this task:

	Command or Action	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config-if)#` **`interface`** *`type`*[1] *`slot/port`*	Selects a LAN port to configure.
Step 4	`Router(config-if)#` **`no ip address`**	Ensures that there is no IP address assigned to the LAN port.
Step 5	`Router(config-if)#` **`switchport`**	Configures the LAN port for Layer 2 switching.
Step 6	`Router(config-if)#` **`switchport mode access`**	Configures the LAN port as an access port.
Step 7	`Router(config-if)#` **`switchport access vlan`** *`vlan_id`*	Configures the LAN port as a member of a VLAN.
Step 8	`Router(config-if)#` **`channel-protocol lacp`**	Enables the LACP EtherChannel protocol.
Step 9	`Router(config-if)#` **`channel-group`** *`port_channel_id`* **`mode`** `{`**`active`** `|` **`passive`**`}`	Configures the LAN port as a member of an mLACP port-channel interface and specifies the mode. Use the *port_channel_id* value configured on the appropriate mLACP port-channel interface.
Step 10	`Router(config-if)#` **`mlacp lag-priority`** `{`**`1`**`|`**`2`**`}`	Defines the the mLACP port priority value used as part of the mLACP port identifier value that is used to select the active link. • Configure the active link with priority 1. • Configure the standby link with priority 2.
Step 11	`Router(config-if)#` **`shutdown`**	Disables the interface.
Step 12	`Router(config-if)#` **`no shutdown`**	Enables the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure an mLACP member port as the active link:

```
Router> enable
Router# configure terminal
Router(config-if)# interface gigabitethernet 1/1
Router(config-if)# no ip address
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# switchport access vlan 10
Router(config-if)# channel-protocol lacp
Router(config-if)# channel-group 10 mode passive
Router(config-if)# mlacp lag-priority 1
Router(config-if)# no shutdown
```

Forcing a PoA Failover

The **mlacp lag-priority** command also can be used to force a PoA failover in the following two ways:

- Set the active PoA's LAG priority to a value greater than the LAG priority on the standby PoA. This setting results in the quickest failover because it requires the fewest LACP link state transitions on the standby links before they turn active.
- Set the standby PoA's LAG priority to a value numerically less than the LAG priority on the active PoA. This setting results in a slightly longer failover time because standby links have to signal OUT_OF_SYNC to the DHD before the links can be brought up and go active.

In some cases, the operational priority and the configured priority might differ when using dynamic port priority management to force failovers. In this case, the configured version is not changed unless the port channel is operating in nonrevertive mode. Enter the s**how lacp multichassis port-channel** command to view the current operational priorities. Use the **show running-config** command to display the configured priority values.

Troubleshooting mLACP

Use these **debug** commands to troubleshoot mLACP:

	Command or Action	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`debug redundancy interchassis`** `{`**`all`** `|` **`application`** `|` **`error`** `|` **`event`** `|` **`monitor`**`}`	Enables debugging of the interchassis redundancy manager.
Step 3	`Router#` **`debug mpls ldp iccp`**	Enables debugging of the InterChassis Control Protocol (ICCP).
Step 4	`Router#` **`debug lacp`** `[`**`all`** `|` **`event`** `|` **`fsm`** `|` **`misc`** `|` **`multi-chassis`** `[`**`all`** `|` **`database`** `|` **`lacp-mgr`** `|` **`redundancy-group`** `|` **`user-interface`**`] |` **`packet`**`]`	Enables debugging of LACP activity.

Verifying an Active PoA

The following **show** commands can be used to display statistics and configuration parameters to verify the operation of the mLACP feature on an active PoA:

- show lacp multi-chassis group, page 20-19
- show lacp multi-chassis port-channel, page 20-20
- show etherchannel summary, page 20-21
- show etherchannel channel_id port-channel, page 20-21
- show lacp internal, page 20-22
- show lacp neighbor, page 20-22

show lacp multi-chassis group

Use the **show lacp multi-chassis group** command to display the LACP parameters, local configuration, status of the backbone uplink, peer information, node ID, channel, state, priority active, and inactive links.

```
DC35-5# show lacp multi-chassis group 100
Interchassis Redundancy Group 10

        Operational LACP Parameters:
                RG State:     Synchronized
                System-Id:    1.0001.0001.0001
                ICCP Version: 0
        Backbone Uplink Status: Connected
        Local Configuration:
                Node-id:   1
                System-Id: 1.0001.0001.0001

        Peer Information:
                State:        Up
                Node-id:      2
                System-Id:    2.0002.0002.0002
                ICCP Version: 0

State Flags: Active            - A
             Standby           - S
             Down              - D
             AdminDown         - AD
             Standby Reverting - SR
             Unknown           - U

mLACP Channel-groups
Channel    State      Priority     Active Links   Inactive Links
 Group   Local/Peer  Local/Peer     Local/Peer      Local/Peer
   50       A/S         1/2           1/1             0/0
```

show lacp multi-chassis port-channel

Use the **show lacp multi-chassis port-channel** command to display the interface port-channel value channel group, LAG state, priority, inactive links peer configuration, and standby links.

```
DC35-5# show lacp multi-chassis port-channel1
Interface Port-channel50
        Local Configuration:
                Address: 00d0.d32e.d23f
                Channel Group: 50
                State: Active
                LAG State: Up
                Priority: 1
                Inactive Links: 0
                Total Active Links: 1
                           Bundled: 1
                          Selected: 1
                           Standby: 0
                        Unselected: 0

        Peer Configuration:
                Interface: Port-channel50
                Address: 0002.fcbd.cee5
                Channel Group: 50
                State: Standby
                LAG State: Up
                Priority: 2
                Inactive Links: 0
                Total Active Links: 1
                           Bundled: 0
                          Selected: 0
                           Standby: 1
                        Unselected: 0
```

show etherchannel summary

Use the **show etherchannel summary** command to display the status and identity of the mLACP member links.

```
DC35-5# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, no aggregation due to minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        d - default port

        w - waiting to be aggregated
Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
50     Po50(SU)        LACP      Fa1/44(P)
```

show etherchannel *channel_id* port-channel

Use the **show etherchannel** *channel_id* **port-channel** command to display the status and identity of the EtherChannel and and port channel.

```
DC35-5# show etherchannel 50 port-channel
Port-channels in the group:
                ----------------------

Port-channel: Po50    (Primary Aggregator)

------------

Age of the Port-channel   = 0d:01h:15m:10s
Logical slot/port   = 14/5          Number of ports = 1
HotStandBy port = null
Port state          = Port-channel Ag-Inuse
Protocol            =   LACP
Fast-switchover     = disabled


Load share deferral = disabled

Ports in the Port-channel:

Index   Load      Port          EC state       No of bits
------+------+------------+------------------+-----------
 0      FF         Fa1/44   mLACP-active   8

Time since last port bundled:    0d:00h:14m:18s    Fa1/44
Time since last port Un-bundled: 0d:00h:14m:20s    Fa1/44

Last applied Hash Distribution Algorithm: Adaptive
```

show lacp internal

Use the **show lacp internal** command to display the device, port, and member-link information.

```
DC35-5# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 50
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Fa1/44    SA      bndl-act   1             0x32      0x32    0x912D      0x3D

Peer (DC35-6) mLACP member links
```

show lacp neighbor

Use the **show lacp neighbor** command to display the neighbor device, port, and member-link information.

```
DC35-5# show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 50 neighbors

Partner's information:

          Partner Partner   LACP Partner  Partner   Partner  Partner     Partner
Port      Flags   State     Port Priority Admin Key Oper Key Port Number Port State
Fa1/44    SA      bndl-act   32768         0x0       0xAC     0x62D       0x3D
```

Verifying a Standby PoA

The following **show** commands can be used to display statistics and configuration parameters to verify the operation of the mLACP feature on a standby PoA:

- show lacp multi-chassis group, page 20-23
- show lacp multi-chassis portchannel, page 20-24
- show etherchannel summary, page 20-25
- show lacp internal, page 20-26

show lacp multi-chassis group

Use the **show lacp multi-chassis group** command to display the LACP parameters, local configuration, status of the backbone uplink, peer information, node ID, channel, state, priority active, and inactive links.

```
DC35-6# show lacp multi-chassis group
Interchassis Redundancy Group 10

        Operational LACP Parameters:
                RG State:     Synchronized
                System-Id:    1.0001.0001.0001
                ICCP Version: 0
        Backbone Uplink Status: Connected
        Local Configuration:
                Node-id:   2
                System-Id: 2.0002.0002.0002

        Peer Information:
                State:        Up
                Node-id:      1
                System-Id:    1.0001.0001.0001
                ICCP Version: 0

State Flags: Active            - A
             Standby           - S
             Down              - D
             AdminDown         - AD
             Standby Reverting - SR
             Unknown           - U

mLACP Channel-groups
Channel    State      Priority     Active Links   Inactive Links
 Group   Local/Peer  Local/Peer     Local/Peer      Local/Peer
   50       S/A         2/1            1/1             0/0
```

show lacp multi-chassis portchannel

Use the **show lacp multi-chassis portchannel** command to display the interface port-channel value channel group, LAG state, priority, inactive links peer configuration, and standby links.

```
DC35-6# show lacp multi-chassis port-channel
Interface Port-channel50
        Local Configuration:
                Address: 0002.fcbd.cee5
                Channel Group: 50
                State: Standby
                LAG State: Up
                Priority: 2
                Inactive Links: 0
                Total Active Links: 1
                           Bundled: 0
                          Selected: 0
                           Standby: 1
                        Unselected: 0

        Peer Configuration:
                Interface: Port-channel50
                Address: 00d0.d32e.d23f
                Channel Group: 50
                State: Active
                LAG State: Up
                Priority: 1
                Inactive Links: 0
                Total Active Links: 1
                           Bundled: 1
                          Selected: 1
                           Standby: 0
                        Unselected: 0
```

show etherchannel summary

Use the **show etherchannel summary** command to display the status and identity of the mLACP member links.

```
DC35-6# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, no aggregation due to minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        d - default port

        w - waiting to be aggregated
Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
50     Po50(SU)        LACP      Fa3/44(P)
```

show etherchannel *channel_id* port-channel

Use the **show etherchannel** *channel_id* **port-channel** command to display the status and identity of the EtherChannel and port channel.

```
DC35-6# show etherchannel 50 port-channel
                Port-channels in the group:
                ----------------------

Port-channel: Po50    (Primary Aggregator)

------------

Age of the Port-channel   = 0d:01h:17m:40s
Logical slot/port   = 14/5          Number of ports = 1
HotStandBy port = null
Port state          = Port-channel Ag-Inuse
Protocol            =   LACP
Fast-switchover     = disabled


Load share deferral = disabled

Ports in the Port-channel:

Index   Load      Port          EC state       No of bits
------+------+------------+------------------+-----------
 0      FF         Fa3/44   mLACP-stdby  8

Time since last port bundled:    0d:00h:16m:59s    Fa3/44
Time since last port Un-bundled: 0d:00h:17m:00s    Fa3/44

Last applied Hash Distribution Algorithm: Adaptive
```

show lacp internal

Use the **show lacp internal** command to display the device, port, and member-link information.

```
DC35-6# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 50
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Fa3/44    SA      bndl-sby   2             0x32      0x32    0xA32D      0x5

Peer (DC35-5^C^C) mLACP member links
```

show lacp neighbor

Use the **show lacp neighbor** command to display the neighbor device, port, and member-link information.

```
DC35-6# show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 50 neighbors

Partner's information:

          Partner Partner   LACP Partner  Partner   Partner  Partner     Partner
Port      Flags   State     Port Priority Admin Key Oper Key Port Number Port State
Fa3/44    FA      bndl-sby   32768         0x0       0xAC     0x32D       0xF
```

Configuring IEEE 802.1ak MVRP and MRP

This chapter describes how to configure the IEEE 802.1ak Multiple VLAN Registration Protocol (MVRP) and Multiple Registration Protocol (MRP) as implemented in accordance with the IEEE 802.1ak standard.

Note This feature appears in Cisco Feature navigator as "IEEE 802.1ak - MVRP and MRP."

Understanding IEEE 802.1ak MVRP and MRP

These sctions describe IEEE 802.1ak MVRP and MRP:

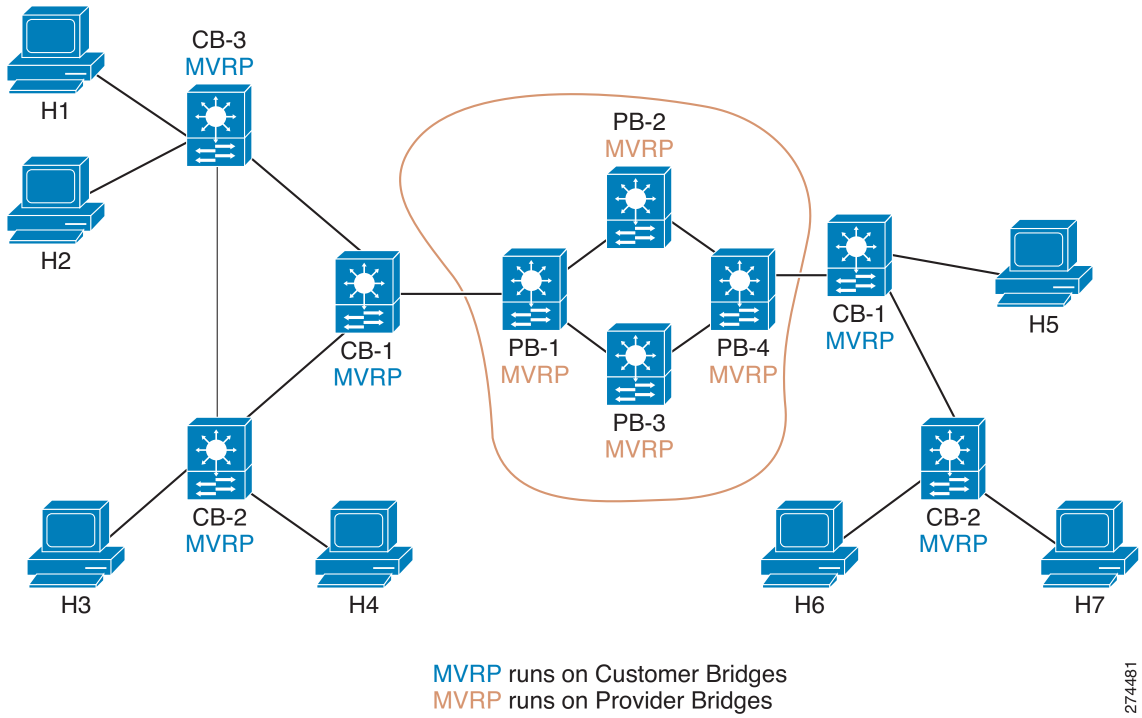
Overview

MVRP supports dynamic registration and deregistration of VLANs on ports in a VLAN bridged network. IEEE 802.1ak uses more efficient Protocol Data Units (PDUs) and protocol design to provide better performance than the Generic VLAN Registration Protocol (GARP) VLAN Registration Protocol (GVRP) and GARP Multicast Registration Protocol (GMRP) protocols.

A VLAN-bridged network usually restricts unknown unicast, multicast, and broadcast traffic to those links that the traffic uses to access the appropriate network devices. In a large network, localized topology changes can affect the service over a much larger portion of the network. IEEE 802.1ak replaces GARP with MRP, which provides improved resource utilization and bandwidth conservation.

With the 802.1ak MRP attribute encoding scheme, MVRP only needs to send one PDU that includes the state of all 4094 VLANs on a port. MVRP also transmits Topology Change Notifications (TCNs) for individual VLANs. This is an important feature for service providers because it allows them to localize topology changes. Figure 21-1 illustrates MVRP deployed in a provider network on provider and customer bridges.

Figure 21-1 MVRP Deployed on Provider and Customer Bridges

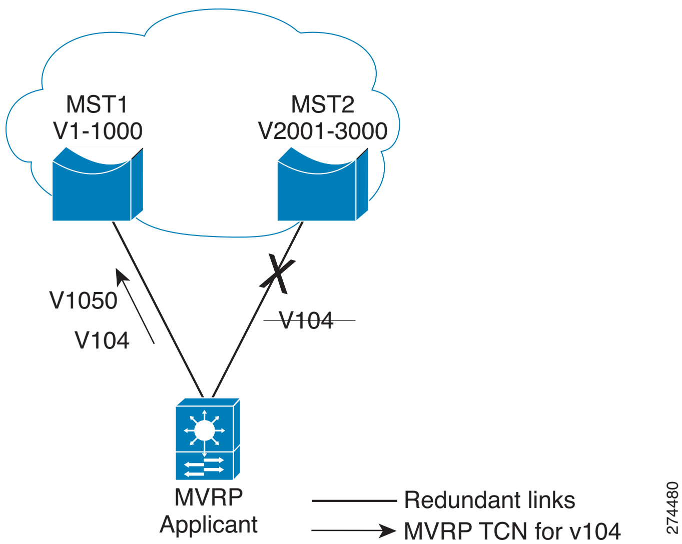


Because most providers do not wish to filter traffic by destination MAC addresses, a pruning protocol like MVRP is important in a Metro Ethernet provider network, which often uses thousands of VLANs.

Figure 21-2 dispalys redundant links that are configured between the access switch and two distribution switches on the cloud. When the link with VLAN 104 fails over, MVRP needs to send only one TCN for VLAN 104. Without MVRP, an STP TCN would need to be sent out for the whole MST region (VLANs1-1000), which could cause unnecessary network interruption.

STP sets the tcDetected variable to signal MVRP that MVRP must decide whether to send an MVRP TCN. MVRP can flush filtering database entries rapidly on a per-VLAN basis following a topology change because when a port receives an attribute declaration marked as new, any entries in the filtering database for that port and for that VLAN are removed.

Figure 21-2 MVRP TCN Application



Dynamic VLAN Creation

Virtual Trunking Protocol (VTP) is a Cisco proprietary protocol that distributes VLAN configuration information across multiple devices within a VTP domain. When VTP is running on MVRP-aware devices, all of the VLANs allowed on the Cisco bridged LAN segments are determined by VTP.

Only the VTP transparent mode supports MVRP dynamic VLAN creation. When dynamic VLAN creation is disabled, the MVRP trunk ports can register and propagate the VLAN messages only for existing VLANs. MVRP PDUs and MVRP messages for the nonexistant VLANs are discarded.

For a switch to be configured in full compliance with the MVRP standard, the switch VTP mode must be transparent and MVRP dynamic VLAN creation must be enabled.

MVRP Interoperability with VTP

These sections describe MVRP and VTP:

- Overview, page 21-4
- VTP in Transparent or Off Mode, page 21-4

- VTP in Server or Client Mode and VTP Pruning is Disabled, page 21-4
- VTP in Server or Client Mode and VTP Pruning is Enabled, page 21-4

Overview

The VLAN Trunking Protocol (VTP) is a Cisco proprietary protocol that distributes VLAN configuration information across multiple devices within a VTP domain. VTP pruning is an extension of VTP. It has its own Join message that can be exchanged with VTP PDUs. VTP PDUs can be transmitted on both 802.1Q trunks and ISL trunks. A VTP-capable device is in one of the VTP modes: server, client, transparent, or off.

When VTP Pruning and MVRP are both enabled globally, MVRP runs on trunks where it is enabled and VTP Pruning runs on other trunks. MVRP or VTP pruning can be enabled on a trunk, but not both.

VTP in Transparent or Off Mode

When VTP is in transparent or off mode, VTP pruning is not supported and VTP PDUs are not processed.

When a port receives an MVRP join message for a VLAN, the port transmits broadcast, multicast, and unknown unicast frames in that VLAN and adds the traffic definition to the MRP Attribute Propagation (MAP) port configured for that VLAN. The mapping is removed when the VLAN is no longer registered on the port.

For each interface that is forwarding in each VLAN, MVRP issues a join request to each MRP Attribute Declaration (MAD) instance and an MVRP Join message is sent out on each corresponding MVRP port.

MVRP dynamic VLAN creation can be enabled in VTP transparent or off mode. If it is enabled and the VLAN registered by a join message does not exist in the VLAN database in the device, then the VLAN will be created.

VTP in Server or Client Mode and VTP Pruning is Disabled

MVRP functions like VTP in transparent or off mode, except that MVRP dynamic VLAN creation is not allowed.

VTP in Server or Client Mode and VTP Pruning is Enabled

MVRP and VTP with pruning disabled can be supported on the same port and these two protocols need to communicate and exchange pruning information.

When VTP receives a VTP join message on a VTP trunk, MVRP is notified so that join request can be posted to the MVRP port MAD instances, and MVRP join messages are out on the MVRP ports to the MVRP network.

When VTP pruning removes a VLAN from a VTP trunk, MVRP sends a leave request to all the MAD instances and the MAD instances send a leave or empty message from the MVRP ports to indicate that the VLAN is not configured on the device.

When an MVRP port received an MVRP join message, MVRP propagates the event to other MVRP ports in the same MAP context, and notifies VTP so that VTP pruning can send a VTP join message from the VTP trunk ports.

If MVRP learns that a VLAN is no longer declared by the neighboring devices, MVRP sends a withdrawal event to VTP and then VTP pruning verifies that it should continue sending VTP join messages.

For VLANs that are configured as VTP pruning non-eligible on the VTP trunks, the VTP pruning state variables are set to joined for the VLANs. MVRP join requests are sent to those VLANs through the MVRP ports.

MVRP Interoperation with Non-Cisco Devices

Non-Cisco devices can interoperate with a Cisco device only through 802.1q trunks.

MVRP Interoperability with Other Software Features and Protocols

These sections describes MVRP interoperability with these protocols:

802.1x and Port Security

802.1x authenticates and authorizes a port after it transitions to the link-up state, but before DTP negotiation occurs and MVRP runs on a port. Port security works independently of MVRP.

Note When MVRP is globally enabled, the MVRP MAC address auto detect and provision feature is disabled by default (**mvrp mac-learning auto**). In some situations, MVRP MAC address auto detect and provision can disable MAC address learning and prevent correct port security operation. For example, on ports where port security is configured, when the number of streams exceeds the configured maximum number of MAC addresses, no port security violation occurs because MAC address learning is disabled, which prevents updates to port security about the streams coming into the port. To avoid incorrect port security operation, use caution when enabling the MVRP MAC address auto detect and provision feature on ports where port security is configured.

DTP

DTP negotiation occurs after ports transition to the link-up state and before transition to the forwarding state. If MVRP is administratively enabled globally and enabled on a port, it becomes operational when the port starts trunking.

EtherChannel

An EtherChannel port-channel interface can be configured as an MVRP participant. The EtherChannel member ports cannot be MVRP participants. MVRP learns the STP state of EtherChannel port-channel interfaces. The MAP context applies to the EtherChannel port-channel interfaces, but not to the EtherChannel member ports.

Flex Links

MVRP declares VLANs on STP forwarding ports but not on ports in the blocking state. On flex links ports, MVRP declares VLANs on the active ports but not on the standby ports. when a standby port takes over and an active port transitions to the link-down state, MVRP declares the VLANs on the newly active port.

High Availability

State Switchover (SSO) and ISSU supports MVRP.

ISSU and eFSU

Enhanced Fast Software Upgrade (EFSU) is an enhanced software upgrade procedure. MVRP is serviced by the ISSU client identified as ISSU_MVRP_CLIENT_ID.

L2PT

Layer 2 Protocol Tunneling (L2PT) does not support MVRP PDUs on 802.1Q tunnel ports.

SPAN

MVRP ports can be configured as either Switched Port Analyzer (SPAN) sources or destinations.

Unknown Unicast and Multicast Flood Control

MVRP and the Unknown Unicast and Multicast Flood Control feature, configured with the **switchport block** command, cannot be configured on the same port.

STP

An STP mode change causes forwarding ports to leave the forwarding state until STP reconverges in the newly configured mode. The reconvergence might cause an MVRP topology change because join messages might be received on different forwarding ports, and leave timers might expire on other ports.

UDLR

MVRP and unidirectional link routing (UDLR) cannot be configured on the same port.

VLANs with MVRP

This sections decribe interactions between VLANs and MVRP:

- VLAN Translation, page 21-7
- 802.1Q Native VLAN Tagging, page 21-7
- Private VLANs, page 21-7

VLAN Translation

VLAN translation and MVRP cannot be configured on the same port.

802.1Q Native VLAN Tagging

Other MVRP participants might not be able to accept tagged MVRP PDUs in the 802.1Q native VLAN. Compatibility between MVRP and 802.1Q native VLAN tagging depends on the specific network configuration.

Private VLANs

Private VLAN ports cannot support MVRP.

IEEE 802.1ak MVRP and MRP Guidelines and Restrictions

When configuring IEEE 802.1ak MVRP and MRP, follow these guidelines and restrictions:

- In releases where CSCta96338 is not resolved, a physical port with an MVRP configuration and enable state that differs from what is configured on a port-channel interface cannot become an active member of that EtherChannel.
- In releases where CSCta96338 is resolved, a physical port with an MVRP configuration and enable state that differs from what is configured on a port-channel interface can become an active member of the EtherChannel because the physical port will use the port-channel interface MVRP configuration and enable state.
- A non-Cisco device can interoperate with a Cisco device only through 802.1Q trunks.
- MVRP runs on ports where it is enabled. VTP pruning can run on ports where MVRP is not enabled.
- MVRP can be configured on both physical interfaces and EtherChannel interfaces, but is not supported on EtherChannel member ports.
- MVRP dynamic VLAN creation is not supported when the device is running in VTP server or client mode.
- MVRP and Connectivity Fault Management (CFM) can coexist but if the module does not have enough MAC address match registers to support both protocols, the MVRP ports on those modules are put in the error-disabled state. To use the ports that have been shut down, disable MVRP on the ports, and then enter **shutdown** and **no shutdown** commands.
- 802.1X authentication and authorization takes place after the port becomes active and before the Dynamic Trunking Protocol (DTP) negotiations start prior to MVRP running on the port.
- Do not enable MVRP automatic MAC address learning on edge switches that are configured with access ports. Enable MVRP automatic MAC address learning only on core switches where all the trunk interfaces are running MVRP.

- MVRP is supported only on Layer 2 trunks. MVRP is not supported on subinterfaces.

Configuring IEEE 802.1ak MVRP and MRP

These sections describe how to configure IEEE 802.1ak MVRP and MRP:

Enabling MVRP

MVRP must be enabled globally and on trunk ports. To enable MVRP, perform this task:

	Command or Action	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# mvrp global`	Globally enables MVRP.
Step 4	`Router(config)# interface type number`	Specifies and interface and enters interface configuration mode.
Step 5	`Router(config-if)# mvrp`	Enables MVRP on the interface. **Note** If MVRP is not successfully enabled on the port, the port is put in the errdisabled state. Enter the **shutdown** and **no shutdown** commands to clear the errdisabled state.

This example shows how to enable MVRP globally and on an interface:

```
Router> enable
Router# configure terminal
Router(config)# mvrp global
Router(config)# interface FastEthernet 2/1
Router(config-if)# mvrp
```

Enabling Automatic Detection of MAC Addresses

MVRP automatic detection of MAC addresses is disabled by default. To enable MVRP automatic detection of MAC addresses on VLANs, perform this task:

	Command or Action	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# mvrp mac-learning auto`	Enables MAC address learning.

This example shows how to enable automatic MAC address learning:

```
Router> enable
Router# configure terminal
Router(config)# mvrp mac-learning auto
```

Enabling MVRP Dynamic VLAN Creation

To enable MVRP dynamic VLAN creation, perform this task:

	Command or Action	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# vtp mode transparent`	Sets VTP mode to transparent. **Note** Required for MVRP dynamic VLAN creation.
Step 4	`Router(config)# mvrp vlan creation`	Enables MRVP dynamic VLAN creation.

This example shows how to enable MVRP dynamic VLAN creation:

```
Router> enable
Router# configure terminal
Router(config)# vtp mode transparent
Router(config)# mvrp vlan create
```

Changing the MVRP Registrar State

The MRP protocol allows one participant per application in an end station, and one per application per port in a bridge. To set the MVRP registrar state, perform this task:

	Command or Action	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	`Router(config)# interface type number`	Specifies and interface and enters interface configuration mode.
Step 4	`Router(config-if)# mvrp registration [normal	fixed	forbidden]`	Registers MVRP with the MAD instance.

This example shows how to set the MVRP registrar state to normal:

```
Router> enable
Router# configure terminal
Router(config)# interface FastEthernet 2/1
Router(config-if)# mvrp registration normal
```

Troubleshooting the MVRP Configuration

Use the **show mvrp summary** and **show mvrp interface** commands to display configuration information and interface states, and the **debug mvrp** command to enable all or a limited set of output messages related to an interface.

To troubleshoot the MVRP configuration, perform this task:

	Command or Action	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# show mvrp summary`	Displays the MVRP configuration.
Step 3	`Router# show mvrp interface interface-type port/slot`	Displays the MVRP interface states for the specified interface.
Step 4	`Router# debug mvrp`	Displays MVRP debugging information.
Step 5	`Router# clear mvrp statistics`	Clears MVRP statistics on all interfaces.

The following is sample output from the **show mvrp summary** command. This command can be used to display the MVRP configuration at the device level.

```
Router# show mvrp summary

MVRP global state            : enabled
MVRP VLAN creation           : disabled
VLANs created via MVRP       : 20-45, 3001-3050
Learning disabled on VLANs   : none
```

The following is sample output from the **show mvrp interface** command. This command can be used to display MVRP interface details of the administrative and operational MVRP states of all or one particular trunk port in the device.

```
Router# show mvrp interface

Port       Status   Registrar State
Fa3/1      off      normal

Port       Join Timeout   Leave Timeout   Leaveall Timeout
Fa3/1      201  600       700             1000

Port       Vlans Declared
```

```
Fa3/1      none

Port       Vlans Registered
Fa3/1      none

Port       Vlans Registered and in Spanning Tree Forwarding State
Fa3/1      none
```

Configuration Examples for IEEE 802.1ak MVRP and MRP

This section provides the following configuration examples:

Enabling MVRP

The following example shows how to enable MVRP:

```
Router> enable
Router# configure terminal
Router(config)# mvrp global
Router(config)# interface fastethernet2/1
Router(config-if)# mvrp
```

Enabling MVRP Automatic Detection of MAC Addresses

The following example shows how to enable MAC address learning:

```
Router> enable
Router# configure terminal
Router(config)# mvrp mac-learning auto
```

Enabling Dynamic VLAN Creation

The following example shows how to enable dynamic VLAN creation:

```
Router> enable
Router# configure terminal
Router(config)# vtp mode transparent
Router(config)# mvrp vlan create
```

Changing the MVRP Registrar State

The following example shows how to change the MVRP registrar state:

```
Router> enable
Router# configure terminal
Router(config)# mvrp registration normal
```

Configuring VTP

This chapter describes how to configure the VLAN Trunking Protocol (VTP) in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding VTP

VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs within a VTP domain. A VTP domain (also called a VLAN management domain) is made up of one or more network devices that share the same VTP domain name and that are interconnected with trunks. VTP minimizes misconfigurations and configuration inconsistencies that can result in a number of problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations. Before you create VLANs, you must decide whether to use VTP in your network. With VTP, you can make configuration changes centrally on one or more network devices and have those changes automatically communicated to all the other network devices in the network.

Note For complete information on configuring VLANs, see Chapter 23, "Configuring VLANs."

These sections describe how VTP works:

Understanding the VTP Domain

A VTP domain (also called a VLAN management domain) is made up of one or more interconnected network devices that share the same VTP domain name. A network device can be configured to be in one and only one VTP domain. You make global VLAN configuration changes for the domain using either the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

VTP server mode is the default and the switch is in the no-management domain state until it receives an advertisement for a domain over a trunk link or you configure a management domain.

If the switch receives a VTP advertisement over a trunk link, it inherits the management domain name and the VTP configuration revision number. The switch ignores advertisements with a different management domain name or an earlier configuration revision number.

If you configure the switch as VTP transparent, you can create and modify VLANs but the changes affect only the individual switch. The valid VLAN ranges are as follows:

- VTP version 1 and version 2 support VLANs 1 to 1000 only.
- In Cisco IOS Release 12.2(33)SXI and later releases, VTP version 3 is supported. In VTP version 3, the entire VLAN range is supported (VLANs 1 to 4094).
- The pruning of VLANs still applies to VLANs 1 to 1000 only.
- Extended-range VLANs are supported only in VTP version 3. If converting from VTP version 3 to VTP version 2, VLANs in the range 1006 to 4094 are removed from VTP control.

By default, all devices come up as secondary servers. You can enter the **vtp primary** privileged EXEC mode command to specify a primary server.

In Cisco IOS releases prior to Release 12.2(33)SXI, when using VTP version 1 and version 2, a VTP server is used to back up the database to the NVRAM and allows you to change the database information.

In Cisco IOS Release 12.2(33)SXI and later releases, VTP version 3 is supported. In VTP version 3, there is a VTP-primary server and a VTP-secondary server. A primary server allows you to alter the database information and the database updates sent out are honored by all the devices in the system. A secondary server can only back up the updated VTP configuration received from the primary server in the NVRAMs. The status of the primary and secondary servers is a runtime status and is not configurable.

VTP maps VLANs dynamically across multiple LAN types with unique names and internal index associations. Mapping eliminates excessive device administration required from network administrators.

Understanding VTP Modes

You can configure any one of these VTP modes:

- Server—In VTP server mode, you can create, modify, and delete VLANs and specify other configuration parameters (such as VTP version and VTP pruning) for the entire VTP domain. VTP servers advertise their VLAN configuration to other network devices in the same VTP domain and synchronize their VLAN configuration with other network devices based on advertisements received over trunk links. VTP server is the default mode.
- Client—VTP clients behave the same way as VTP servers, but you cannot create, change, or delete VLANs on a VTP client.
- Transparent—VTP transparent network devices do not participate in VTP. A VTP transparent network device does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements. However, in VTP version 2, a transparent network device will forward received VTP advertisements from its trunking LAN ports. In VTP version 3, a transparent network device is specific to an instance.
- Off—In VTP off mode, a network device functions in the same manner as a VTP transparent device except that it does not forward VTP advertisements.

Note The VTP server mode automatically changes from VTP server mode to VTP client mode if the switch detects a failure while writing configuration to NVRAM. If this happens, the switch cannot be returned to VTP server mode until the NVRAM is functioning.

Understanding VTP Advertisements

Each network device in the VTP domain sends periodic advertisements out each trunking LAN port to a reserved multicast address. VTP advertisements are received by neighboring network devices, which update their VTP and VLAN configurations as necessary.

The following global configuration information is distributed in VTP version 1 and version 2 advertisements:

- VLAN IDs (ISL and 802.1Q).
- Emulated LAN names (for ATM LANE).
- 802.10 SAID values (FDDI).
- VTP domain name.
- VTP configuration revision number.
- VLAN configuration, including the maximum transmission unit (MTU) size for each VLAN.
- Frame format.

In VTP version 3, the information distributed in VTP version 1 and version 2 advertisements are supported, as well as the following information:

- A primary server ID.
- An instance number.

- A start index.
- An advertisement request is sent by a Client or a Server in these situations:
  - On a trunk coming up on a switch with an invalid database.
  - On all trunks when the database of a switch becomes invalid as a result of a configuration change or a takeover message.
  - On a specific trunk where a superior database has been advertised.
- VTP version 3 adds the following fields to the subset advertisement request:
  - A primary server ID.
  - An instance number.
  - A window size.
  - A start index.

Understanding VTP Authentication

In releases prior to Cisco IOS Release 12.2(33)SXI, the secret that is used to validate the received VTP updates is visible in plain text in the **show** commands and the NVRAM file, const_nvram:vlan.dat. In the event that a device in a VTP domain is compromised, the administrator had to change the VTP secret across all the devices in the VTP domain.

In Cisco IOS Release 12.2(33)SXI and later releases, VTP version 3 is supported. In VTP version 3, you can configure the authentication password to be hidden using the **vtp password** command. When you configure the authentication password to be hidden, it does not appear in plain text in the configuration. Instead, the secret associated with the password is saved in hexadecimal format in the running configuration. The *password-string* argument is an ASCII string from 1 to 64 characters identifying the administrative domain for the device.

Understanding VTP Version 2

If you use VTP in your network, you must decide whether to use VTP version 1 or version 2.

Note If you are using VTP in a Token Ring environment, you must use version 2.

VTP version 2 supports the following features not supported in version 1:

- Token Ring support—VTP version 2 supports Token Ring LAN switching and VLANs (Token Ring Bridge Relay Function [TrBRF] and Token Ring Concentrator Relay Function [TrCRF]). For more information about Token Ring VLANs, see the "Understanding VLANs" section on page 23-1.
- Unrecognized Type-Length-Value (TLV) Support—A VTP server or client propagates configuration changes to its other trunks, even for TLVs that it is not able to parse. The unrecognized TLV is saved in NVRAM.
- Version-Dependent Transparent Mode—In VTP version 1, a VTP transparent network device inspects VTP messages for the domain name and version and forwards a message only if the version and domain name match. Because only one domain is supported, VTP version 2 forwards VTP messages in transparent mode without checking the version.

- Consistency Checks—In VTP version 2, VLAN consistency checks (such as VLAN names and values) are performed only when you enter new information through the CLI or SNMP. Consistency checks are not performed when new information is obtained from a VTP message, or when information is read from NVRAM. If the digest on a received VTP message is correct, its information is accepted without consistency checks.

Understanding VTP Version 3

Note If you are using VTP in a Token Ring environment, you must use version 2.

In Cisco IOS Release 12.2(33)SXI and later releases, VTP version 3 is supported. VTP version 3 supports all the features in version 1 and version 2. VTP version 3 also supports the following features not supported in version 1 and version 2:

- Enhanced authentication—In VTP version 3, you can configure the authentication password to be hidden using the **vtp password** command. When you configure the authentication password to be hidden, it does not appear in plain text in the configuration. Instead, the secret associated with the password is saved in hexadecimal format in the running configuration. The *password-string* argument is an ASCII string from 1 to 64 characters identifying the administrative domain for the device.

  The **hidden** and **secret** keywords for VTP password are supported only in VTP version 3. If converting to VTP version 2 from VTP version 3, you must remove the **hidden** or **secret** keyword prior to the conversion. These keywords are supported on the Catalyst 6500 series switch only.
- Support for extended range VLAN database propagation—VTP version 1 and version 2 support VLANs 1 to 1000 only. In VTP version 3, the entire VLAN range is supported (VLANs 1 to 4094). The pruning of VLANs still applies to VLANs 1 to 1000 only. Extended-range VLANs are supported in VTP version 3 only. Private VLANs are supported in VTP version 3. If you convert from VTP version 3 to VTP version 2, the VLANs in the range 1006 to 4094 are removed from VTP control.
- VLANs 1002 to 1005 are reserved VLANs in VTP version 1, version 2, and version 3.
- Support for propagation of any database in a domain—In VTP version 1 and version 2, a VTP server is used to back up the database to the NVRAM and allows you to change the database information.

  Note In Cisco IOS Release 12.2(33)SXI and later releases, VTP version 3 supports Multiple Spanning Tree (802.1s) (MST) database propagation separate from the VLAN database only. In the MST database propagation, there is a VTP primary server and a VTP econdary server. A primary server allows you to alter the database information, and the database updates sent out are honored by all the devices in the system. A secondary server can only back up the updated VTP configuration received from the primary server in the NVRAMs. The status of the primary and secondary servers is a runtime status and is not configurable.

  By default, all devices come up as secondary servers. You can enter the **vtp primary** privileged EXEC mode command to specify a primary server.

  The primary-server status is needed only when database changes have to be performed and is obtained when the administrator issues a takeover message in the domain. The primary-server status is lost when you reload, switch over, or the domain parameters change. The secondary servers back up the configuration and continue to propagate the database. You can have a working VTP domain without any primary servers. Primary and secondary servers may exist on an instance in the domain.

In VTP version 3, there is no longer a restriction to propagate only VLAN database information. You can use VTP version 3 to propagate any database information across the VTP domain. A separate instance of the protocol is running for each application that uses VTP.

Two VTP version 3 regions can only communicate over a VTP version 1 or VTP version 2 region in transparent mode.

- CLI to turn off/on VTP on a per-trunk basis—You can enable VTP on a per-trunk basis using the **vtp** interface configuration mode command. You can disable VTP on a per-trunk basis using the **no** form of this command. When you disable VTP on the trunking port, all the VTP instances for that port are disabled. You will not be provided with the option of setting VTP to OFF for the MST database and ON for the VLAN database.

VTP on a global basis—When you set VTP mode to OFF globally, this applies to all the trunking ports in the system. Unlike the per-port configuration, you can specify the OFF option on a per-VTP instance basis. For example, the system could be configured as VTP-server for the VLAN database and as VTP-off for the MST database. In this case, VLAN databases are propagated by VTP, MST updates are sent out on the trunk ports in the system, and the MST updates received by the system are discarded.

Understanding VTP Pruning

VTP pruning enhances network bandwidth use by reducing unnecessary flooded traffic, such as broadcast, multicast, unknown, and flooded unicast packets. VTP pruning increases available bandwidth by restricting flooded traffic to those trunk links that the traffic must use to access the appropriate network devices. By default, VTP pruning is disabled.

In VTP versions 1 and 2, when you enable or disable pruning, it is propagated to the entire domain and accepted by all the devices in that domain. In VTP version 3, the domain administrator must manually enable or disable VTP pruning explicitly on each device.

For VTP pruning to be effective, all devices in the management domain must support VTP pruning. On devices that do not support VTP pruning, you must manually configure the VLANs allowed on trunks.

Figure 22-1 shows a switched network without VTP pruning enabled. Interface 1 on network Switch 1 and port 2 on Switch 4 are assigned to the Red VLAN. A broadcast is sent from the host connected to Switch 1. Switch 1 floods the broadcast, and every network device in the network receives it, even though Switches 3, 5, and 6 have no ports in the Red VLAN.

You enable pruning globally on the switch (see the "Enabling VTP Pruning" section on page 22-13). You configure pruning on Layer 2 trunking LAN ports (see the "Configuring a Layer 2 Switching Port as a Trunk" section on page 17-10).

Figure 22-1 Flooding Traffic without VTP Pruning

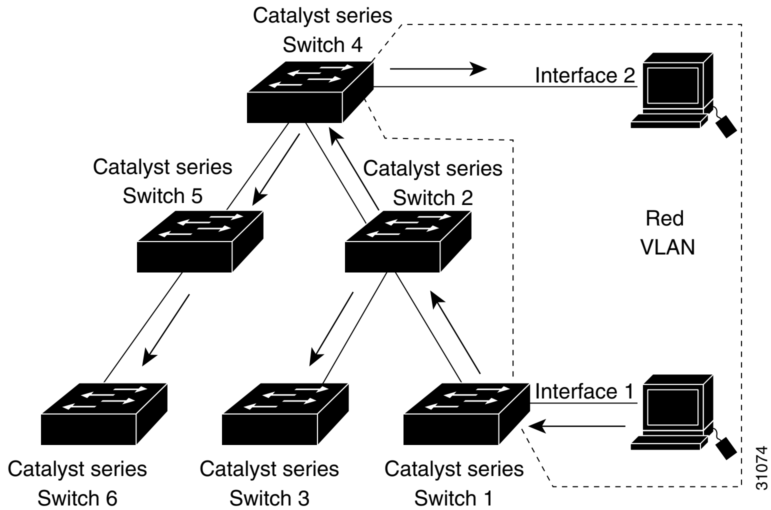
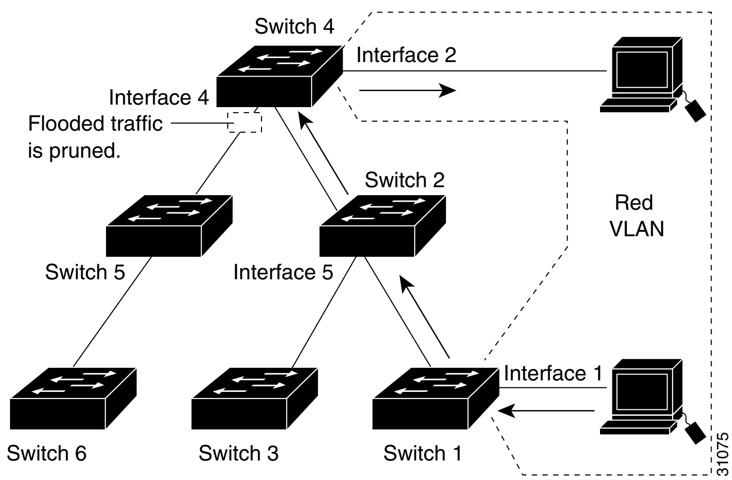


Figure 22-2 shows the same switched network with VTP pruning enabled. The broadcast traffic from Switch 1 is not forwarded to Switches 3, 5, and 6 because traffic for the Red VLAN has been pruned on the links indicated (port 5 on Switch 2 and port 4 on Switch 4).

Figure 22-2 Flooding Traffic with VTP Pruning



Enabling VTP pruning on a VTP server enables pruning for the entire management domain. VTP pruning takes effect several seconds after you enable it. By default, VLANs 2 through 1000 are pruning eligible. VTP pruning does not prune traffic from pruning-ineligible VLANs. VLAN 1 is always pruning ineligible; traffic from VLAN 1 cannot be pruned.

To configure VTP pruning on a trunking LAN port, use the **switchport trunk pruning vlan** command (see the "Configuring a Layer 2 Switching Port as a Trunk" section on page 17-10). VTP pruning operates when a LAN port is trunking. You can set VLAN pruning eligibility when VTP pruning is enabled or disabled for the VTP domain, when any given VLAN exists or not, and when the LAN port is currently trunking or not.

VLAN Interaction

This section describes the VLAN interaction between devices with different VTP versions:

- Interaction Between VTP Version 3 and VTP Version 2 Devices, page 22-8
- Interaction Between VTP Version 3 and VTP Version 1 Devices, page 22-8

Interaction Between VTP Version 3 and VTP Version 2 Devices

When a VTP version 3 device on a trunk port receives messages from a VTP version 2 device, the VTP version 3 device sends a scaled-down version of the VLAN database on that particular trunk in a VTP version 2 format. A VTP version 3 device does not send out VTP version 2-formatted packets on a trunk port unless it first receives VTP version 2 packets on that trunk. If the VTP version 3 device does not receive VTP version 2 packets for an interval of time on the trunk port, the VTP version 3 device stops transmitting VTP version 2 packets on that trunk port.

Even when a VTP version 3 device detects a VTP version 2 device on a trunk port, the VTP version 3 device continues to send VTP version 3 packets in addition to VTP version 3 device 2 packets, to allow two kinds of neighbors to coexist on the trunk. VTP version 3 sends VTP version 3 and VTP version 2 updates on VTP version 2-detected trunks.

A VTP version 3 device does not accept configuration from a VTP version 2 (or VTP version 1) device.

Unlike in VTP version 2, when you configure the VTP version to be version 3, version 3 does not configure all the VTP version 3-capable devices in the domain to start behaving as VTP version 3 systems.

Interaction Between VTP Version 3 and VTP Version 1 Devices

When a VTP version 1 device that is capable of VTP version 2 or VTP version 3 receives a VTP version 3 packet, it will be configured as a VTP version 2 device if VTP version 2 conflicts do not exist.

VTP version 1-only capable devices cannot interoperate with VTP version 3 devices.

VTP Default Configuration

Table 22-1 shows the default VTP configuration.

Table 22-1 VTP Default Configuration

Feature	Default Value
VTP domain name	Null
VTP version 1 and version 2 mode	Server
VTP version 3 mode	The VTP version 3 VLAN database mode is the same as the VLAN database mode in VTP version 1 or 2 after the conversion from VTP version 1 or 2 to VTP version 3. For example, the VTP version 1 or 2 VLAN database mode is carried over to VTP version 3 VLAN database mode.
MST database mode	Transparent

Table 22-1 VTP Default Configuration (continued)

Feature	Default Value
VTP version 3 server type	Secondary
VTP version 2 state	Version 2 is disabled
VTP password	None
VTP pruning	Disabled

VTP Configuration Guidelines and Restrictions

When implementing VTP in your network, follow these guidelines and restrictions:

- Supervisor engine redundancy does not support nondefault VLAN data filenames or locations. Do not enter the **vtp file** *file_name* command on a switch that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- All network devices in a VTP domain must run the same VTP version.
- You must configure a password on each network device in the management domain when in secure mode.

Caution If you configure VTP in secure mode, the management domain will not function properly if you do not assign a management domain password to each network device in the domain.

- A VTP version 2-capable network device can operate in the same VTP domain as a network device that runs VTP version 1 if VTP version 2 is disabled on the VTP version 2-capable network device (VTP version 2 is disabled by default).
- Do not enable VTP version 2 on a network device unless all of the network devices in the same VTP domain are version 2-capable. When you enable VTP version 2 on a network device, all of the version 2-capable network devices in the domain enable VTP version 2.
- In a Token Ring environment, you must enable VTP version 2 for Token Ring VLAN switching to function properly.
- When you enable or disable VTP pruning on a VTP server, VTP pruning for the entire management domain is enabled or disabled.
- The pruning-eligibility configuration applies globally to all trunks on the switch. You cannot configure pruning eligibility separately for each trunk.
- When you configure VLANs as pruning eligible or pruning ineligible, pruning eligibility for those VLANs is affected on that switch only, not on all network devices in the VTP domain.
- VTP version 1 and VTP version 2 do not propagate configuration information for extended-range VLANs (VLAN numbers 1006 to 4094). You must configure extended-range VLANs manually on each network device.
- VTP version 3 supports extended-range VLANs (VLAN numbers 1006 to 4094). If you convert from VTP version 3 to VTP version 2, the VLANs in the range 1006 to 4094 are removed from VTP control.
- VTP version 3 supports propagation of any database in a domain by allowing you to configure a primary and secondary server.

- In Cisco IOS Release 12.2(33)SXI and later releases, the network administrator has to manually configure VTP version 3 on the switches that need to run VTP version 3.
- Prior to configuring VTP version 3, you must ensure that the **spanning-tree extend** *system-id* command has been enabled.
- VTP version 3 is not supported on private VLAN (PVLAN) ports.
- If there is insufficient DRAM available for use by VTP, the VTP mode changes to transparent.
- Network devices in VTP transparent mode do not send VTP Join messages. On trunk connections to network devices in VTP transparent mode, configure the VLANs that are used by the transparent-mode network devices or that need to be carried across trunks as pruning ineligible. For information about configuring prune eligibility, see the "Configuring the List of Prune-Eligible VLANs" section on page 17-14.

Configuring VTP

These sections describe how to configure VTP:

Configuring VTP Global Parameters

These sections describe configuring the VTP global parameters:

Note You can enter the VTP global parameters in either global configuration mode or in EXEC mode.

Configuring VTP Version 1 and Version 2 Passwords

To configure the VTP version 1 and version 2 global parameters, perform this task:

Command	Purpose
`Router(config)#` **vtp password** *password-string*	Sets a password, which can be from 1 to 64 characters long, for the VTP domain.
`Router(config)#` **no vtp password**	Clears the password.

This example shows one way to configure a VTP password in global configuration mode:

```
Router# configure terminal
Router(config)# vtp password WATER
Setting device VLAN database password to WATER.
Router#
```

This example shows how to configure a VTP password in EXEC mode:

```
Router# vtp password WATER
Setting device VLAN database password to WATER.
Router#
```

Note The password is not stored in the running-config file.

Configuring VTP Version 3 Password

To configure the VTP version 3 password, perform this task:

Command	Purpose
Router(config)# **vtp password** *password-string* [**hidden** | **secret**]	Configures a password, which can be from 1 to 64 characters long or in 32-digit hexadecimal format, for the VTP domain. **Note** When entering the **secret** keyword, the *password-string* must be entered in 32-digit hexadecimal format.
Router(config)# **no vtp password**	Clears the password.

This example shows one way to configure a VTP password in global configuration mode:

```
Router# configure terminal
Router(config)# vtp password water
Setting device VTP database password to water.
Router#
```

Note If you configure a VTP password in EXEC mode, the password is not stored in the running-config file.

This example shows one way to configure the password with a hidden key saved in hexadecimal format in the running configuration:

```
Router# configure terminal
Router(config)# vtp password 82214640C5D90868B6A0D8103657A721 hidden
Setting device VTP password
Router#
```

This example shows how you configure the password secret key in hexadecimal format:

```
Router# configure terminal
Router(config)# vtp password 300F060A2B06010353010201070102 01 secret
Setting device VTP password
Router#
```

Configuring VTP Version 3 Server Type

To specify a primary server, perform this task:

	Command	Purpose
Step 1	Router# **vtp primary** [**vlan** | **mst**] [**force**]	Configure this device as the primary server.
Step 2	Router# **show vtp status**	Verifies the configuration.

The **vtp primary** command does not have a **no** form. To return to the secondary server status, one of the following conditions must be met:

- System reload.
- Switchover between redundant supervisors.
- Takeover from another server.
- Change in the mode configuration.
- Any domain configuration change (version, domain name, domain password).

This example shows how to configure this device as the primary server if the password feature is disabled:

```
Router# vtp primary
This system is becoming primary server for feature vlan
No conflicting VTP version 3 devices found.
Do you want to continue? [confirm]y
Router#
```

This example shows how to configure this device as the primary server for the VTP VLAN feature if the password feature is disabled:

```
Router# vtp primary vlan
This system is becoming primary server for feature vlan
No conflicting VTP version 3 devices found.
Do you want to continue? [confirm]y
Router#
```

This example shows how to force this device to be the primary server for the VTP MST feature if the password feature is disabled:

```
Router# vtp primary mst force
This system is becoming primary server for feature MST
No conflicting VTP version 3 devices found.
Do you want to continue? [confirm]y
Router#
```

This example shows how to force this device to be the primary server for the VTP MST feature when the domain VTP password is set with the **hidden** or **secret** keyword:

```
Router# vtp primary mst force
Enter VTP password: water1
This switch is becoming Primary server for mst feature in the VTP domain
VTP Database Conf Switch ID      Primary Server Revision System Name
------------ ---- -------------- -------------- -------- --------------------
VLANDB       Yes  00d0.00b8.1400=00d0.00b8.1400 1        stp7
Do you want to continue (y/n) [n]? y
Router#
```

Enabling VTP Pruning

To enable VTP pruning in the management domain, perform this task:

	Command	Purpose
Step 1	`Router(config)# vtp pruning`	Enables VTP pruning in the management domain.
	`Router(config)# no vtp pruning`	Disables VTP pruning in the management domain.
Step 2	`Router# show vtp status`	Verifies the configuration.

This example shows one way to enable VTP pruning in the management domain:

```
Router# configure terminal
Router(config)# vtp pruning
Pruning switched ON
```

This example shows how to enable VTP pruning in the management domain with any release:

```
Router# vtp pruning
Pruning switched ON
```

This example shows how to verify the configuration:

```
Router# show vtp status | include Pruning
VTP Pruning Mode: Enabled
Router#
```

For information about configuring prune eligibility, see the "Configuring the List of Prune-Eligible VLANs" section on page 17-14.

Enabling VTP Version 2

VTP version 2 is disabled by default on VTP version 2-capable network devices. When you enable VTP version 2 on a network device, every VTP version 2-capable network device in the VTP domain enables version 2.

Caution VTP version 1 and VTP version 2 are not interoperable on network devices in the same VTP domain. Every network device in the VTP domain must use the same VTP version. Do not enable VTP version 2 unless every network device in the VTP domain supports version 2.

Note In a Token Ring environment, you must enable VTP version 2 for Token Ring VLAN switching to function properly on devices that support Token Ring interfaces.

To enable VTP version 2, perform this task:

	Command	Purpose
Step 1	`Router(config)# vtp version {1 | 2}`	Enables VTP version 2.
	`Router(config)# no vtp version`	Reverts to the default (VTP version 1).
Step 2	`Router# show vtp status`	Verifies the configuration.

This example shows one way to enable VTP version 2:

```
Router# configure terminal
Router(config)# vtp version 2
V2 mode enabled.
Router(config)#
```

This example shows how to enable VTP version 2 with any release:

```
Router# vtp version 2
V2 mode enabled.
Router#
```

This example shows how to verify the configuration:

```
Router# show vtp status | include V2
VTP V2 Mode: Enabled
Router#
```

Enabling VTP Version 3

VTP version 3 is disabled by default. You can enable version 3 in global configuration mode only. In Cisco IOS Release 12.2(33)SXI and later releases, the network administrator has to manually configure VTP version 3 on the switches that need to run VTP version 3.

Note Prior to configuring VTP version 3, you must ensure that the **spanning-tree extend** *system-id* command has been enabled.

Caution In VTP version 3, both the primary and secondary servers may exist on an instance in the domain.

To enable VTP version 3, perform this task:

	Command	Purpose
Step 1	Router(config)# **vtp version 3**	Enables VTP version 3.
	Router(config)# **no vtp version**	Reverts to the default (VTP version 1).
Step 2	Router# **show vtp status**	Verifies the configuration.

This example shows one way to enable VTP version 3:

```
Router# configure terminal
Router(config)# vtp version 3
Router(config)#
```

This example shows how to verify the configuration:

```
Router# show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 3
VTP Domain Name                 : lab_switch
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0015.c724.0040

Feature VLAN:
--------------
```

```
VTP Operating Mode                 : Server
Number of existing VLANs           : 6
Number of existing extended VLANs  : 0
Configuration Revision             : 0
Primary ID                         : 0000.0000.0000
Primary Description                :
MD5 digest                         : 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
                                     0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00


Feature MST:
--------------
VTP Operating Mode                 : Transparent


Feature UNKNOWN:
--------------
VTP Operating Mode                 : Transparent
Router#
```

Configuring the VTP Mode

To configure the VTP mode, perform this task:

	Command	Purpose
Step 1	Router(config)# **vtp mode** {**client** | **server** | **transparent** | **off**} {**vlan** | **mst** | **unknown**}	Configures the VTP mode.
Step 2	Router(config)# **vtp domain** *domain-name*	(Optional for server mode) Defines the VTP domain name, which can be up to 32 characters long. VTP server mode requires a domain name. If the switch has a trunk connection to a VTP domain, the switch learns the domain name from the VTP server in the domain. **Note** You cannot clear the domain name.
Step 3	Router(config)# **end**	Exits VLAN configuration mode.
Step 4	Router# **show vtp status**	Verifies the configuration.

Note When VTP is disabled, you can enter VLAN configuration commands in configuration mode instead of the VLAN database mode and the VLAN configuration is stored in the startup configuration file.

This example shows how to configure the switch as a VTP server:

```
Router# configuration terminal
Router(config)# vtp mode server
Setting device to VTP SERVER mode.
Router(config)# vtp domain lab_network
Setting VTP domain name to lab_network
Router(config)# end
Router#
```

This example shows how to configure the switch as a VTP client:

```
Router# configuration terminal
Router(config)# vtp mode client
```

```
Setting device to VTP CLIENT mode.
Router(config)# exit
Router#
```

This example shows how to disable VTP on the switch:

```
Router# configuration terminal
Router(config)# vtp mode transparent
Setting device to VTP TRANSPARENT mode.
Router(config)# end
Router#
```

This example shows how to disable VTP on the switch and to disable VTP advertisement forwarding:

```
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# vtp mode off
Setting device to VTP OFF mode.
Router(config)# exit
Router#
```

This example shows how to verify the configuration:

```
Router# show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 3
VTP Domain Name                 : lab_network
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0015.c724.0040

Feature VLAN:
--------------
VTP Operating Mode                : Server
Number of existing VLANs          : 6
Number of existing extended VLANs : 0
Configuration Revision            : 0
Primary ID                        : 0000.0000.0000
Primary Description               :
MD5 digest                        : 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
                                    0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

Feature MST:
--------------
VTP Operating Mode                : Transparent

Feature UNKNOWN:
--------------
VTP Operating Mode                : Transparent

Router#
```

Configuring VTP Mode on a Per-Port Basis

With Release 12.2(33)SXH and later releases, you can configure VTP mode on a per-port basis. The VTP enable value will be applied only when a port becomes switched port in trunk mode. Incoming and outgoing vtp pdus are blocked; *not* forwarded. With Release 12.2(33)SXI and later releases, in VTP version 3, you can also configure VTP mode on a per-trunk basis. To configure VTP mode, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** *type*[1] *slot/port*	Selects an interface to configure.
Step 2	Router(config-if)# **vtp**	Enables VTP on the specified port.
Step 3	Router(config-if)# **end**	Exits interface configuration mode.
Step 4	Router# **show running-config interface** *type slot/port*	Verifies the change to the port.
Step 5	Router# **show vtp interface**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure VTP mode on a port:

```
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 3/5
Router(config-if)# vtp
Router(config-if)# end
Router#
```

This example shows how to disable VTP mode on a port:

```
Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 3/5
Router(config-if)# no vtp
Router(config-if)# end
Router#
```

This example shows how to verify the configuration change:

```
Router# show vtp interface gigabitethernet 3/5

 Interface                VTP Status
-----------------------------------
 GigabitEthernet3/5        disabled
Router#
```

This example shows how to verify the interface:

```
Router# show vtp interface

 Interface                VTP Status
-----------------------------------
 GigabitEthernet3/1        enabled
 GigabitEthernet3/2        enabled
 GigabitEthernet3/3        enabled
 GigabitEthernet3/4        enabled
 GigabitEthernet3/5        disabled
 GigabitEthernet3/6        enabled
 ...
```

Displaying VTP Statistics

To display VTP statistics, including VTP advertisements sent and received and VTP errors, perform this task:

Command	Purpose
`Router# **show vtp counters**`	Displays VTP statistics.

This example shows how to display VTP statistics:

```
Router# show vtp counters
VTP statistics:
Summary advertisements received    : 7
Subset advertisements received     : 5
Request advertisements received    : 0
Summary advertisements transmitted : 997
Subset advertisements transmitted  : 13
Request advertisements transmitted : 3
Number of config revision errors   : 0
Number of config digest errors     : 0
Number of V1 summary errors        : 0

VTP pruning statistics:

Trunk            Join Transmitted Join Received    Summary advts received from
                                                   non-pruning-capable device
---------------- ---------------- ---------------- ---------------------------
Fa5/8               43071            42766            5
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring VLANs

This chapter describes how to configure VLANs in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding VLANs

The following sections describe how VLANs work:

VLAN Overview

A VLAN is a group of end stations with a common set of requirements, independent of physical location. VLANs have the same attributes as a physical LAN but allow you to group end stations even if they are not located physically on the same LAN segment.

VLANs are usually associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. Traffic between VLANs must be routed. LAN port VLAN membership is assigned manually on an port-by-port basis.

VLAN Ranges

Note You must enable the extended system ID to use 4096 VLANs (see the "Understanding the Bridge ID" section on page 28-2).

Cisco IOS Release 12.2SX supports 4096 VLANs in accordance with the IEEE 802.1Q standard. These VLANs are organized into several ranges; you use each range slightly differently. Some of these VLANs are propagated to other switches in the network when you use the VLAN Trunking Protocol (VTP). The extended-range VLANs are not propagated, so you must configure extended-range VLANs manually on each network device.

Table 23-1 describes the VLAN ranges.

Table 23-1 VLAN Ranges

VLANs	Range	Usage	Propagated by VTP
0, 4095	Reserved	For system use only. You cannot see or use these VLANs.	—
1	Normal	Cisco default. You can use this VLAN but you cannot delete it.	Yes
2–1001	Normal	For Ethernet VLANs; you can create, use, and delete these VLANs.	Yes
1002–1005	Normal	Cisco defaults for FDDI and Token Ring. You cannot delete VLANs 1002–1005.	Yes
1006–4094	Extended	For Ethernet VLANs only.	No

The following information applies to VLAN ranges:

- Layer 3 LAN ports, WAN interfaces and subinterfaces, and some software features use internal VLANs in the extended range. You cannot use an extended range VLAN that has been allocated for internal use.
- To display the VLANs used internally, enter the **show vlan internal usage** command. With earlier releases, enter the **show vlan internal usage** and **show cwan vlans** commands.
- You can configure ascending internal VLAN allocation (from 1006 and up) or descending internal VLAN allocation (from 4094 and down).
- You must enable the extended system ID to use extended range VLANs (see the "Understanding the Bridge ID" section on page 28-2).

VLAN Configuration Guidelines and Restrictions

When creating and modifying VLANs in your network, follow these guidelines and restrictions:

- VLANs support a number of parameters that are not discussed in detail in this section. For complete information, see the Cisco IOS Master Command List publication.
- If the switch is in VTP server or transparent mode (see the "Configuring VTP" section on page 22-10), you can configure VLANs in global and config-vlan configuration modes. When you configure VLANs in global and config-vlan configuration modes, the VLAN configuration is saved in the vlan.dat files. To display the VLAN configuration, enter the **show vlan** command.

  If the switch is in VLAN transparent mode, use the copy **running-config startup-config** command to save the VLAN configuration to the startup-config file. After you save the running configuration as the startup configuration, use the **show running-config** and **show startup-config** commands to display the VLAN configuration.
- When the switch boots, if the VTP domain name and the VTP mode in the startup-config file and vlan.dat files do not match, the switch uses the configuration in the vlan.dat file.
- You can configure extended-range VLANs only in global configuration mode.
- Supervisor engine redundancy does not support nondefault VLAN data file names or locations. Do not enter the **vtp file** *file_name* command on a switch that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- Before you can create a VLAN, the switch must be in VTP server mode or VTP transparent mode. For information on configuring VTP, see Chapter 22, "Configuring VTP."
- The VLAN configuration is stored in the vlan.dat file, which is stored in nonvolatile memory. You can cause inconsistency in the VLAN database if you manually delete the vlan.dat file. If you want to modify the VLAN configuration or VTP, use the commands described in this guide and in the Cisco IOS Master Command List, publication.
- To do a complete backup of your configuration, include the vlan.dat file in the backup.

Configuring VLANs

These sections describe how to configure VLANs:

Configurable VLAN Parameters

Note
- Ethernet VLAN 1 uses only default values.
- Except for the VLAN name, Ethernet VLANs 1006 through 4094 use only default values.
- You can configure the VLAN name for Ethernet VLANs 1006 through 4094.

You can configure the following parameters for VLANs 2 through 1001:

- VLAN name
- VLAN type (Ethernet, FDDI, FDDI network entity title [NET], TrBRF, or TrCRF)
- VLAN state (active or suspended)
- Security Association Identifier (SAID)
- Bridge identification number for TrBRF VLANs
- Ring number for FDDI and TrCRF VLANs
- Parent VLAN number for TrCRF VLANs
- Spanning Tree Protocol (STP) type for TrCRF VLANs

Ethernet VLAN Default Parameters

- VLAN ID: 1; range: 1–4094
- VLAN name:
  - VLAN 1: “default”
  - Other VLANs: “VLAN*vlan_ID*”
- 802.10 SAID: 10*vlan_ID*; range: 100001–104094
- MTU size: 1500; range: 1500–18190
- Translational bridge 1: 0; range: 0–1005
- Translational bridge 2: 0; range: 0–1005
- VLAN state: active: active, suspend
- Pruning eligibility:
  - VLANs 2–1001 are pruning eligible
  - VLANs 1006–4094 are not pruning eligible

VLAN Locking

Release 12.2(33)SXH and later releases support the VLAN locking feature, which provides an extra level of verification to ensure that you have configured the intended VLAN.

When VLAN locking is enabled, you need to specify the VLAN name when you change a port from one VLAN to another. This feature affects **switchport** commands (in interface configuration mode) that specify the VLANs or private VLANs for access and trunk ports.

For additional information about how to configure access and trunk ports with VLAN locking enabled, see the "Configuring LAN Interfaces for Layer 2 Switching" section on page 17-6.

For additional information about how to configure ports in private VLANs with VLAN locking enabled, see the "Configuring Private VLANs" section on page 24-11.

By default, the VLAN locking is disabled. To enable VLAN locking, perform this task:

	Command	Purpose
Step 1	`Router(config)# vlan port provisioning`	Enables VLAN locking.
Step 2	`Router# show vlan port provisioning`	Verifies the VLAN locking status (enabled or disabled).

Creating or Modifying an Ethernet VLAN

User-configured VLANs have unique IDs from 1 to 4094, except for reserved VLANs (see Table 23-1 on page 23-2). Enter the **vlan** command with an unused ID to create a VLAN. Enter the **vlan** command for an existing VLAN to modify the VLAN (you cannot modify an existing VLAN that is being used by a Layer 3 port or a software feature).

See the "Ethernet VLAN Default Parameters" section on page 23-4 for the list of default parameters that are assigned when you create a VLAN. If you do not specify the VLAN type with the **media** keyword, the VLAN is an Ethernet VLAN.

To create or modify a VLAN, perform this task:

	Command	Purpose

Step 1	`Router# configure terminal`
	or
	`Router# vlan database`
		Enters VLAN configuration mode.

Step 2	`Router(config)# vlan vlan_ID{[-vlan_ID]|[,vlan_ID])`
	`Router(config-vlan)#`
	or
	`Router(vlan)# vlan vlan_ID`
		Creates or modifies an Ethernet VLAN, a range of Ethernet VLANs, or several Ethernet VLANs specified in a comma-separated list (do not enter space characters).

Step 3	`Router(config-vlan)# end`
	or
	`Router(vlan)# exit`
		Updates the VLAN database and returns to privileged EXEC mode.

Step 4	`Router# show vlan [id	name] vlan`	Verifies the VLAN configuration.

When you create or modify an Ethernet VLAN, note the following information:

- Because Layer 3 ports and some software features require internal VLANs allocated from 1006 and up, configure extended-range VLANs starting with 4094.
- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode.
- Layer 3 ports and some software features use extended-range VLANs. If the VLAN you are trying to create or modify is being used by a Layer 3 port or a software feature, the switch displays a message and does not modify the VLAN configuration.

When deleting VLANs, note the following information:

- You cannot delete the default VLANs for the different media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005.
- When you delete a VLAN, any LAN ports configured as access ports assigned to that VLAN become inactive. The ports remain associated with the VLAN (and inactive) until you assign them to a new VLAN.

This example shows how to create an Ethernet VLAN in global configuration mode and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 3
Router(config-vlan)# end
Router# show vlan id 3

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
3    VLAN0003                         active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
3    enet  100003     1500  -      -      -        -    -        0      0

Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
```

This example shows how to create an Ethernet VLAN in VLAN database mode:

```
Router# vlan database
Router(vlan)# vlan 3
VLAN 3 added:
    Name: VLAN0003
Router(vlan)# exit
APPLY completed.
Exiting....
```

This example shows how to verify the configuration:

```
Router# show vlan name VLAN0003
VLAN Name                             Status    Ports
---- -------------------------------- --------- ---------------------
3    VLAN0003                         active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- ------ ------
3    enet  100003     1500  -      -      -        -    0      0
Router#
```

Assigning a Layer 2 LAN Interface to a VLAN

A VLAN created in a management domain remains unused until you assign one or more LAN ports to the VLAN.

Note Make sure you assign LAN ports to a VLAN of the appropriate type. Assign Ethernet ports to Ethernet-type VLANs.

To assign one or more LAN ports to a VLAN, complete the procedures in the "Configuring LAN Interfaces for Layer 2 Switching" section on page 17-6.

Configuring the Internal VLAN Allocation Policy

For more information about VLAN allocation, see the "VLAN Ranges" section on page 23-2.

> **Note** The internal VLAN allocation policy is applied only following a reload.

To configure the internal VLAN allocation policy, perform this task:

	Command	Purpose
Step 1	`Router(config)# vlan internal allocation policy {ascending | descending}`	Configures the internal VLAN allocation policy.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# reload`	Applies the new internal VLAN allocation policy. **Caution** You do not need to enter the **reload** command immediately. Enter the **reload** command during a planned maintenance window.

When you configure the internal VLAN allocation policy, note the following information:

- Enter the **ascending** keyword to allocate internal VLANs from 1006 and up.
- Enter the **descending** keyword to allocate internal VLAN from 4094 and down.

This example shows how to configure descending as the internal VLAN allocation policy:

```
Router# configure terminal
Router(config)# vlan internal allocation policy descending
```

Configuring VLAN Translation

On trunk ports, you can translate one VLAN number to another VLAN number, which transfers all traffic received in one VLAN to the other VLAN.

These sections describe VLAN translation:

- VLAN Translation Guidelines and Restrictions, page 23-8
- Configuring VLAN Translation on a Trunk Port, page 23-9
- Enabling VLAN Translation on Other Ports in a Port Group, page 23-10

> **Note** To avoid spanning tree loops, be careful not to misconfigure the VLAN translation feature.

VLAN Translation Guidelines and Restrictions

When translating VLANs, follow these guidelines and restrictions:

- A VLAN translation configuration is inactive if it is applied to ports that are not Layer 2 trunks.
- Do not configure translation of ingress native VLAN traffic on an 802.1Q trunk. Because 802.1Q native VLAN traffic is untagged, it cannot be recognized for translation. You can translate traffic from other VLANs to the native VLAN of an 802.1Q trunk.
- Do not remove the VLAN to which you are translating from the trunk.
- The VLAN translation configuration applies to all ports in a port group. VLAN translation is disabled by default on all ports in a port group. Enable VLAN translation on ports as needed.
- For the modules that support VLAN translation, Table 23-2 lists:
  - The port groups to which VLAN translation configuration applies
  - The number of VLAN translations supported by the port groups
  - The trunk types supported by the modules

Table 23-2 Module Support for VLAN Translation

Product Number	Number of Ports	Number of Port Groups	Port Ranges per Port Group	Translations per Port Group	VLAN Translation Trunk-Type Support
VS-S720-10G-3CXL VS-S720-10G-3C	5	5	1 port in each group	16	802.1Q
WS-SUP720-3BXL WS-SUP720-3B WS-SUP720	2	1	1–2	32	802.1Q
WS-SUP32-10GE	3	2	1, 2–3	16	ISL 802.1Q
WS-SUP32-GE	9	1	1–9	16	ISL 802.1Q
WS-X6716-10T	16	16	1 port in each group	16	802.1Q
WS-X6716-10GE	16	16	1 port in each group	16	802.1Q
WS-X6708-10GE	8	8	1 port in each group	16	ISL 802.1Q
WS-X6704-10GE	4	4	1 port in each group	128	ISL 802.1Q
WS-X6502-10GE	1	1	1 port in 1 group	32	802.1Q
WS-X6724-SFP	24	2	1–12 13–24	128	ISL 802.1Q
WS-X6816-GBIC	16	2	1–8 9–16	32	802.1Q
WS-X6516A-GBIC	16	2	1–8 9–16	32	802.1Q
WS-X6516-GBIC	16	2	1–8 9–16	32	802.1Q

Table 23-2 Module Support for VLAN Translation (continued)

Product Number	Number of Ports	Number of Port Groups	Port Ranges per Port Group	Translations per Port Group	VLAN Translation Trunk-Type Support
WS-X6748-GE-SFP	48	4	1–23, odd 2–24, even 25–47, odd 26–48, even	128	ISL 802.1Q
WS-X6748-GE-TX	48	4	1–12 13–24 25–36 37–48	128	ISL 802.1Q
WS-X6516-GE-TX	16	2	1–8 9–16	32	802.1Q
WS-X6524-100FX-MM	24	1	1–24	32	ISL 802.1Q
WS-X6548-RJ-45	48	1	1–48	32	ISL 802.1Q
WS-X6548-RJ-21	48	1	1–48	32	ISL 802.1Q

Note To configure a port as a trunk, see the "Configuring a Layer 2 Switching Port as a Trunk" section on page 17-10.

Configuring VLAN Translation on a Trunk Port

To translate VLANs on a trunk port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects the Layer 2 trunk port to configure.
Step 2	`Router(config-if)# switchport vlan mapping enable`	Enables VLAN translation.
Step 3	`Router(config-if)# switchport vlan mapping original_vlan_ID translated_vlan_ID`	Translates a VLAN to another VLAN. The valid range is 1 to 4094. When you configure a VLAN mapping from the original VLAN to the translated VLAN on a port, traffic arriving on the original VLAN gets mapped or translated to the translated VLAN at the ingress of the switch port, and the traffic internally tagged with the translated VLAN gets mapped to the original VLAN before leaving the switch port. This method of VLAN mapping is a two-way mapping.
Step 4	`Router(config-if)# end`	Exits configuration mode.
Step 5	`Router# show interface type[1] slot/port vlan mapping`	Verifies the VLAN mapping.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to map VLAN 1649 to VLAN 755 Gigabit Ethernet port 5/2:

```
Router# configure terminal
Router(config)# interface gigabitethernet 5/2
Router(config-if)# switchport vlan mapping 1649 755
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show interface gigabitethernet 5/2 vlan mapping
State: enabled
Original VLAN Translated VLAN
------------- ---------------
  1649           755
```

Enabling VLAN Translation on Other Ports in a Port Group

To enable VLAN translation on other ports in a port group, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects the LAN port to configure.
Step 2	`Router(config-if)# switchport vlan mapping enable`	Enables VLAN translation.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show interface type[1] slot/port vlan mapping`	Verifies the VLAN mapping.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable VLAN translation on a port:

```
Router# configure terminal
Router(config)# interface gigabitethernet 5/2
Router(config-if)# switchport vlan mapping enable
Router(config-if)# end
Router#
```

Mapping 802.1Q VLANs to ISL VLANs

The valid range of user-configurable ISL VLANs is 1 through 1001 and 1006 through 4094. The valid range of VLANs specified in the IEEE 802.1Q standard is 1 to 4094. You can map 802.1Q VLAN numbers to ISL VLAN numbers.

802.1Q VLANs in the range 1 through 1001 and 1006 through 4094 are automatically mapped to the corresponding ISL VLAN. 802.1Q VLAN numbers corresponding to reserved VLAN numbers must be mapped to an ISL VLAN in order to be recognized and forwarded by Cisco network devices.

These restrictions apply when mapping 802.1Q VLANs to ISL VLANs:

- You can configure up to eight 802.1Q-to-ISL VLAN mappings.
- You can only map 802.1Q VLANs to Ethernet-type ISL VLANs.
- Do not enter the native VLAN of any 802.1Q trunk in the mapping table.

- When you map an 802.1Q VLAN to an ISL VLAN, traffic on the 802.1Q VLAN corresponding to the mapped ISL VLAN is blocked. For example, if you map 802.1Q VLAN 1007 to ISL VLAN 200, traffic on 802.1Q VLAN 200 is blocked.
- VLAN mappings are local to each switch. Make sure that you configure the same VLAN mappings on all appropriate network devices.

To map an 802.1Q VLAN to an ISL VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`vlan mapping dot1q`** *`dot1q_vlan_ID`* **`isl`** *`isl_vlan_ID`*	Maps an 802.1Q VLAN to an ISL Ethernet VLAN. The valid range for *dot1q_vlan_ID* is 1001 to 4094. The valid range for *isl_vlan_ID* is the same.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.
Step 3	`Router#` **`show vlan`**	Verifies the VLAN mapping.

This example shows how to map 802.1Q VLAN 1003 to ISL VLAN 200:

```
Router# configure terminal
Router(config)# vlan mapping dot1q 1003 isl 200
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show vlan
<...output truncated...>
802.1Q Trunk Remapped VLANs:
802.1Q VLAN    ISL VLAN
-----------   -----------
   1003          200
```

Saving VLAN Information

The VLAN database is stored in the vlan.dat file. You should create a backup of the vlan.dat file in addition to backing up the running-config and startup-config files. If you replace the existing supervisor engine, copy the startup-config file as well as the vlan.dat file to restore the system. The vlan.dat file is read on bootup and you will have to reload the supervisor engine after uploading the file. To view the file location, use the **dir vlan.dat** command. To copy the file (binary), use the **copy vlan.dat tftp** command.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Private VLANs

This chapter describes how to configure private VLANs in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Private VLANs

These sections describe how private VLANs work:

Private VLAN Domains

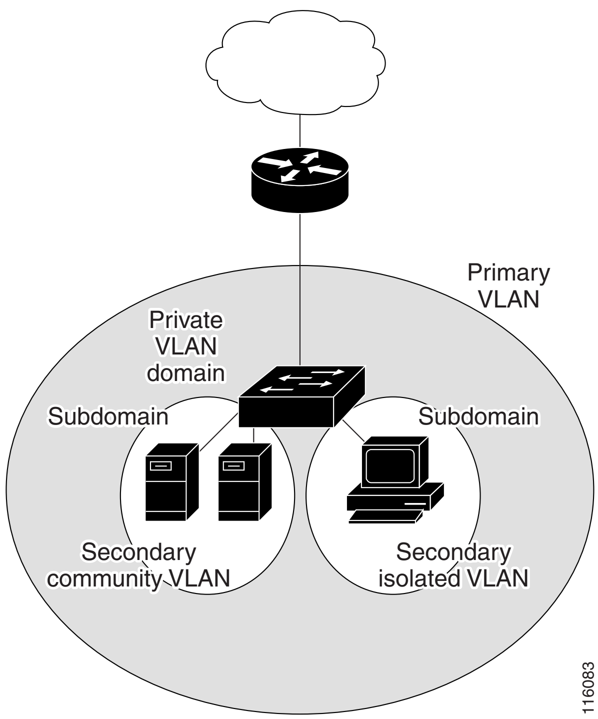
The private VLAN feature addresses two problems that service providers encounter when using VLANs:

- The switch supports up to 4096 VLANs. If a service provider assigns one VLAN per customer, the number of customers that service provider can support is limited.
- To enable IP routing, each VLAN is assigned a subnet address space or a block of addresses, which can result in wasting the unused IP addresses and creating IP address management problems.

Using private VLANs solves the scalability problem and provides IP address management benefits for service providers and Layer 2 security for customers.

The private VLAN feature partitions the Layer 2 broadcast domain of a VLAN into subdomains. A subdomain is represented by a pair of private VLANs: a primary VLAN and a secondary VLAN. A private VLAN domain can have multiple private VLAN pairs, one pair for each subdomain. All VLAN pairs in a private VLAN domain share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another (see Figure 24-1).

Figure 24-1 Private VLAN Domain



A private VLAN domain has only one primary VLAN. Every port in a private VLAN domain is a member of the primary VLAN. In other words, the primary VLAN is the entire private VLAN domain.

Secondary VLANs provide Layer 2 isolation between ports within the same private VLAN domain. There are two types of secondary VLANs:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other communities at the Layer 2 level.

Private VLAN Ports

There are three types of private VLAN ports:

- Promiscuous—A promiscuous port belongs to the primary VLAN and can communicate with all interfaces, including the community and isolated host ports that belong to the secondary VLANs that are associated with the primary VLAN.
- Isolated—An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete Layer 2 isolation from other ports within the same private VLAN domain, except for the promiscuous ports. Private VLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports.
- Community—A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with promiscuous ports. These interfaces are isolated at Layer 2 from all other interfaces in other communities and from isolated ports within their private VLAN domain.

Note Because trunks can support the VLANs carrying traffic between isolated, community, and promiscuous ports, isolated and community port traffic might enter or leave the switch through a trunk interface.

Primary, Isolated, and Community VLANs

Primary VLANs and the two types of secondary VLANs, isolated VLANs and community VLANs, have these characteristics:

- Primary VLAN— The primary VLAN carries unidirectional traffic downstream from the promiscuous ports to the (isolated and community) host ports and to other promiscuous ports.
- Isolated VLAN —A private VLAN domain has only one isolated VLAN. An isolated VLAN is a secondary VLAN that carries unidirectional traffic upstream from the hosts toward the promiscuous ports and the gateway.
- Community VLAN—A community VLAN is a secondary VLAN that carries upstream traffic from the community ports to the promiscuous port gateways and to other host ports in the same community. You can configure multiple community VLANs in a private VLAN domain.

A promiscuous port can serve only one primary VLAN, one isolated VLAN, and multiple community VLANs. Layer 3 gateways are connected typically to the switch through a promiscuous port. With a promiscuous port, you can connect a wide range of devices as access points to a private VLAN. For example, you can use a promiscuous port to monitor or back up all the private VLAN servers from an administration workstation.

In a switched environment, you can assign an individual private VLAN and associated IP subnet to each individual or common group of end stations. The end stations need to communicate only with a default gateway to communicate outside the private VLAN.

Private VLAN Port Isolation

You can use private VLANs to control access to end stations in these ways:

- Configure selected interfaces connected to end stations as isolated ports to prevent any communication at Layer 2. For example, if the end stations are servers, this configuration prevents Layer 2 communication between the servers.
- Configure interfaces connected to default gateways and selected end stations (for example, backup servers) as promiscuous ports to allow all end stations access to a default gateway.

You can extend private VLANs across multiple devices by trunking the primary, isolated, and community VLANs to other devices that support private VLANs. To maintain the security of your private VLAN configuration and to avoid other use of the VLANs configured as private VLANs, configure private VLANs on all intermediate devices, including devices that have no private VLAN ports.

IP Addressing Scheme with Private VLANs

When you assign a separate VLAN to each customer, an inefficient IP addressing scheme is created as follows:

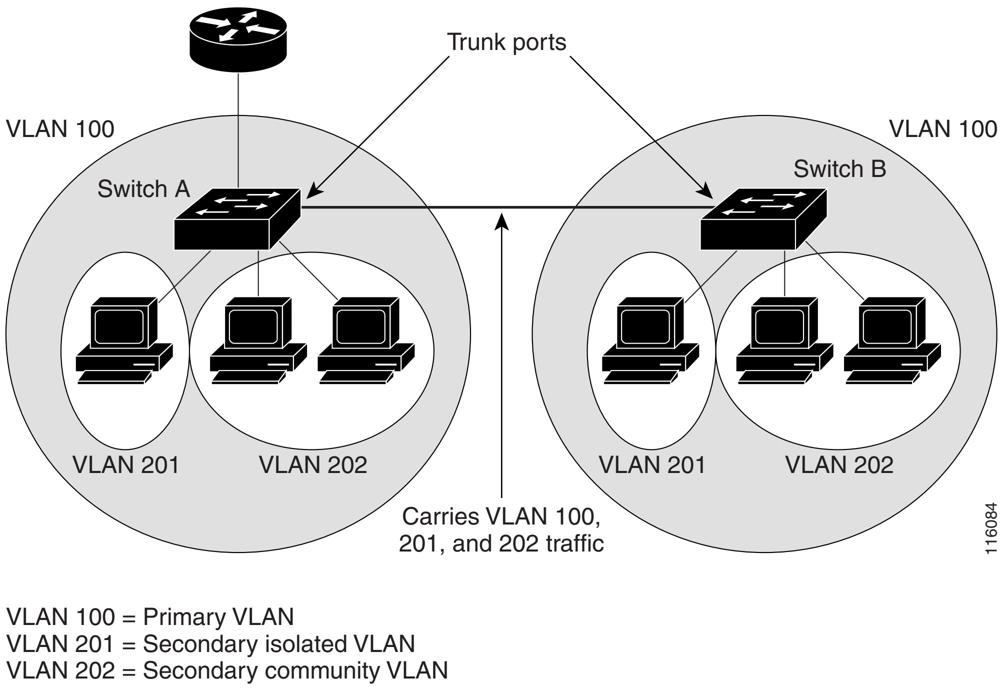
- Assigning a block of addresses to a customer VLAN can result in unused IP addresses.
- If the number of devices in the VLAN increases, the number of assigned addresses might not be large enough to accommodate them.

These problems are reduced by using private VLANs, where all members in the private VLAN share a common address space, which is allocated to the primary VLAN. Hosts are connected to secondary VLANs, and the DHCP server assigns them IP addresses from the block of addresses allocated to the primary VLAN. Subsequent IP addresses can be assigned to customer devices in different secondary VLANs, but in the same primary VLAN. When new devices are added, the DHCP server assigns them the next available address from a large pool of subnet addresses.

Private VLANs Across Multiple Switches

As with regular VLANs, private VLANs can span multiple switches. A trunk port carries the primary VLAN and secondary VLANs to a neighboring switch. The trunk port deals with the private VLAN as any other VLAN. A feature of private VLANs across multiple switches is that traffic from an isolated port in switch A does not reach an isolated port on Switch B. (See Figure 24-2.)

Figure 24-2 Private VLANs Across Switches



Because VTP versions 1 and 2 do not support private VLANs, you must manually configure private VLANs on all switches in the Layer 2 network. If you do not configure the primary and secondary VLAN association in some switches in the network, the Layer 2 databases in these switches are not merged. This situation can result in unnecessary flooding of private VLAN traffic on those switches.

VTP version 3 does support private VLANs, so you do not need to manually configure private VLANs on all switches in the Layer 2 network.

Private VLAN Interaction with Other Features

These sections describe how private VLANs interact with some other features:

- Private VLANs and Unicast, Broadcast, and Multicast Traffic, page 24-6
- Private VLANs and SVIs, page 24-6

See also the "Private VLAN Configuration Guidelines and Restrictions" section on page 24-6.

Private VLANs and Unicast, Broadcast, and Multicast Traffic

In regular VLANs, devices in the same VLAN can communicate with each other at the Layer 2 level, but devices connected to interfaces in different VLANs must communicate at the Layer 3 level. In private VLANs, the promiscuous ports are members of the primary VLAN, while the host ports belong to secondary VLANs. Because the secondary VLAN is associated to the primary VLAN, members of the these VLANs can communicate with each other at the Layer 2 level.

In a regular VLAN, broadcasts are forwarded to all ports in that VLAN. Private VLAN broadcast forwarding depends on the port sending the broadcast:

- An isolated port sends a broadcast only to the promiscuous ports or trunk ports.
- A community port sends a broadcast to all promiscuous ports, trunk ports, and ports in the same community VLAN.
- A promiscuous port sends a broadcast to all ports in the private VLAN (other promiscuous ports, trunk ports, isolated ports, and community ports).

Multicast traffic is routed or bridged across private VLAN boundaries and within a single community VLAN. Multicast traffic is not forwarded between ports in the same isolated VLAN or between ports in different secondary VLANs.

Private VLANs and SVIs

A switch virtual interface (SVI) is the Layer 3 interface of a Layer 2 VLAN. Layer 3 devices communicate with a private VLAN only through the primary VLAN and not through secondary VLANs. Configure Layer 3 VLAN SVIs only for primary VLANs. Do not configure Layer 3 VLAN interfaces for secondary VLANs. SVIs for secondary VLANs are inactive while the VLAN is configured as a secondary VLAN.

- If you try to configure a VLAN with an active SVI as a secondary VLAN, the configuration is not allowed until you disable the SVI.
- If you try to create an SVI on a VLAN that is configured as a secondary VLAN, and the secondary VLAN is already mapped at Layer 3, the SVI is not created, and an error is returned. If the SVI is not mapped at Layer 3, the SVI is created, but it is automatically shut down.

When the primary VLAN is associated with and mapped to the secondary VLAN, any configuration on the primary VLAN is propagated to the secondary VLAN SVIs. For example, if you assign an IP subnet to the primary VLAN SVI, this subnet is the IP subnet address of the entire private VLAN.

Private VLAN Configuration Guidelines and Restrictions

The guidelines for configuring private VLANs are described in the following sections:

- Secondary and Primary VLAN Configuration, page 24-7
- Private VLAN Port Configuration, page 24-9
- Limitations with Other Features, page 24-9

Secondary and Primary VLAN Configuration

When configuring private VLANs consider these guidelines:

- After you configure a private VLAN and set VTP to transparent mode, you are not allowed to change the VTP mode to client or server. For information about VTP, see Chapter 22, "Configuring VTP."
- You must use VLAN configuration (config-vlan) mode to configure private VLANs. You cannot configure private VLANs in VLAN database configuration mode. For more information about VLAN configuration, see "Configurable VLAN Parameters" section on page 23-4.
- After you have configured private VLANs, use the **copy running-config startup config** privileged EXEC command to save the VTP transparent mode configuration and private VLAN configuration in the startup-config file. If the switch resets it must default to VTP transparent mode to support private VLANs.
- In VTP versions 1 and 2, VTP does not propagate a private VLAN configuration and you must configure private VLANs on each device where you want private VLAN ports. In VTP version 3, VTP does propagate private VLAN configurations automatically.
- You cannot configure VLAN 1 or VLANs 1002 to 1005 as primary or secondary VLANs. Extended VLANs (VLAN IDs 1006 to 4094) cannot belong to private VLANs. Only Ethernet VLANs can be private VLANs.
- A primary VLAN can have one isolated VLAN and multiple community VLANs associated with it. An isolated or community VLAN can have only one primary VLAN associated with it.
- When a secondary VLAN is associated with the primary VLAN, the STP parameters of the primary VLAN, such as bridge priorities, are propagated to the secondary VLAN. However, STP parameters do not necessarily propagate to other devices. You should manually check the STP configuration to ensure that the primary, isolated, and community VLANs' spanning tree topologies match so that the VLANs can properly share the same forwarding database.
- If you enable MAC address reduction on the switch, we recommend that you enable MAC address reduction on all the devices in your network to ensure that the STP topologies of the private VLANs match.
- In a network where private VLANs are configured, if you enable MAC address reduction on some devices and disable it on others (mixed environment), use the default bridge priorities to make sure that the root bridge is common to the primary VLAN and to all its associated isolated and community VLANs. Be consistent with the ranges employed by the MAC address reduction feature regardless of whether it is enabled on the system. MAC address reduction allows only discrete levels and uses all intermediate values internally as a range. You should disable a root bridge with private VLANs and MAC address reduction, and configure the root bridge with any priority higher than the highest priority range used by any nonroot bridge.
- You cannot apply VACLs to secondary VLANs. (See Chapter 51, "Configuring Port ACLs and VLAN ACLs".)
- You can enable DHCP snooping on private VLANs. When you enable DHCP snooping on the primary VLAN, it is propagated to the secondary VLANs. If you configure DHCP on a secondary VLAN, the configuration does not take effect if the primary VLAN is already configured.
- We recommend that you prune the private VLANs from the trunks on devices that carry no traffic in the private VLANs.
- You can apply different quality of service (QoS) configurations to primary, isolated, and community VLANs. (See Chapter 43, "Configuring PFC QoS".)

- When you configure private VLANs, sticky Address Resolution Protocol (ARP) is enabled by default, and ARP entries learned on Layer 3 private VLAN interfaces are sticky ARP entries. For security reasons, private VLAN port sticky ARP entries do not age out. For information about configuring sticky ARP, see the "Configuring Sticky ARP" section on page 52-18.
- We recommend that you display and verify private VLAN interface ARP entries.
- Sticky ARP prevents MAC address spoofing by ensuring that ARP entries (IP address, MAC address, and source VLAN) do not age out. You can configure sticky ARP on a per-interface basis. For information about configuring sticky ARP, see the "Configuring Sticky ARP" section on page 52-18. The following guidelines and restrictions apply to private VLAN sticky ARP:
  - ARP entries learned on Layer 3 private VLAN interfaces are sticky ARP entries.
  - Connecting a device with a different MAC address but with the same IP address generates a message and the ARP entry is not created.
  - Because the private VLAN port sticky ARP entries do not age out, you must manually remove private VLAN port ARP entries if a MAC address changes. You can add or remove private VLAN ARP entries manually as follows:

```
Router(config)# no arp 11.1.3.30
IP ARP:Deleting Sticky ARP entry 11.1.3.30

Router(config)# arp 11.1.3.30 0000.5403.2356 arpa
IP ARP:Overwriting Sticky ARP entry 11.1.3.30, hw:00d0.bb09.266e by
hw:0000.5403.2356
```

- You can configure VLAN maps on primary and secondary VLANs. (See the "Applying a VLAN Access Map" section on page 51-15.) However, we recommend that you configure the same VLAN maps on private VLAN primary and secondary VLANs.
- When a frame is Layer 2 forwarded within a private VLAN, the same VLAN map is applied at the ingress side and at the egress side. When a frame is routed from inside a private VLAN to an external port, the private VLAN map is applied at the ingress side.
  - For frames going upstream from a host port to a promiscuous port, the VLAN map configured on the secondary VLAN is applied.
  - For frames going downstream from a promiscuous port to a host port, the VLAN map configured on the primary VLAN is applied.

  To filter out specific IP traffic for a private VLAN, you should apply the VLAN map to both the primary and secondary VLANs.
- To apply Cisco IOS output ACLs to all outgoing private VLAN traffic, configure them on the Layer 3 VLAN interface of the primary VLAN. (See Chapter 47, "Configuring Network Security".)
- Cisco IOS ACLs applied to the Layer 3 VLAN interface of a primary VLAN automatically apply to the associated isolated and community VLANs.
- Do not apply Cisco IOS ACLs to isolated or community VLANs. Cisco IOS ACL configuration applied to isolated and community VLANs is inactive while the VLANs are part of the private VLAN configuration.
- Although private VLANs provide host isolation at Layer 2, hosts can communicate with each other at Layer 3.

- Private VLANs support these Switched Port Analyzer (SPAN) features:
  - You can configure a private VLAN port as a SPAN source port.
  - You can use VLAN-based SPAN (VSPAN) on primary, isolated, and community VLANs or use SPAN on only one VLAN to separately monitor egress or ingress traffic.
  - For more information about SPAN, see Chapter 68, "Configuring Local SPAN, RSPAN, and ERSPAN."

Private VLAN Port Configuration

When configuring private VLAN ports follow these guidelines:

- Use only the private VLAN configuration commands to assign ports to primary, isolated, or community VLANs. Layer 2 access ports assigned to the VLANs that you configure as primary, isolated, or community VLANs are inactive while the VLAN is part of the private VLAN configuration. Layer 2 trunk interfaces remain in the STP forwarding state.
- Do not configure ports that belong to a PAgP or LACP EtherChannel as private VLAN ports. While a port is part of the private VLAN configuration, any EtherChannel configuration for it is inactive.
- Enable PortFast and BPDU guard on isolated and community host ports to prevent STP loops due to misconfigurations and to speed up STP convergence. (See Chapter 29, "Configuring Optional STP Features".) When enabled, STP applies the BPDU guard feature to all PortFast-configured Layer 2 LAN ports. Do not enable PortFast and BPDU guard on promiscuous ports.
- If you delete a VLAN used in the private VLAN configuration, the private VLAN ports associated with the VLAN become inactive.
- Private VLAN ports can be on different network devices if the devices are trunk-connected and the primary and secondary VLANs have not been removed from the trunk.
- All primary, isolated, and community VLANs associated within a private VLAN must maintain the same topology across trunks. You are highly recommended to configure the same STP bridge parameters and trunk port parameters on all associated VLANs in order to maintain the same topology.

Limitations with Other Features

When configuring private VLANs, consider these configuration limitations with other features:

Note In some cases, the configuration is accepted with no error messages, but the commands have no effect.

- VTP version 3 is not supported on private VLAN (PVLAN) ports.
- Do not configure fallback bridging on switches with private VLANs.
- A port is only affected by the private VLAN feature if it is currently in private VLAN mode and its private VLAN configuration indicates that it is a primary, isolated, or community port. If a port is in any other mode, such as Dynamic Trunking Protocol (DTP), it does not function as a private port.
- Do not configure private VLAN ports on interfaces configured for these other features:
  - Port Aggregation Protocol (PAgP)
  - Link Aggregation Control Protocol (LACP)

  - Voice VLAN
- You can configure IEEE 802.1x port-based authentication on a private VLAN port, but do not configure 802.1x with port security, voice VLAN, or per-user ACL on private VLAN ports.
- IEEE 802.1q mapping works normally. Traffic is remapped to or from dot1Q ports as configured, as if received from the ISL VLANs.
- Do not configure a remote SPAN (RSPAN) VLAN as a private VLAN primary or secondary VLAN. For more information about SPAN, see Chapter 68, "Configuring Local SPAN, RSPAN, and ERSPAN."
- A private VLAN host or promiscuous port cannot be a SPAN destination port. If you configure a SPAN destination port as a private VLAN port, the port becomes inactive.
- A destination SPAN port should not be an isolated port. (However, a source SPAN port can be an isolated port.) VSPAN could be configured to span both primary and secondary VLANs or, alternatively, to span either one if the user is interested only in ingress or egress traffic.
- If using the shortcuts between different VLANs (if any of these VLANs is private) consider both primary and isolated and community VLANs. The primary VLAN should be used both as the destination and as the virtual source, because the secondary VLAN (the real source) is always remapped to the primary VLAN in the Layer 2 FID table.
- If you configure a static MAC address on a promiscuous port in the primary VLAN, you must add the same static address to all associated secondary VLANs. If you configure a static MAC address on a host port in a secondary VLAN, you must add the same static MAC address to the associated primary VLAN. When you delete a static MAC address from a private VLAN port, you must remove all instances of the configured MAC address from the private VLAN.

Note Dynamic MAC addresses learned in one VLAN of a private VLAN are replicated in the associated VLANs. For example, a MAC address learned in a secondary VLAN is replicated in the primary VLAN. When the original dynamic MAC address is deleted or aged out, the replicated addresses are removed from the MAC address table.

- Do not configure private VLAN ports as EtherChannels. A port can be part of the private VLAN configuration, but any EtherChannel configuration for the port is inactive.
- When you enter the **shutdown** or the **no shutdown** command on the primary VLAN, the corresponding secondary VLANs also are shutdown or brought up.
- These restrictions apply when you configure groups of 12 ports as secondary ports:

  The 12-port restriction applies to these 10 Mb, 10/100 Mb, and 100 Mb Ethernet switching modules: WS-X6324-100FX, WS-X6348-RJ-45, WS-X6348-RJ-45V, WS-X6348-RJ-21V, WS-X6248-RJ-45, WS-X6248A-TEL, WS-X6248-TEL, WS-X6148-RJ-45, WS-X6148-RJ-45V, WS-X6148-45AF, WS-X6148-RJ-21, WS-X6148-RJ-21V, WS-X6148-21AF, WS-X6024-10FL-MT.

  Within groups of 12 ports (1–12, 13–24, 25–36, and 37–48), do not configure ports as isolated ports or community VLAN ports when one port within the group of 12 ports is any of these:

  - A trunk port
  - A SPAN destination port
  - A promiscuous private VLAN port
  - In releases where CSCsb44185 is resolved, a port that has been configured with the **switchport mode dynamic auto** or **switchport mode dynamic desirable** command.

If one port within the group of 12 ports is one of these ports listed and has the above properties, any isolated or community VLAN configuration for other ports within the 12 ports is inactive. To reactivate the ports, remove the isolated or community VLAN port configuration and enter the **shutdown** and **no shutdown** commands.

- These restrictions apply when you configure groups of 24 ports as secondary ports:

  In all releases, this 24-port restriction applies to the WS-X6548-GE-TX and WS-X6148-GE-TX 10/100/1000 Mb Ethernet switching modules.

  Within groups of 24 ports (1–24, 25–48), do not configure ports as isolated ports or community VLAN ports when one port within the group of 24 ports is any of these:

  – A trunk port
  – A SPAN destination port
  – A promiscuous private VLAN port
  – In releases where CSCsb44185 is resolved, a port that has been configured with the **switchport mode dynamic auto** or **switchport mode dynamic desirable** command.

  If one port within the group of 24 ports is one of these ports listed and has the above properties, any isolated or community VLAN configuration for other ports within the 24 ports is inactive. To reactivate the ports, remove the isolated or community VLAN port configuration and enter the **shutdown** and **no shutdown** commands.

Configuring Private VLANs

These sections contain configuration information:

Note If the VLAN is not defined already, the private VLAN configuration process defines it.

Configuring a VLAN as a Private VLAN

To configure a VLAN as a private VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# vlan vlan_ID`	Enters VLAN configuration submode.
Step 2	`Router(config-vlan)# private-vlan {community | isolated | primary}`	Configures a VLAN as a private VLAN.
	`Router(config-vlan)# no private-vlan {community | isolated | primary}`	Clears the private VLAN configuration. **Note** These commands do not take effect until you exit VLAN configuration submode.

	Command	Purpose
Step 3	Router(config-vlan)# **end**	Exits configuration mode.
Step 4	Router# **show vlan private-vlan** [**type**]	Verifies the configuration.

This example shows how to configure VLAN 202 as a primary VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 202
Router(config-vlan)# private-vlan primary
Router(config-vlan)# end
Router# show vlan private-vlan

Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
```

This example shows how to configure VLAN 303 as a community VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 303
Router(config-vlan)# private-vlan community
Router(config-vlan)# end
Router# show vlan private-vlan

Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
        303       community
```

This example shows how to configure VLAN 440 as an isolated VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 440
Router(config-vlan)# private-vlan isolated
Router(config-vlan)# end
Router# show vlan private-vlan

Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
        303       community
        440       isolated
```

Associating Secondary VLANs with a Primary VLAN

To associate secondary VLANs with a primary VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# **vlan** *primary_vlan_ID*	Enters VLAN configuration submode for the primary VLAN.
Step 2	Router(config-vlan)# **private-vlan association** {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Associates the secondary VLANs with the primary VLAN.
	Router(config-vlan)# **no private-vlan association**	Clears all secondary VLAN associations.

	Command	Purpose
Step 3	`Router(config-vlan)# end`	Exits VLAN configuration mode.
Step 4	`Router# show vlan private-vlan [type]`	Verifies the configuration.

When you associate secondary VLANs with a primary VLAN, note the following information:

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single private VLAN ID or a hyphenated range of private VLAN IDs.
- The *secondary_vlan_list* parameter can contain multiple community VLAN IDs.
- The *secondary_vlan_list* parameter can contain only one isolated VLAN ID.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to associate secondary VLANs with a primary VLAN.
- Use the **remove** keyword with a *secondary_vlan_list* to clear the association between secondary VLANs and a primary VLAN.
- The command does not take effect until you exit VLAN configuration submode.
- When you exit the VLAN configuration submode, only the last specified configuration takes effect.

This example shows how to associate community VLANs 303 through 307 and 309 and isolated VLAN 440 with primary VLAN 202 and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 202
Router(config-vlan)# private-vlan association 303-307,309,440
Router(config-vlan)# end
Router# show vlan private-vlan

Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202     303       community
202     304       community
202     305       community
202     306       community
202     307       community
202     309       community
202     440       isolated
        308       community
```

Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN

Note Isolated and community VLANs are both called secondary VLANs.

To map secondary VLANs to the Layer 3 VLAN interface of a primary VLAN to allow Layer 3 switching of private VLAN ingress traffic, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface vlan`** *`primary_vlan_ID`*	Enters interface configuration mode for the primary VLAN.
Step 2	`Router(config-if)#` **`private-vlan mapping`** {*`secondary_vlan_list`* | **`add`** *`secondary_vlan_list`* | **`remove`** *`secondary_vlan_list`*}	Maps the secondary VLANs to the Layer 3 VLAN interface of a primary VLAN to allow Layer 3 switching of private VLAN ingress traffic.
	`Router(config-if)#` [**`no`**] **`private-vlan mapping`**	Clears the mapping between the secondary VLANs and the primary VLAN.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show interface private-vlan mapping`**	Verifies the configuration.

When you map secondary VLANs to the Layer 3 VLAN interface of a primary VLAN, note the following information:

- The **private-vlan mapping** interface configuration command only affects private VLAN ingress traffic that is Layer 3-switched.
- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single private VLAN ID or a hyphenated range of private VLAN IDs.
- Enter a *secondary_vlan_list* parameter or use the **add** keyword with a *secondary_vlan_list* parameter to map the secondary VLANs to the primary VLAN.
- Use the **remove** keyword with a *secondary_vlan_list* parameter to clear the mapping between secondary VLANs and the primary VLAN.

This example shows how to permit routing of secondary VLAN ingress traffic from private VLANs 303 through 307, 309, and 440 and verify the configuration:

```
Router# configure terminal
Router(config)# interface vlan 202
Router(config-if)# private-vlan mapping add 303-307,309,440
Router(config-if)# end
Router# show interfaces private-vlan mapping
Interface Secondary VLAN Type
--------- -------------- -----------------
vlan202   303            community
vlan202   304            community
vlan202   305            community
vlan202   306            community
vlan202   307            community
vlan202   309            community
vlan202   440            isolated

Router#
```

Configuring a Layer 2 Interface as a Private VLAN Host Port

To configure a Layer 2 interface as a private VLAN host port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects the LAN port to configure.
Step 2	`Router(config-if)# switchport`	Configures the LAN port for Layer 2 switching. • You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional **switchport** commands with keywords. • Required only if you have not entered the **switchport** command already for the interface.
Step 3	`Router(config-if)# switchport mode private-vlan {host | promiscuous}`	Configures the Layer 2 port as a private VLAN host port.
Step 4	`Router(config-if)# switchport private-vlan host-association primary_vlan_ID secondary_vlan_ID`	Associates the Layer 2 port with a private VLAN. **Note** If VLAN locking is enabled, enter the VLAN name instead of the VLAN number. For more information, see the "VLAN Locking" section on page 23-4.
	`Router(config-if)# no switchport private-vlan host-association`	Clears the association.
Step 5	`Router(config-if)# end`	Exits configuration mode.
Step 6	`Router# show interfaces [type[1] slot/port] switchport`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure interface FastEthernet 5/1 as a private VLAN host port and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# switchport mode private-vlan host
Router(config-if)# switchport private-vlan host-association 202 303
Router(config-if)# end
Router# show interfaces fastethernet 5/1 switchport
Name: Fa5/1
Switchport: Enabled
→ Administrative Mode: private-vlan host
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
→ Administrative private-vlan host-association: 202 (VLAN0202) 303 (VLAN0303)
Administrative private-vlan mapping: none
→ Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port

To configure a Layer 2 interface as a private VLAN promiscuous port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type1 slot/port`	Selects the LAN interface to configure.
Step 2	`Router(config-if)# switchport`	Configures the LAN interface for Layer 2 switching: • You must enter the **switchport** command once without any keywords to configure the LAN interface as a Layer 2 interface before you can enter additional **switchport** commands with keywords. • Required only if you have not entered the **switchport** command already for the interface.
Step 3	`Router(config-if)# switchport mode private-vlan {host | promiscuous}`	Configures the Layer 2 port as a private VLAN promiscuous port.
Step 4	`Router(config-if)# switchport private-vlan mapping primary_vlan_ID {secondary_vlan_list | add secondary_vlan_list | remove secondary_vlan_list}`	Maps the private VLAN promiscuous port to a primary VLAN and to selected secondary VLANs. **Note** If VLAN locking is enabled, enter the VLAN name instead of the VLAN number. For more information, see the "VLAN Locking" section on page 23-4.
	`Router(config-if)# no switchport private-vlan mapping`	Clears all mapping between the private VLAN promiscuous port and the primary VLAN and any secondary VLANs.
Step 5	`Router(config-if)# end`	Exits configuration mode.
Step 6	`Router# show interfaces [type1 slot/port] switchport`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When you configure a Layer 2 interface as a private VLAN promiscuous port, note the following information:

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single private VLAN ID or a hyphenated range of private VLAN IDs.
- If VLAN locking is enabled, enter VLAN names instead of VLAN numbers in the *secondary_vlan_list*. When entering a range of VLAN names, you must leave spaces between the VLAN names and the dash.
- Enter a *secondary_vlan_list* value or use the **add** keyword with a *secondary_vlan_list* value to map the secondary VLANs to the private VLAN promiscuous port.
- Use the **remove** keyword with a *secondary_vlan_list* value to clear the mapping between secondary VLANs and the private VLAN promiscuous port.

This example shows how to configure interface FastEthernet 5/2 as a private VLAN promiscuous port and map it to a private VLAN:

```
Router# configure terminal
Router(config)# interface fastethernet 5/2
Router(config-if)# switchport mode private-vlan promiscuous
Router(config-if)# switchport private-vlan mapping 202 303,440
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show interfaces fastethernet 5/2 switchport
Name: Fa5/2
Switchport: Enabled
➔ Administrative Mode: private-vlan promiscuous
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: none ((Inactive))
➔ Administrative private-vlan mapping: 202 (VLAN0202) 303 (VLAN0303) 440 (VLAN0440)
➔ Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

Monitoring Private VLANs

Table 24-1 shows the privileged EXEC commands for monitoring private VLAN activity.

Table 24-1 Private VLAN Monitoring Commands

Command	Purpose
show interfaces status	Displays the status of interfaces, including the VLANs to which they belong.
show vlan private-vlan [**type**]	Displays the private VLAN information for the switch.
show interface switchport	Displays private VLAN configuration on interfaces.
show interface private-vlan mapping	Displays information about the private VLAN mapping for VLAN SVIs.

This is an example of the output from the **show vlan private-vlan** command:

```
Switch(config)# show vlan private-vlan
Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
10      501       isolated          Fa2/0/1, Gi3/0/1, Gi3/0/2
10      502       community         Fa2/0/11, Gi3/0/1, Gi3/0/4
10      503       non-operational
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Private Hosts

This chapter describes how to configure the private hosts feature in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Private Hosts

The sections that follow provide more detail about the following private hosts concepts:

Private Hosts Overview

Service providers typically deliver triple-play services (voice, video, and data) using three different VLANs over a single physical interface for each end user. The service infrastructure would be simpler and more scalable if the service provider could deploy a single set of VLANs to multiple end users for the same set of services, but the service provider must be able to isolate traffic between the users (hosts) at Layer 2. The private hosts feature provides this isolation, allowing VLAN sharing among multiple end users.

The private hosts feature provides these key benefits:

- Isolates traffic among hosts (subscribers) that share the same VLAN ID.
- Reuses VLAN IDs across different subscribers, which improves VLAN scalability by making better use of the 4096 VLANs allowed.
- Prevents media access control (MAC) address spoofing to prevent denial of service (DOS) attacks.

The private hosts feature uses protocol-independent port-based access control lists (PACLs) to provide Layer 2 isolation between hosts on trusted ports within a strictly Layer 2 domain. The PACLs isolate the hosts by imposing Layer 2 forwarding constraints on the switch ports.

Isolating Hosts in a VLAN

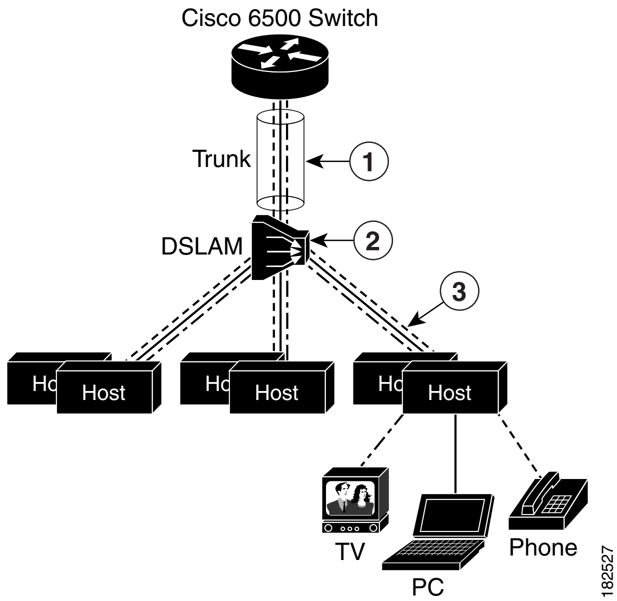
By isolating the hosts, a service provider can use a single set of VLANs to deliver the same set of broadband or metro Ethernet services to multiple end users while ensuring that none of the hosts in the VLAN can communicate directly with each other. For example, VLAN 10 can be used for voice traffic, VLAN 20 for video traffic, and VLAN 30 for data traffic.

When the switch is used as a Digital Subscriber Line Access Multiplexer (DSLAM) gigabit Ethernet aggregator, the DSLAM is connected to the switch through a trunk port that can carry data for multiple VLANs. The service provider uses a single physical port and a single set of VLANs to deliver the same set of services to different end users (isolated hosts). A separate VLAN is used for each service (voice, video, and data).

Figure 25-1 shows an example of triple-play services being delivered from the switch to multiple end users attached to a DSLAM. In the figure, note the following:

- A single trunk link between the switch and the DSLAM carries traffic for all three VLANs.
- Virtual circuits (VCs) deliver the VLAN traffic from the DSLAM to individual end users.

Figure 25-1 VC to VLAN Mapping



1	The trunk link carries: • One voice VLAN • One video VLAN • One data VLAN	**2**	The DSLAM maps voice, video, and data traffic between VLANs and VCs.
		3	Individual VCs carry voice, video, and data traffic between the DSLAM and each host.

Restricting Traffic Flow (Using Private Hosts Port Mode and PACLs)

The private hosts feature uses PACLs to restrict the type of traffic that is allowed to flow through each of the ports configured for private hosts. A port's mode (specified when you enable private hosts on the port) determines what type of PACL is applied to the port. Each type of PACL restricts the traffic flow for a different type of traffic (for example, from content servers to isolated hosts, from isolated hosts to servers, and traffic between isolated hosts).

The following list describes the port modes used by the private hosts feature (see Figure 25-2):

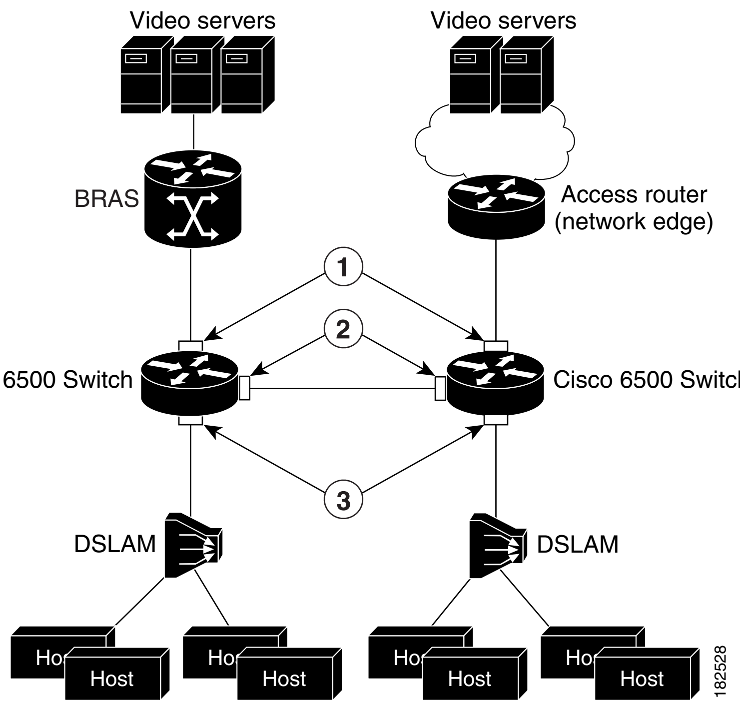
- Isolated—Ports connected to the DSLAMs that the end users (isolated hosts) are connected to. The hosts on the VLANs on these ports need to be isolated from each other. Hosts connected through these ports are allowed to pass unicast traffic to upstream devices only.
- Promiscuous—Ports that face the core network or the Broadband Remote Access Server (BRAS) devices and multicast servers that are providing the broadband services.
- Mixed—Ports that interconnect switches. These ports can function as either isolated ports or promiscuous ports, depending on Spanning Tree Protocol (STP) topology changes. These ports allow unicast traffic to upstream devices (such as a BRAS or multicast server) only.

The private hosts feature restricts traffic flow in these ways:

- Broadcast traffic at the ingress of the service provider network is redirected to BRAS and multicast servers (such as video servers).
- All unicast traffic between access switches (switches connected to each other) is blocked except for traffic directed toward a BRAS or a multicast server.
- The unknown unicast flood blocking (UUFB) feature is used to block unknown unicast traffic on DSLAM-facing ports.

Figure 25-2 shows the different types of port modes (isolated, promiscuous, and mixed) used in a private hosts configuration.

Figure 25-2 Private Hosts Port Types (Modes)



1	Promiscuous ports	Permit all traffic from a BRAS to hosts.
2	Mixed ports	Permit broadcast traffic from a BRAS. Redirect broadcast traffic from hosts to promiscuous and mixed-mode ports. Permit traffic from a BRAS to hosts and from hosts to a BRAS. Deny all other host to host traffic.
3	Isolated ports	Permit unicast traffic from host to a BRAS only; block unicast traffic between ports. Redirect all broadcast traffic from host to a BRAS. Deny traffic from a BRAS (to prevent spoofing). Permit multicast traffic (IPv4 and IPv6).

Note In this description of port types, the term BRAS represents an upstream devices such as a BRAS, a multicast server (such as a video server), or a core network device that provides access to these devices.

Port ACLs

The private hosts feature creates several types of port ACLs (PACLs) to impose Layer 2 forwarding constraints on switch ports. The software creates PACLs for the different types of private hosts ports based on the MAC addresses of the content servers providing broadband services and the VLAN IDs of the isolated hosts to deliver those services to. You specify the mode in which each private hosts port is to operate and the software applies the appropriate PACL to the port based on the port's mode (isolated, promiscuous, or mixed).

The following are examples of the different types of PACLs that are used by the private hosts feature.

Isolated Hosts PACL

This example shows a PACL for isolated ports:

```
deny host BRAS_MAC any
permit any host BRAS_MAC
redirect any host FFFF.FFFF.FFFF to LTLIndex 6
permit any 0100.5E00.0000/0000.007F.FFFF
permit any 3333.0000.0000/000.FFFF.FFFF
deny any any
```

Promiscuous Port PACL

This example shows a PACL for promiscuous ports:

```
permit host BRAS_MAC any
deny any any
```

Mixed Port PACL

This example shows a PACL for mixed ports:

```
permit host BRAS_MAC ffff.ffff.ffff
redirect any host FFFF.FFFF.FFFF to LTLIndex 6
permit host BRAS_MAC any
permit any host BRAS_MAC
deny any any
```

Configuration Guidelines and Limitations

General Restrictions

When you configure the private hosts feature, observe the following guidelines and limitations:

- The SIP-400 and Enhanced FlexWAN modules do not support private hosts.
- Private hosts and private VLANs cannot both be configured on the same port (interface). Both features can coexist on the switch, but the features must be configured on different ports.
- Private hosts is an end-to-end feature. You must enable the feature on all of the switches between the DSLAMs and an upstream device such as a BRAS or a multicast server.
- Only trusted ports can be configured as isolated ports.
- The private hosts feature is supported on Layer 2 interfaces that are configured as trunking switch ports (802.1Q or ISL trunk ports).
- The private hosts feature is supported on port-channel interfaces (EtherChannel, Fast EtherChannel, and Gigabit EtherChannel). You must enable private hosts on the port-channel interface; you cannot enable the feature on member ports.
- DAI and DHCP snooping cannot be enabled on a private hosts port unless all of the VLANs on the port are configured for snooping.

ACL Guidelines

The following configuration guidelines and limitations apply to access control lists (ACLs):

- This release of the private hosts feature uses protocol-independent MAC ACLs.

  Do not apply IP-based ACLs to any port configured for private hosts or you will defeat the private hosts feature (because the switch will not be able to apply a private hosts MAC ACL to the port).
- You can configure the following interface types for protocol-independent MAC ACL filtering:
  - VLAN interfaces with no IP address
  - Physical LAN ports that support EoMPLS
  - Logical LAN subinterfaces that support EoMPLS
- Protocol-independent MAC ACL filtering applies MAC ACLs to all ingress traffic types (for example, IPv4 traffic, IPv6 traffic, and MPLS traffic, in addition to MAC-layer traffic).
- Ingress traffic that is permitted or denied by a protocol-independent MAC ACL is processed by egress interfaces as MAC-layer traffic. You cannot apply egress IP ACLs to traffic permitted or denied by a MAC ACL on an interface configured for protocol-independent MAC ACL filtering.

- Do not configure protocol-independent MAC ACL filtering on VLAN interfaces where you have configured an IP address.
- Do not configure protocol-independent MAC ACL filtering with microflow policing when the permitted traffic would be bridged or Layer 3 switched in hardware by the PFC3.
- Protocol-independent MAC ACL filtering supports microflow policing when the permitted traffic is routed in software by the route processor (RP).
- Do not apply any VACLs or RACLs to a port configured for private hosts. To prevent any existing VLAN ACLs (VACLs) and routing ACLs (RACLs) from interfering with a PACL, configure the **access-group mode prefer port** command on the port.

VLANs on the Trunk Port

The following guidelines and limitations apply to VLANs:

- You can enable IGMP snooping on VLANs that use trunk ports configured for private hosts.
- You cannot enable IP multicast on a VLAN that uses a trunk port that is configured for private hosts.
- Because PACLs operate in override mode on trunk ports, you cannot apply VLAN-based features to switch ports.
- The Multicast VLAN Registration (MVR) feature can coexist with private hosts as long as the multicast source exists on a promiscuous port.

Interaction with Other Features

The private hosts feature interacts with other features that are configured on the switch as follows:

- Private hosts do not affect Layer 2-based services such as MAC limiting, unicast flood protection (UFP), or unknown unicast flood blocking (UUFB).
- The private hosts features does not affect IGMP snooping. However, if IGMP snooping is globally disabled, IGMP control packets will be subject to ACL checks. To permit IGMP control packets, the private hosts software adds a multicast permit statement to the PACLs for isolated hosts. This operation occurs automatically and no user intervention is required.
- Port security can be enabled on isolated ports to provide added security to those ports.
- When enabled on promiscuous or mixed-mode ports, the port security feature may restrict a change in source port for an upstream device (such as a BRAS or a multicast server).
- When enabled on an access port, 802.1X is not affected by the private hosts feature.

Spoofing Protection

The private hosts feature prevents MAC address spoofing but does not validate the customer MAC or IP address. To prevent MAC address spoofing, the private hosts feature does the following:

- Uses a static MAC address for a BRAS or a multicast server.
- Disables learning in the Layer 2 forwarding table.
- Alerts the switch software when a BRAS or multicast server moves from one source port to another. The software then validates the move and updates the Layer 2 forwarding table.

Multicast Operation

Multicast traffic that originates from an upstream device (such as a BRAS or a multicast server) is always permitted. In addition, the private hosts PACLs are not applied to multicast control packets (such as IGMP query and join requests). This operation allows isolated hosts to participate in multicast groups, respond to IGMP queries, and receive traffic from any groups of interest.

Multicast traffic that originates from a host is dropped by the private hosts PACLs. However, if other hosts need to receive multicast traffic originating from a host, the private hosts feature adds a *multicast permit* entry to the PACLs.

Configuring Private Hosts

The following sections provide information about configuring the private hosts feature in Cisco IOS Release 12.2SX:

Configuration Summary

The following is a summary of the steps to configure the private hosts feature. Detailed configuration instructions are provided in the next section.

1. Determine which switch ports (interfaces) to use for the private hosts feature.
   - You can configure the feature on switch ports (802.1Q or ISL trunk ports)
   - You can configure the feature on port-channel interfaces. Private hosts must be enabled on the port-channel interface; you cannot enable the feature on member ports.
2. Configure each port (interface) for normal, non-private hosts service. Configure the **access-group mode prefer port** command on the port. You can configure the VLANs at this step or later.
3. Determine which VLAN or set of VLANs will be used to deliver broadband services to end users. The private hosts feature will provide Layer 2 isolation among the hosts in these VLANs.
4. Identify the MAC addresses of all of the BRASs and multicast servers that are being used to provide broadband services to end users (isolated hosts).

Note If a server is not connected directly to the switch, determine the MAC address of the core network device that provides access to the server.

5. (Optional) If you plan to offer different types of broadband service to different sets of isolated hosts, create multiple MAC and VLAN lists.
   - Each MAC address list identifies a server or set of servers providing a particular type of service.
   - Each VLAN list identifies the isolated hosts to deliver that service to.
6. Configure promiscuous ports and specify a MAC and VLAN list to identify the server and receiving hosts for a particular type of service.

Note You can specify multiple MAC and VLAN combinations to allow different types of services to be delivered to different sets of hosts. For example, the BRAS at xxxx.xxxx.xxxx could be used to deliver a basic set of services over VLANs 20, 25, and 30, and the BRAS at yyyy.yyyy.yyyy could be used to deliver a premium set of services over VLANs 5, 10, and 15.

7. Globally enable private hosts.
8. Enable private hosts on individual ports (interfaces) and specify the mode in which the port is to operate. To determine port mode, you need to know if the port faces upstream (toward content servers or core network), faces downstream (toward DSLAMs), or is connected to another switch (typically, in a ring topology). See the "Restricting Traffic Flow (Using Private Hosts Port Mode and PACLs)" section on page 25-3.

After you enable the feature on individual ports, the switch is ready to run the private hosts feature. The private hosts software uses the MAC and VLAN lists you defined to create the isolated, promiscuous, and mixed-mode PACLs for your configuration. The software then applies the appropriate PACL to each private hosts port based on the port's mode.

Detailed Configuration Steps

To configure the private hosts feature, perform the following steps. These steps assume that you have already configured the Layer 2 interfaces that you plan for private hosts.

Note You can configure private hosts only on trunking switch ports (802.1Q or ISL trunk ports) or EtherChannel ports. In addition, you must enable private hosts on all of the switches between the DSLAMs and upstream devices.

	Command or Action	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`private-hosts mac-list`** *`mac-list-name mac-address`* `[`**`remark`** *`device-name`* `|` *`comment`*`]`	Creates a list of MAC addresses that identify the BRAS and multicast servers providing broadband services. • *mac-list-name* specifies a name to assign to this list of content servers. • *mac-address* identifies the BRAS or multicast server (or set of servers) providing a particular broadband service or set of services. • **remark** allows you to specify an optional device name or comment to assign to this MAC list. Specify the MAC address of every content server being used to deliver services. If you plan to offer different types of services to different sets of hosts, create a separate MAC list for each server or set of servers providing a particular service. **Note** If a server is not directly connected to the switch, specify the MAC address of the core network device that provides access to the server.

	Command or Action	Purpose
Step 3	`Router(config)# private-hosts vlan-list vlan-IDs`	Creates a list of the VLANs (*vlan-IDs*) whose hosts need to be isolated so that the hosts can receive broadband services. Create separate VLAN lists if you plan to offer particular services to different sets of hosts. Otherwise, all of the broadband services will be delivered to all isolated hosts.
Step 4	`Router(config)# private-hosts promiscuous mac-list-name [vlan-list vlan-IDs]`	Identifies the content servers for broadband services and the end users (isolated hosts) to which to deliver the services. • *mac-list-name* specifies the name of the MAC address lists that identifies the BRAS or multicast server (or set of servers) providing a particular type of broadband service or set of services. • *vlan-IDs* identifies the VLAN or set of VLANs whose hosts are to receive services from the above servers. If no VLAN list is specified, the software uses the global VLAN list (configured in Step 3). **Note** You can enter this command multiple times to configure multiple MAC and VLAN combinations, each defining the server and receiving hosts for a particular type of service.
Step 5	`Router(config)# private-hosts`	Globally enables private hosts on the switch.
Step 6	`Router(config)# interface interface`	Selects the switch port (802.1Q or ISL trunk port) or EtherChannel port to enable for private hosts.
Step 7	`Router(config-if)# access-group mode prefer port`	Specifies that any existing VACLs or RACLs on the port will be ignored.
Step 8	`Router(config-if)# private-hosts mode {promiscuous | isolated | mixed}`	Enables private hosts on the port. Use one of the following keywords to define the mode that the port is to operate in: • **promiscuous**—Upstream-facing ports that connect to broadband servers (BRAS, multicast, or video) or to core network devices providing access to the servers. • **isolated**—Ports that connect to DSLAMs. • **mixed**—Ports that connect to other switches, typically in a ring topology. **Note** You must perform this step on each port being used for private hosts.
Step 9	`Router(config-if)# end`	Exits interface and global configuration modes and returns to privileged EXEC mode. Private Hosts configuration is complete.

Configuration Examples

The following example creates a MAC address list and a VLAN list and isolates the hosts in VLANs 10, 12, 15, and 200 through 300. The BRAS-facing port is made promiscuous and two host-connected ports are made isolated:

```
Router# configure terminal
Router(config)# private-hosts mac-list BRAS_list 0000.1111.1111 remark BRAS_SanJose
Router(config)# private-hosts vlan-list 10,12,15,200-300
Router(config)# private-hosts promiscuous BRAS_list vlan-list 10,12,15,200-300
Router(config)# private-hosts
Router(config)# interface gig 4/2
Router(config-if)# private-hosts mode promiscuous
Router(config-if)# exit
Router(config)# interface gig 5/2
Router(config-if)# private-hosts mode isolated
Router(config-if)# exit
Router(config)# interface gig 5/3
Router(config-if)# private-hosts mode isolated
Router(config-if)# end
Router#
```

The following example shows the interface configuration of a private hosts isolated port:

```
Router# show run interface gig 5/2
Building configuration...

Current configuration : 200 bytes
!
interface GigabitEthernet5/2
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 access-group mode prefer port
 private-hosts mode isolated
end
```

The following example shows the interface configuration of a private hosts promiscuous port:

```
Router# show run interface gig 4/2
Building configuration...

Current configuration : 189 bytes
!
interface GigabitEthernet4/2
 switchport
 switchport access vlan 200
 switchport mode access
 private-hosts mode promiscuous
end

private-hosts
private-hosts vlan-list 200
private-hosts promiscuous bras-list
private-hosts mac-list bras-list 0000.1111.1111 remark BRAS-SERVER
```

Enabling Index Learning on Isolated Mode Ports

By default, MAC address movement (index learning) is disabled between isolated-mode ports.

When wireless access points are connected to isolated-mode ports, a wireless user can make an initial connection, but with the default private hosts configuration (index learning disabled), the MAC address of the wireless user will not be learned on any other isolated-mode ports, which prevents connection through any other wireless access point that is connected to an isolated-mode port.

To allow wireless users to move from one wireless access point to another, enable index learning on isolated mode ports, which enables MAC address movement between isolated-mode ports.

To enable index learning on the switch, perform this task:

	Command or Action	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`private-hosts index learning`**	Enables index learning on the switch.

This example shows how to enable index learning on the switch:

```
Router# configure terminal
Router(config)# private-hosts index learning
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring IEEE 802.1Q Tunneling

This chapter describes how to configure IEEE 802.1Q tunneling in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX switching modules do not support IEEE 802.1Q tunneling.

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding 802.1Q Tunneling

802.1Q tunneling enables service providers to use a single VLAN to support customers who have multiple VLANs, while preserving customer VLAN IDs and keeping traffic in different customer VLANs segregated.

A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that you dedicate to tunneling, which then becomes a tunnel VLAN. To keep customer traffic segregated, each customer requires a separate tunnel VLAN, but that one tunnel VLAN supports all of the customer's VLANs.

802.1Q tunneling is not restricted to point-to-point tunnel configurations. Any tunnel port in a tunnel VLAN is a tunnel entry and exit point. An 802.1Q tunnel can have as many tunnel ports as are needed to connect customer switches.

The customer switches are trunk connected, but with 802.1Q tunneling, the service provider switches only use one service provider VLAN to carry all the customer VLANs, instead of directly carrying all the customer VLANs.

With 802.1Q tunneling, tagged customer traffic comes from an 802.1Q trunk port on a customer device and enters the service-provider edge switch through a tunnel port. The link between the 802.1Q trunk port on a customer device and the tunnel port is called an asymmetrical link because one end is configured as an 802.1Q trunk port and the other end is configured as a tunnel port. You assign the tunnel port to an access VLAN ID unique to each customer. See Figure 26-1 on page 26-2 and Figure 26-2 on page 26-3.

Figure 26-1 IEEE 802.1Q Tunnel Ports in a Service-Provider Network

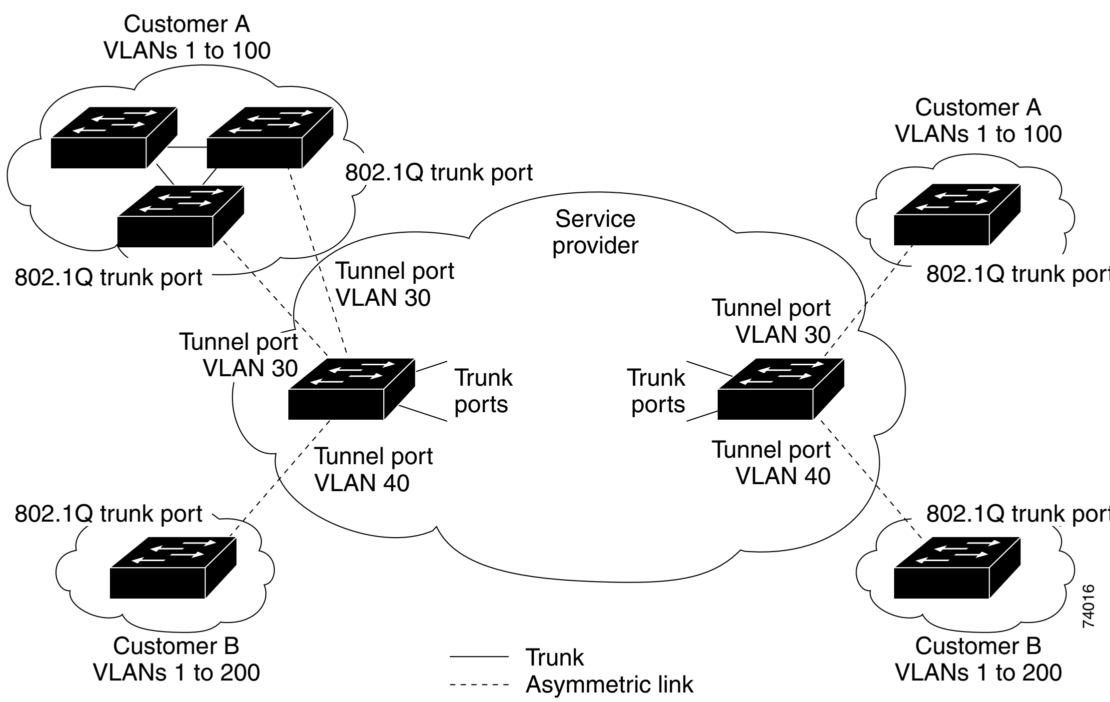
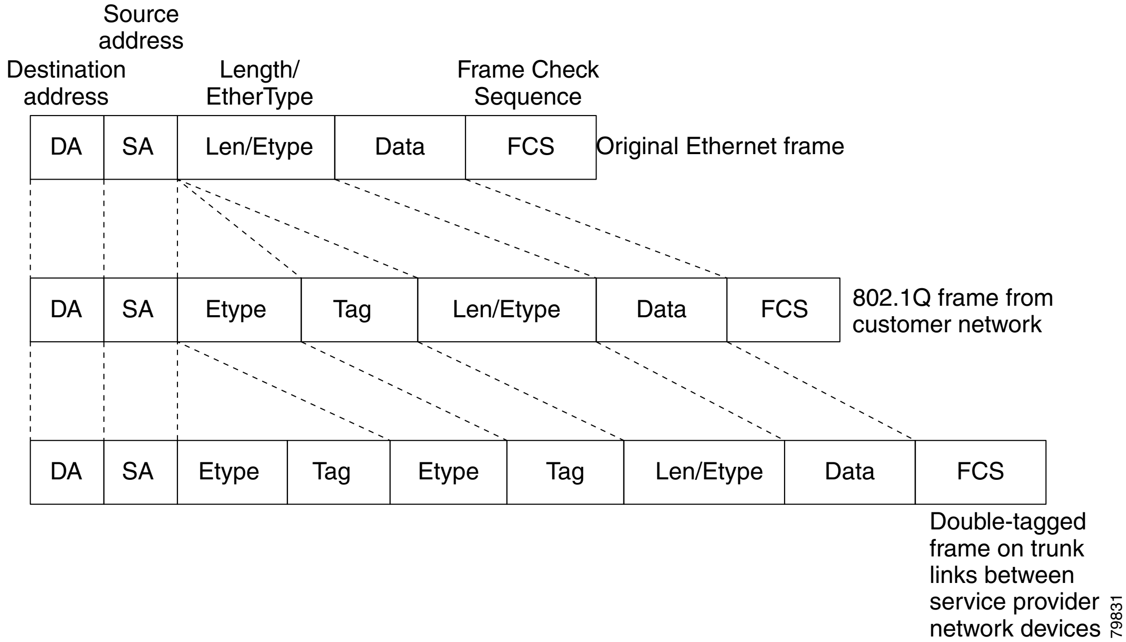


Figure 26-2 Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames



When a tunnel port receives tagged customer traffic from an 802.1Q trunk port, it does not strip the received 802.1Q tag from the frame header; instead, the tunnel port leaves the 802.1Q tag intact, adds a 2-byte Ethertype field (0x8100) followed by a 2-byte field containing the priority (CoS) and the VLAN. The received customer traffic is then put into the VLAN to which the tunnel port is assigned. This Ethertype 0x8100 traffic, with the received 802.1Q tag intact, is called tunnel traffic.

A VLAN carrying tunnel traffic is an 802.1Q tunnel. The tunnel ports in the VLAN are the tunnel's ingress and egress points.

The tunnel ports do not have to be on the same network device. The tunnel can cross other network links and other network devices before reaching the egress tunnel port. A tunnel can have as many tunnel ports as required to support the customer devices that need to communicate through the tunnel.

An egress tunnel port strips the 2-byte Ethertype field (0x8100) and the 2-byte length field and transmits the traffic with the 802.1Q tag still intact to an 802.1Q trunk port on a customer device. The 802.1Q trunk port on the customer device strips the 802.1Q tag and puts the traffic into the appropriate customer VLAN.

Note Tunnel traffic carries a second 802.1Q tag only when it is on a trunk link between service-provider network devices, with the outer tag containing the service-provider-assigned VLAN ID and the inner tag containing the customer-assigned VLAN IDs.

802.1Q Tunneling Configuration Guidelines and Restrictions

When configuring 802.1Q tunneling in your network, follow these guidelines and restrictions:

- Use asymmetrical links to put traffic into a tunnel or to remove traffic from a tunnel.
- Configure tunnel ports only to form an asymmetrical link.

- Dedicate one VLAN for each tunnel.
- Assign only tunnel ports to VLANs used for tunneling.
- Trunks require no special configuration to carry tunnel VLANs.
- Tunnel ports are not trunks. Any commands to configure trunking are inactive while the port is configured as a tunnel port.
- Tunnel ports learn customer MAC addresses.
- We recommend that you use ISL trunks to carry tunnel traffic between devices that do not have tunnel ports. Because of the 802.1Q native VLAN feature, using 802.1Q trunks requires that you be very careful when you configure tunneling: a mistake might direct tunnel traffic to a non-tunnel port.
- By default, the native VLAN traffic of a dot1q trunk is sent untagged, which cannot be double-tagged in the service provider network. Because of this situation, the native VLAN traffic might not be tunneled correctly. Be sure that the native VLAN traffic is always sent tagged in an asymmetrical link. To tag the native VLAN egress traffic and drop all untagged ingress traffic, enter the global **vlan dot1q tag native** command.
- Configure jumbo frame support on tunnel ports:
  - See the "Configuring Jumbo Frame Support" section on page 8-10.
  - Take note of the modules listed in the "Configuring Jumbo Frame Support" section that do not support jumbo frames.
- Jumbo frames can be tunneled as long as the jumbo frame length combined with the 802.1Q tag does not exceed the maximum frame size.
- Because tunnel traffic has the added ethertype and length field and retains the 802.1Q tag within the switch, the following restrictions exist:
  - The Layer 3 packet within the Layer 2 frame cannot be identified in tunnel traffic.
  - Layer 3 and higher parameters cannot be identified in tunnel traffic (for example, Layer 3 destination and source addresses).
  - Because the Layer 3 addresses cannot be identified within the packet, tunnel traffic cannot be routed.
  - The switch can provide only MAC-layer filtering for tunnel traffic (VLAN IDs and source and destination MAC addresses).
  - The switch can provide only MAC-layer access control and QoS for tunnel traffic.
  - QoS cannot detect the received CoS value in the 802.1Q 2-byte Tag Control Information field.
- On an asymmetrical link, the Cisco Discovery Protocol (CDP) reports a native VLAN mismatch if the VLAN of the tunnel port does not match the native VLAN of the 802.1Q trunk. The 802.1Q tunnel feature does not require that the VLANs match. Ignore the messages if your configuration requires nonmatching VLANs.
- Asymmetrical links do not support the Dynamic Trunking Protocol (DTP) because only one port on the link is a trunk. Configure the 802.1Q trunk port on an asymmetrical link to trunk unconditionally.
- The 802.1Q tunneling feature cannot be configured on ports configured to support private VLANs.
- The following Layer 2 protocols work between devices connected by an asymmetrical link:
  - CDP
  - UniDirectional Link Detection (UDLD)
  - Port Aggregation Protocol (PAgP)
  - Link Aggregation Control Protocol (LACP)

- Spanning-tree BPDU filtering is enabled automatically on tunnel ports.
- CDP is automatically disabled on tunnel ports.
- VLAN Trunk Protocol (VTP) does not work between the following devices:
  - Devices connected by an asymmetrical link
  - Devices communicating through a tunnel

Note VTP works between tunneled devices if Layer 2 protocol tunneling is enabled. See Chapter 27, "Configuring Layer 2 Protocol Tunneling," for configuration details.

- To configure an EtherChannel as an asymmetrical link, all ports in the EtherChannel must have the same tunneling configuration. Because the Layer 3 packet within the Layer 2 frame cannot be identified, you must configure the EtherChannel to use MAC-address-based frame distribution.

The following configuration guidelines are *required* for your Layer 2 protocol tunneling configuration:

- On all the service provider edge switches, PortFast BPDU filtering must be enabled on the 802.1Q tunnel ports as follows:

```
Router(config-if)# spanning-tree bpdufilter enable
Router(config-if)# spanning-tree portfast
```

Note Spanning-tree BPDU filtering is enabled automatically on tunnel ports.

- At least one VLAN must be available for native VLAN tagging (**vlan dot1q tag native** option). If you use all the available VLANs and then try to enable the **vlan dot1q tag native** option, the option will not be enabled.
- On all the service provider core switches, tag native VLAN egress traffic and drop untagged native VLAN ingress traffic by entering the following command:

```
Router(config)# vlan dot1q tag native
```

- On all the customer switches, *either* enable or disable the global **vlan dot1q tag native** option.

Note If this option is enabled on one switch and disabled on another switch, all traffic is dropped; all customer switches must have this option configured the same on each switch.

The following configuration guidelines are *optional* for your Layer 2 protocol tunneling configuration:

- Because all the BPDUs are being dropped, spanning tree PortFast can be enabled on Layer 2 protocol tunnel ports as follows:

```
Router(config-if)# spanning-tree portfast trunk
```

- If the service provider does not want the customer to see its switches, CDP should be disabled on the 802.1Q tunnel port as follows:

```
Router(config-if)# no cdp enable
```

Configuring 802.1Q Tunneling

These sections describe 802.1Q tunneling configuration:

- Configuring 802.1Q Tunnel Ports, page 26-6
- Configuring the Switch to Tag Native VLAN Traffic, page 26-7

Caution Ensure that only the appropriate tunnel ports are in any VLAN used for tunneling and that one VLAN is used for each tunnel. Incorrect assignment of tunnel ports to VLANs can forward traffic inappropriately.

Configuring 802.1Q Tunnel Ports

To configure 802.1Q tunneling on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`switchport`**	Configures the LAN port for Layer 2 switching. • You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional **switchport** commands with keywords. • Required only if you have not entered the **switchport** command already for the interface.
Step 3	`Router(config-if)#` **`switchport mode dot1q-tunnel`**	Configures the Layer 2 port as a tunnel port.
Step 4	`Router(config-if)#` **`no lldp transmit`**	(Required on PE ports) Disables LLDP. **Note** CDP is automatically disabled.
Step 5	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 6	`Router#` **`show dot1q-tunnel`** `[{`**`interface`** *`type`*[1] *`interface-number`*`}]`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure tunneling on port 4/1 and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 4/1
Router(config-if)# switchport mode dot1q-tunnel
Router(config-if)# no lldp transmit
Router(config-if)# end
Router# show dot1q-tunnel interface
```

Configuring the Switch to Tag Native VLAN Traffic

The **vlan dot1q tag native** command is a global command that configures the switch to tag native VLAN traffic, and admit only 802.1Q tagged frames on 802.1Q trunks, dropping any untagged traffic, including untagged traffic in the native VLAN.

On ports where you enter the **no switchport trunk native vlan tag** interface command, the function of the **vlan dot1q tag native** global command is disabled.

These sections describe how to configure the switch to tag native VLAN traffic:

- Configuring the Switch to Tag Native VLAN Traffic Globally, page 26-7
- Configuring Ports Not to Tag Native VLAN Traffic, page 26-7

Configuring the Switch to Tag Native VLAN Traffic Globally

With Release 12.2(33)SXH and later releases, to configure the switch to tag traffic in the native VLAN globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# vlan dot1q tag native`	Configures the switch to tag native VLAN traffic globally.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show vlan dot1q tag native | include globally`	Verifies the configuration.

This example shows how to configure the switch to tag native VLAN traffic and verify the configuration:

```
Router# configure terminal
Router(config)# vlan dot1q tag native
Router(config)# end
Router# show vlan dot1q tag native | include globally
dot1q native vlan tagging is enabled globally
Router(config)#
```

Configuring Ports Not to Tag Native VLAN Traffic

When the switch is configured to tag native VLAN traffic globally, you can disable native VLAN tagging on a per-port basis.

To configure a port not to tag traffic in the native VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Selects the LAN port to configure.
Step 2	`Router(config-if)# switchport`	Configures the LAN port for Layer 2 switching. • You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional **switchport** commands with keywords. • Required only if you have not entered the **switchport** command already for the port.

	Command	Purpose
Step 3	`Router(config-if)#` **`no switchport trunk native vlan tag`**	Configures the Layer 2 port not to tag native VLAN traffic.
Step 4	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 5	`Router#` **`show interface`** *`type`*[1] *`interface-number`* `|` **`include tagging`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note The **switchport trunk native vlan tag** interface command does not enable native VLAN tagging unless the switch is configured to tag native VLAN traffic globally.

This example shows how to configure Gigabit Ethernet port 1/4 to tag traffic in the native VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/4
Router(config-if)# switchport trunk native vlan tag
Router(config-if)# end
Router# show interface gigabitethernet 1/4 switchport | include tagging
Administrative Native VLAN tagging: enabled
Operational Native VLAN tagging: disabled
Router#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Layer 2 Protocol Tunneling

This chapter describes how to configure Layer 2 protocol tunneling in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html
- The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX switching modules do not support Layer 2 protocol tunneling.

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

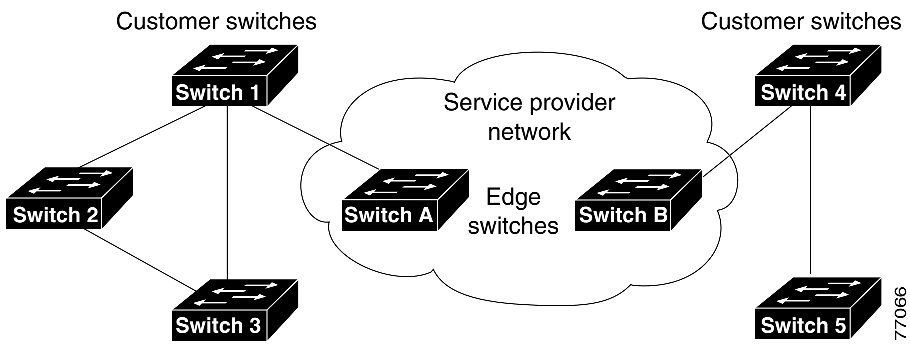
Understanding Layer 2 Protocol Tunneling

Layer 2 protocol tunneling allows Layer 2 protocol data units (PDUs) (CDP, STP, and VTP) to be tunneled through a network. This section uses the following terminology:

- Edge switch—The switch connected to the customer switch and placed on the boundary of the service provider network (see Figure 27-1).
- Layer 2 protocol tunnel port—A port on the edge switch on which a specific tunneled protocol can be encapsulated or deencapsulated. The Layer 2 protocol tunnel port is configured through CLI commands.
- Tunneled PDU—A CDP, STP, or VTP PDU.

Without Layer 2 protocol tunneling, tunnel ports drop STP and VTP packets and process CDP packets. This handling of the PDUs creates different spanning tree domains (different spanning tree roots) for the customer switches. For example, STP for a VLAN on switch 1 (see Figure 27-1) builds a spanning tree topology on switches 1, 2, and 3 without considering convergence parameters based on switches 4 and 5. To provide a single spanning tree domain for the customer, a generic scheme to tunnel BPDUs was created for control protocol PDUs (CDP, STP, and VTP). This process is referred to as Generic Bridge PDU Tunneling (GBPT).

Figure 27-1 Layer 2 Protocol Tunneling Network Configuration



GBPT provides a scalable approach to PDU tunneling by software encapsulating the PDUs in the ingress edge switches and then multicasting them in hardware. All switches inside the service provider network treat these encapsulated frames as data packets and forward them to the other end. The egress edge switch listens for these special encapsulated frames and deencapsulates them; they are then forwarded out of the tunnel.

The encapsulation involves rewriting the destination media access control (MAC) address in the PDU. An ingress edge switch rewrites the destination MAC address of the PDUs received on a Layer 2 tunnel port with the Cisco proprietary multicast address (01-00-0c-cd-cd-d0). The PDU is then flooded to the native VLAN of the Layer 2 tunnel port. If you enable Layer 2 protocol tunneling on a port, PDUs of an enabled protocol are not sent out. If you disable Layer 2 protocol tunneling on a port, the disabled protocols function the same way they were functioning before Layer 2 protocol tunneling was enabled on the port.

Configuring Support for Layer 2 Protocol Tunneling

Note

- Encapsulated PDUs received by an 802.1Q tunnel port are transmitted from other tunnel ports in the same VLAN on the switch.
- Configure jumbo frame support on Layer 2 protocol tunneling ports:
  - See the "Configuring Jumbo Frame Support" section on page 8-10.
  - Take note of the modules listed in the "Configuring Jumbo Frame Support" section that do not support jumbo frames.

To configure Layer 2 protocol tunneling on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`switchport`**	Configures the LAN port for Layer 2 switching. • You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional **switchport** commands with keywords. • Required only if you have not entered the **switchport** command already for the interface.
Step 3	`Router(config-if)#` **`l2protocol-tunnel`** `[`**`cdp`** `|` **`lldp`** `|` **`stp`** `|` **`vtp`**`]`	Configures the Layer 2 port as a Layer 2 protocol tunnel port for all protocols or only the specified protocol.
	`Router(config-if)#` **`l2protocol-tunnel drop-threshold`** `{[`**`cdp`** `|` **`lldp`** `|` **`stp`** `|` **`vtp`**`]` *`packets`*`}`	(Optional) Configures the port as a Layer 2 protocol tunnel port and sets a drop threshold for all protocols or only the specified protocol.
	`Router(config-if)#` **`l2protocol-tunnel shutdown-threshold`** `{[`**`cdp`** `|` **`lldp`** `|` **`stp`** `|` **`vtp`**`]` *`packets`*`}`	(Optional) Configures the port as a Layer 2 protocol tunnel port and sets a shutdown threshold for all protocols or only the specified protocol.
Step 4	`Router(config-if)#` **`no lldp transmit`**	(Required on PE ports) Disables LLDP. **Note** CDP is automatically disabled.
Step 5	`Router(config)#` **`end`**	Exits configuration mode.
Step 6	`Router#` **`show l2protocol-tunnel`** `[`**`interface`** *`type`*[1] *`slot/port`*`|`**`summary`**`]`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When you configure a Layer 2 port as a Layer 2 protocol tunnel port, note the following information:

- Optionally, you may specify a drop threshold for the port. The drop threshold value, from 1 to 4096, determines the number of packets to be processed for that protocol on that interface in one second. When the drop threshold is exceeded, PDUs for the specified protocol are dropped for the remainder of the one-second period. If a drop threshold is not specified, the value is 0 (drop threshold disabled).
- Optionally, you may specify a shutdown threshold for the port. The shutdown threshold value, from 1 to 4096, determines the number of packets to be processed for that protocol on that interface in one second. When the shutdown threshold is exceeded, the port is put in errdisable state. If a shutdown threshold is not specified, the value is 0 (shutdown threshold disabled).
- If you specify both a drop threshold and a shutdown threshold for the port, packets exceeding the drop threshold will not be forwarded but will be counted toward the shutdown threshold.

Note See the Cisco IOS Master Command List for more information about the **l2ptguard** keyword for the following commands:

- **errdisable detect cause**
- **errdisable recovery**

This example shows how to configure Layer 2 protocol tunneling and drop and shu tdown thresholds on port 5/1 for CDP, STP, and VTP, and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# switchport
Router(config-if)# l2protocol-tunnel shutdown-threshold cdp 400
Router(config-if)# l2protocol-tunnel shutdown-threshold stp 400
Router(config-if)# l2protocol-tunnel shutdown-threshold vtp 400
Router(config-if)# l2protocol-tunnel drop-threshold vtp 200
Router(config-if)# no lldp transmit
Router(config-if)# end
Router# show l2protocol-tunnel summary
COS for Encapsulated Packets: 5
Drop Threshold for Encapsulated Packets: 0

Port      Protocol    Shutdown            Drop                 Status
                      Threshold           Threshold
                      (cdp/lldp/stp/vtp)  (cdp/lldp/stp/vtp)
-------- ------------ ------------------- -------------------- ----------
Fa5/1    -- -- -- ---  400/----/ 400/ 400 ----/----/----/ 200   down(trunk)

Router#
```

This example shows how to display counter information for port 5/1:

```
Router# show l2protocol-tunnel interface fastethernet 5/1
COS for Encapsulated Packets: 5

Port        Protocol Thresholds          Counters
                     Shutdown  Drop      Encap     Decap     Drop
----------- -------- --------- --------- --------- --------- ---------
Router#
```

This example shows how to clear the Layer 2 protocol tunneling configuration from port 5/1:

```
Router(config-if)# no l2protocol-tunnel shutdown-threshold cdp 400
Router(config-if)# no l2protocol-tunnel shutdown-threshold stp 400
Router(config-if)# no l2protocol-tunnel shutdown-threshold vtp 400
Router(config-if)# no l2protocol-tunnel drop-threshold vtp 200
Router(config-if)# no l2protocol-tunnel cdp
Router(config-if)# no l2protocol-tunnel stp
Router(config-if)# no l2protocol-tunnel vtp
Router(config-if)# lldp transmit
Router(config-if)# end
Router# show l2protocol-tunnel summary
COS for Encapsulated Packets: 5
Drop Threshold for Encapsulated Packets: 0

Port      Protocol    Shutdown            Drop                 Status
                      Threshold           Threshold
                      (cdp/lldp/stp/vtp)  (cdp/lldp/stp/vtp)
-------- ------------ ------------------- -------------------- ----------

Router#
```

This example shows how to clear Layer 2 protocol tunneling port counters:

```
Router# clear l2protocol-tunnel counters
Router#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring STP and MST

This chapter describes how to configure the Spanning Tree Protocol (STP) and Multiple Spanning Tree (MST) protocol in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Note For information on configuring the PortFast, UplinkFast, and BackboneFast STP enhancements, see Chapter 29, "Configuring Optional STP Features."

Understanding STP

These sections describe how STP works:

STP Overview

STP, the IEEE 802.1D bridge protocol, is a Layer 2 link-management protocol that provides path redundancy while preventing undesirable loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. STP operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

In an extension known as per-VLAN spanning tree (PVST), Layer 2 Ethernet ports can use STP on all VLANs. By default, a single instance of STP runs on each configured VLAN (provided you do not manually disable STP). You can enable and disable STP on a per-VLAN basis.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The STP algorithm calculates the best loop-free path throughout a switched Layer 2 network. Layer 2 LAN ports send and receive STP frames at regular intervals. Network devices do not forward these frames, but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and network devices might learn end station MAC addresses on multiple Layer 2 LAN ports. These conditions result in an unstable network.

STP defines a tree with a root bridge and a loop-free path from the root to all network devices in the Layer 2 network. STP forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the STP algorithm recalculates the spanning tree topology and activates the standby path.

When two Layer 2 LAN ports on a network device are part of a loop, the STP port priority and port path cost setting determine which port is put in the forwarding state and which port is put in the blocking state. The STP port priority value represents the location of a port in the network topology and how efficiently that location allows the port to pass traffic. The STP port path cost value represents media speed.

Understanding the Bridge ID

Each VLAN on each network device has a unique 64-bit bridge ID consisting of a bridge priority value, an extended system ID, and an STP MAC address allocation.

This section contains these topics:

Bridge Priority Value

The bridge priority is a 4-bit value when the extended system ID is enabled (see Table 28-1 on page 28-3 and the "Configuring the Bridge Priority of a VLAN" section on page 28-35).

Extended System ID

A 12-bit extended system ID field is part of the bridge ID (see Table 28-1 on page 28-3). Chassis that support only 64 MAC addresses always use the 12-bit extended system ID. On chassis that support 1024 MAC addresses, you can enable use of the extended system ID. STP uses the VLAN ID as the extended system ID. See the "Enabling the Extended System ID" section on page 28-29.

Table 28-1 Bridge Priority Value and Extended System ID with the Extended System ID Enabled

Bridge Priority Value				Extended System ID (Set Equal to the VLAN ID)											
Bit 16	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

STP MAC Address Allocation

Catalyst 6500 series switch chassis have either 64 or 1024 MAC addresses available to support software features such as STP. To view the MAC address range on your chassis, enter the **show catalyst6000 chassis-mac-address** command.

For chassis with 64 MAC addresses, STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

When the extended system ID is not enabled, STP uses one MAC address per VLAN to make the bridge ID unique for each VLAN.

If you have a network device in your network with the extended system ID enabled, you should also enable the extended system ID on all other Layer 2 connected network devices to avoid undesirable root bridge election and spanning tree topology issues.

When the extended system ID is enabled, the root bridge priority becomes a multiple of 4096 plus the VLAN ID. With the extended system ID enabled, a switch bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge, the lowest being preferred) can only be specified as a multiple of 4096. Only the following values are possible: 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440.

If another bridge in the same spanning tree domain does not have the extended system ID enabled, it could win root bridge ownership because of the finer granularity in the selection of its bridge ID.

Understanding Bridge Protocol Data Units

Bridge protocol data units (BPDUs) are transmitted in one direction from the root bridge. Each network device sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the network device that the transmitting network device believes to be the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- Message age
- The identifier of the transmitting port
- Values for the hello, forward delay, and max-age protocol timers

When a network device transmits a BPDU frame, all network devices connected to the LAN on which the frame is transmitted receive the BPDU. When a network device receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU, and, if the topology changes, initiate a BPDU transmission.

A BPDU exchange results in the following:

- One network device is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each network device based on the path cost.
- A designated bridge for each LAN segment is selected. This is the network device closest to the root bridge through which frames are forwarded to the root.
- A root port is selected. This is the port providing the best path from the bridge to the root bridge.
- Ports included in the spanning tree are selected.

Election of the Root Bridge

For each VLAN, the network device with the highest-priority bridge ID (the lowest numerical ID value) is elected as the root bridge. If all network devices are configured with the default priority (32768), the network device with the lowest MAC address in the VLAN becomes the root bridge. The bridge priority value occupies the most significant bits of the bridge ID.

When you change the bridge priority value, you change the probability that the switch will be elected as the root bridge. Configuring a higher-priority value increases the probability; a lower-priority value decreases the probability.

The STP root bridge is the logical center of the spanning tree topology in a Layer 2 network. All paths that are not needed to reach the root bridge from anywhere in the Layer 2 network are placed in STP blocking mode.

BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. STP uses this information to elect the root bridge for the Layer 2 network, to elect the root port leading to the root bridge, and to determine the designated port for each Layer 2 segment.

STP Protocol Timers

Table 28-2 describes the STP protocol timers that affect STP performance.

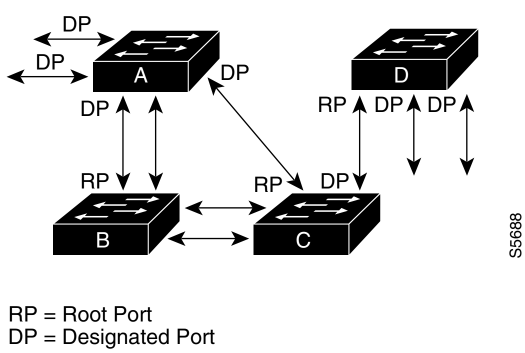
Table 28-2 STP Protocol Timers

Variable	Description
Hello timer	Determines how often the network device broadcasts hello messages to other network devices.
Forward delay timer	Determines how long each of the listening and learning states last before the port begins forwarding.
Maximum age timer	Determines the amount of time protocol information received on an port is stored by the network device.

Creating the Spanning Tree Topology

In Figure 28-1, Switch A is elected as the root bridge because the bridge priority of all the network devices is set to the default (32768) and Switch A has the lowest MAC address. However, due to traffic patterns, number of forwarding ports, or link types, Switch A might not be the ideal root bridge. By increasing the priority (lowering the numerical value) of the ideal network device so that it becomes the root bridge, you force an STP recalculation to form a new spanning tree topology with the ideal network device as the root.

Figure 28-1 Spanning Tree Topology



When the spanning tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change. The goal is to make the fastest link the root port.

For example, assume that one port on Switch B is a fiber-optic link, and another port on Switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the STP port priority on the fiber-optic port to a higher priority (lower numerical value) than the root port, the fiber-optic port becomes the new root port.

STP Port States

These sections describe the STP port states:

STP Port State Overview

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a Layer 2 LAN port transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames. They must allow the frame lifetime to expire for frames that have been forwarded using the old topology.

Each Layer 2 LAN port using STP exists in one of the following five states:

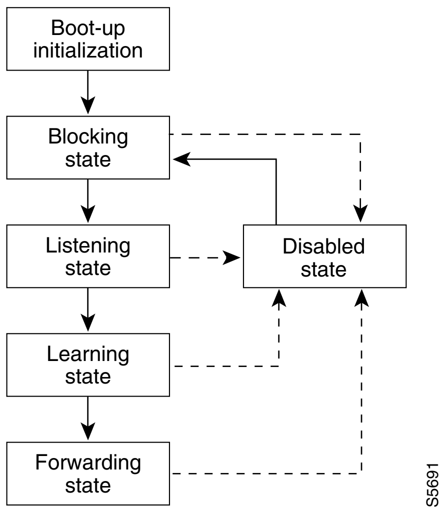
- Blocking—The Layer 2 LAN port does not participate in frame forwarding.
- Listening—First transitional state after the blocking state when STP determines that the Layer 2 LAN port should participate in frame forwarding.
- Learning—The Layer 2 LAN port prepares to participate in frame forwarding.
- Forwarding—The Layer 2 LAN port forwards frames.
- Disabled—The Layer 2 LAN port does not participate in STP and is not forwarding frames.

A Layer 2 LAN port moves through these five states as follows:

- From initialization to blocking
- From blocking to listening or to disabled
- From listening to learning or to disabled
- From learning to forwarding or to disabled
- From forwarding to disabled

Figure 28-2 illustrates how a Layer 2 LAN port moves through the five states.

Figure 28-2 STP Layer 2 LAN Interface States



When you enable STP, every port, VLAN, and network goes through the blocking state and the transitory states of listening and learning at power up. If properly configured, each Layer 2 LAN port stabilizes to the forwarding or blocking state.

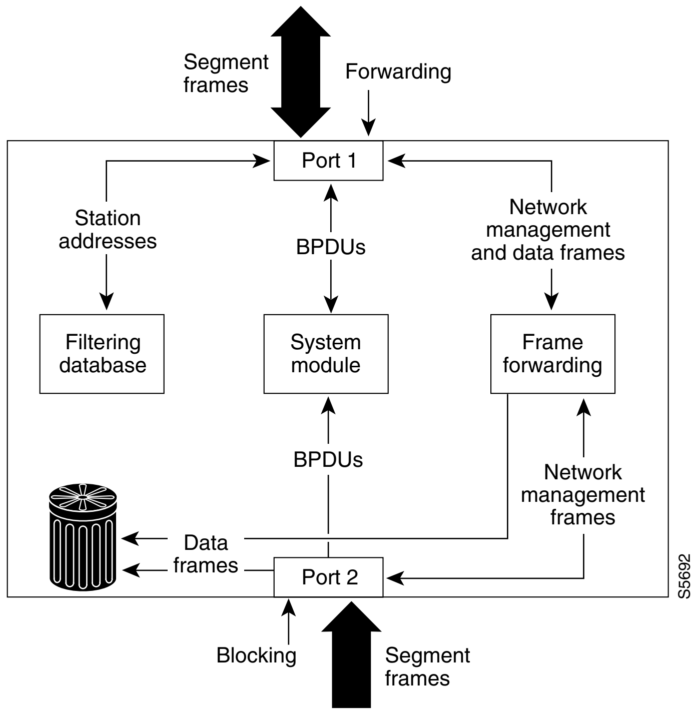
When the STP algorithm places a Layer 2 LAN port in the forwarding state, the following process occurs:

1. The Layer 2 LAN port is put into the listening state while it waits for protocol information that suggests it should go to the blocking state.
2. The Layer 2 LAN port waits for the forward delay timer to expire, moves the Layer 2 LAN port to the learning state, and resets the forward delay timer.
3. In the learning state, the Layer 2 LAN port continues to block frame forwarding as it learns end station location information for the forwarding database.
4. The Layer 2 LAN port waits for the forward delay timer to expire and then moves the Layer 2 LAN port to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A Layer 2 LAN port in the blocking state does not participate in frame forwarding, as shown in Figure 28-3. After initialization, a BPDU is sent out to each Layer 2 LAN port. A network device initially assumes it is the root until it exchanges BPDUs with other network devices. This exchange establishes which network device in the network is the root or root bridge. If only one network device is in the network, no exchange occurs, the forward delay timer expires, and the ports move to the listening state. A port always enters the blocking state following initialization.

Figure 28-3 Interface 2 in Blocking State



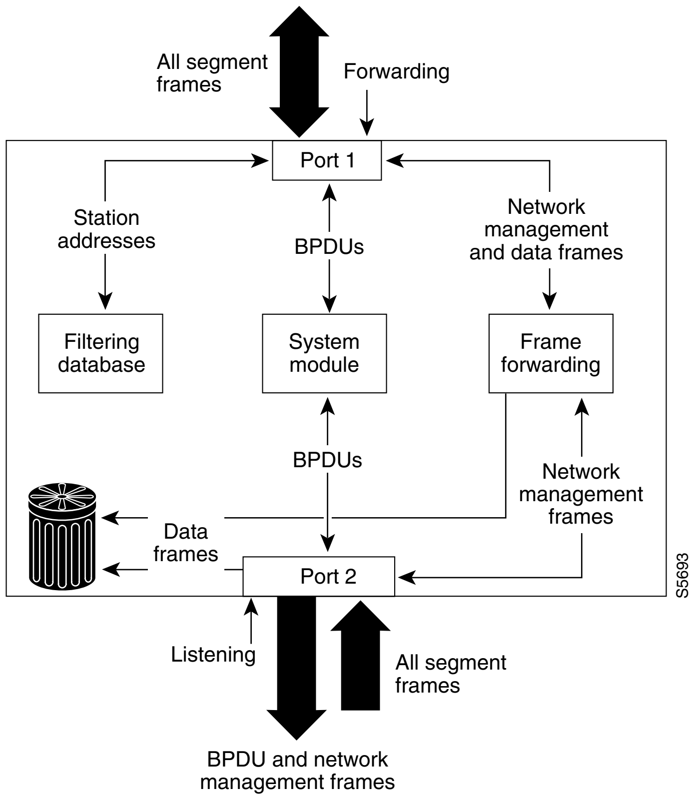
A Layer 2 LAN port in the blocking state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate end station location into its address database. (There is no learning on a blocking Layer 2 LAN port, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Does not transmit BPDUs received from the system module.
- Receives and responds to network management messages.

Listening State

The listening state is the first transitional state a Layer 2 LAN port enters after the blocking state. The Layer 2 LAN port enters this state when STP determines that the Layer 2 LAN port should participate in frame forwarding. Figure 28-4 shows a Layer 2 LAN port in the listening state.

Figure 28-4 Interface 2 in Listening State



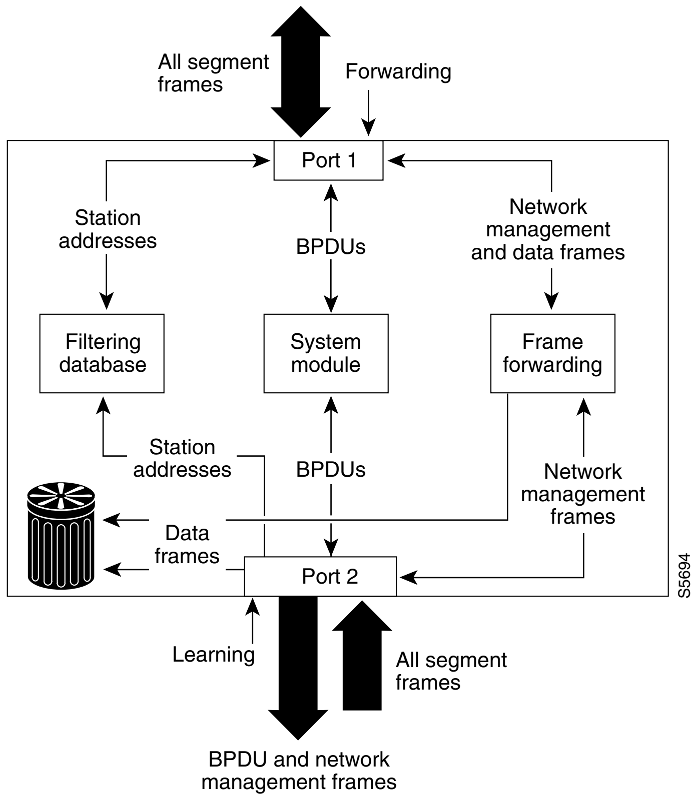
A Layer 2 LAN port in the listening state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another LAN port for forwarding.
- Does not incorporate end station location into its address database. (There is no learning at this point, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

Learning State

A Layer 2 LAN port in the learning state prepares to participate in frame forwarding. The Layer 2 LAN port enters the learning state from the listening state. Figure 28-5 shows a Layer 2 LAN port in the learning state.

Figure 28-5 Interface 2 in Learning State



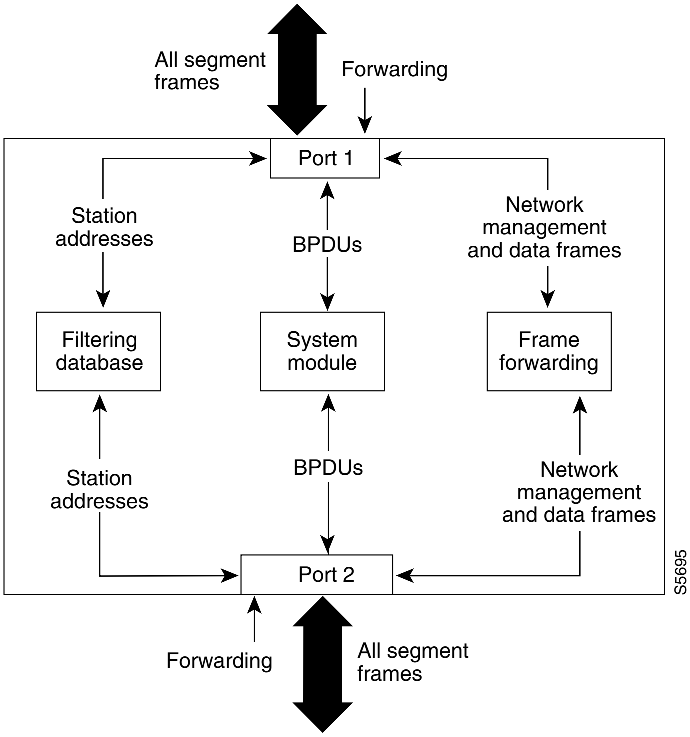
A Layer 2 LAN port in the learning state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Incorporates end station location into its address database.
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to network management messages.

Forwarding State

A Layer 2 LAN port in the forwarding state forwards frames, as shown in Figure 28-6. The Layer 2 LAN port enters the forwarding state from the learning state.

Figure 28-6 Interface 2 in Forwarding State



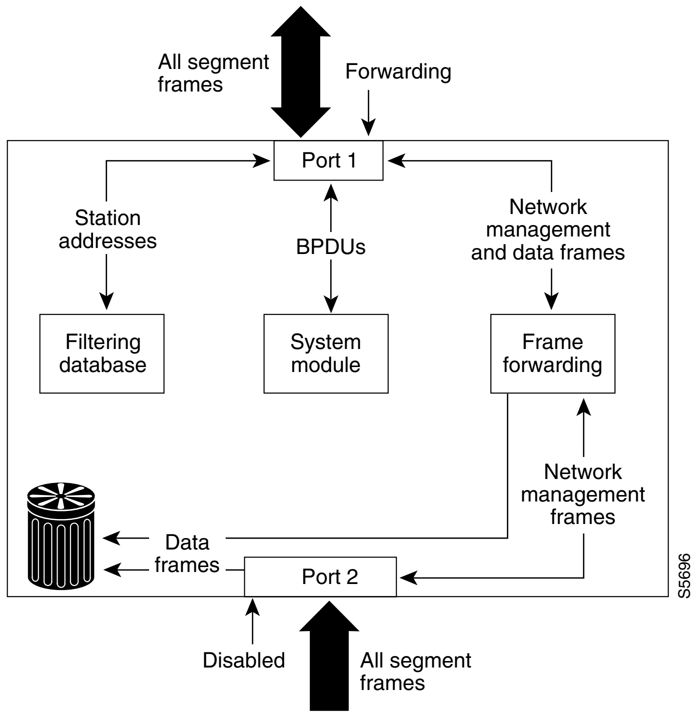
A Layer 2 LAN port in the forwarding state performs as follows:

- Forwards frames received from the attached segment.
- Forwards frames switched from another port for forwarding.
- Incorporates end station location information into its address database.
- Receives BPDUs and directs them to the system module.
- Processes BPDUs received from the system module.
- Receives and responds to network management messages.

Disabled State

A Layer 2 LAN port in the disabled state does not participate in frame forwarding or STP, as shown in Figure 28-7. A Layer 2 LAN port in the disabled state is virtually nonoperational.

Figure 28-7 Interface 2 in Disabled State



A disabled Layer 2 LAN port performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate end station location into its address database. (There is no learning, so there is no address database update.)
- Does not receive BPDUs.
- Does not receive BPDUs for transmission from the system module.

STP and IEEE 802.1Q Trunks

802.1Q trunks impose some limitations on the STP strategy for a network. In a network of Cisco network devices connected through 802.1Q trunks, the network devices maintain one instance of STP for each VLAN allowed on the trunks. However, non-Cisco 802.1Q network devices maintain only one instance of STP for all VLANs allowed on the trunks.

When you connect a Cisco network device to a non-Cisco device through an 802.1Q trunk, the Cisco network device combines the STP instance of the 802.1Q VLAN of the trunk with the STP instance of the non-Cisco 802.1Q network device. However, all per-VLAN STP information is maintained by Cisco network devices separated by a cloud of non-Cisco 802.1Q network devices. The non-Cisco 802.1Q cloud separating the Cisco network devices is treated as a single trunk link between the network devices. Interoperability with non-Cisco 802.1Q network devices is an extension of PVST known as PVST+.

For more information on 802.1Q trunks, see Chapter 17, "Configuring LAN Ports for Layer 2 Switching."

Understanding IEEE 802.1w RSTP

The RSTP takes advantage of point-to-point wiring and provides rapid convergence of the spanning tree. Reconfiguration of the spanning tree can occur in less than 1 second (in contrast to 50 seconds with the default settings in the 802.1D spanning tree).

These section describes how the RSTP works:

Port Roles and the Active Topology

The RSTP provides rapid convergence of the spanning tree by assigning port roles and by learning the active topology. The RSTP builds upon the 802.1D STP to select the switch with the highest switch priority (lowest numerical priority value) as the root bridge as described in the "Election of the Root Bridge" section on page 28-4. The RSTP then assigns one of these port roles to individual ports:

- Root port—Provides the best path (lowest cost) when the switch forwards packets to the root bridge.
- Designated port—Connects to the designated switch, which incurs the lowest path cost when forwarding packets from that LAN to the root bridge. The port through which the designated switch is attached to the LAN is called the designated port.
- Alternate port—Offers an alternate path toward the root bridge to that provided by the current root port.
- Backup port—Acts as a backup for the path provided by a designated port toward the leaves of the spanning tree. A backup port can exist only when two ports are connected in a loopback by a point-to-point link or when a switch has two or more connections to a shared LAN segment.
- Disabled port—Has no role within the operation of the spanning tree.

A port with the root or a designated port role is included in the active topology. A port with the alternate or backup port role is excluded from the active topology.

In a stable topology with consistent port roles throughout the network, the RSTP ensures that every root port and designated port immediately transition to the forwarding state while all alternate and backup ports are always in the discarding state (equivalent to blocking in 802.1D). The port state controls the operation of the forwarding and learning processes. Table 28-3 provides a comparison of 802.1D and RSTP port states.

Table 28-3 Port State Comparison

Operational Status	STP Port State (IEEE 802.1D)	RSTP Port State	Is Port Included in the Active Topology?
Enabled	Blocking	Discarding	No
Enabled	Listening	Discarding	No
Enabled	Learning	Learning	Yes

Table 28-3 Port State Comparison (continued)

Operational Status	STP Port State (IEEE 802.1D)	RSTP Port State	Is Port Included in the Active Topology?
Enabled	Forwarding	Forwarding	Yes
Disabled	Disabled	Discarding	No

To be consistent with Cisco STP implementations, this guide defines the port state as *blocking* instead of *discarding*. Designated ports start in the listening state.

Rapid Convergence

The RSTP provides for rapid recovery of connectivity following the failure of a switch, a switch port, or a LAN. It provides rapid convergence for edge ports, new root ports, and ports connected through point-to-point links as follows:

- Edge ports—If you configure a port as an edge port on an RSTP switch by using the **spanning-tree portfast** interface configuration command, the edge port immediately transitions to the forwarding state. An edge port is the same as a Port Fast-enabled port, and you should enable it only on ports that connect to a single end station.
- Root ports—If the RSTP selects a new root port, it blocks the old root port and immediately transitions the new root port to the forwarding state.
- Point-to-point links—If you connect a port to another port through a point-to-point link and the local port becomes a designated port, it negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology.

  As shown in Figure 28-8, switch A is connected to switch B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of switch A is a smaller numerical value than the priority of switch B. Switch A sends a proposal message (a configuration BPDU with the proposal flag set) to switch B, proposing itself as the designated switch.

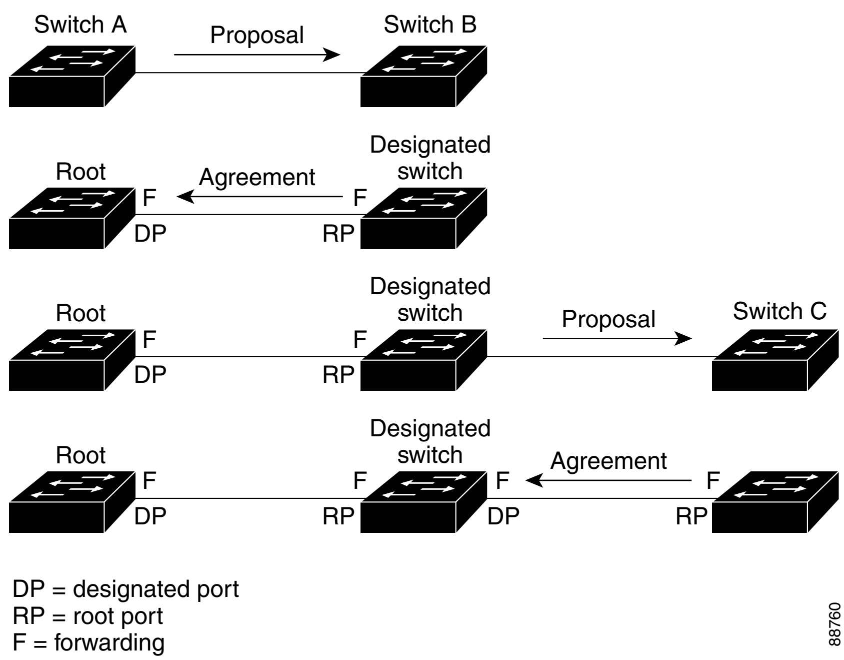
  After receiving the proposal message, switch B selects as its new root port the port from which the proposal message was received, forces all nonedge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

  After receiving switch B's agreement message, switch A also immediately transitions its designated port to the forwarding state. No loops in the network are formed because switch B blocked all of its nonedge ports and because there is a point-to-point link between switches A and B.

  When switch C is connected to switch B, a similar set of handshaking messages are exchanged. Switch C selects the port connected to switch B as its root port, and both ends immediately transition to the forwarding state. With each iteration of this handshaking process, one more switch joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree.

  The switch learns the link type from the port duplex mode: a full-duplex port is considered to have a point-to-point connection and a half-duplex port is considered to have a shared connection. You can override the default setting that is controlled by the duplex setting by using the **spanning-tree link-type** interface configuration command.

Figure 28-8 Proposal and Agreement Handshaking for Rapid Convergence



Synchronization of Port Roles

When the switch receives a proposal message on one of its ports and that port is selected as the new root port, the RSTP forces all other ports to synchronize with the new root information.

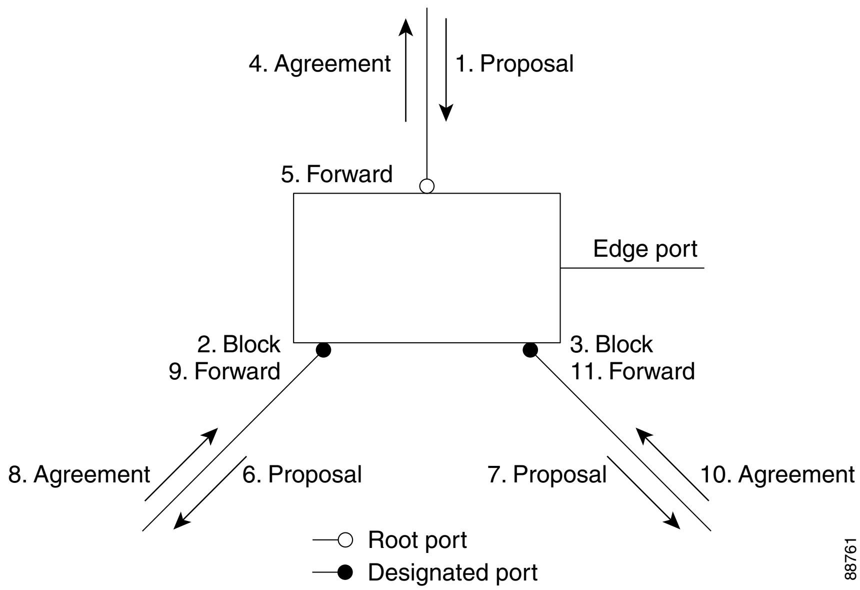
The switch is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the switch is synchronized if:

- That port is in the blocking state.
- It is an edge port (a port configured to be at the edge of the network).

If a designated port is in the forwarding state and is not configured as an edge port, it transitions to the blocking state when the RSTP forces it to synchronize with new root information. In general, when the RSTP forces a port to synchronize with root information and the port does not satisfy any of the above conditions, its port state is set to blocking.

After ensuring that all of the ports are synchronized, the switch sends an agreement message to the designated switch corresponding to its root port. When the switches connected by a point-to-point link are in agreement about their port roles, the RSTP immediately transitions the port states to forwarding. The sequence of events is shown in Figure 28-9.

Figure 28-9 Sequence of Events During Rapid Convergence



Bridge Protocol Data Unit Format and Processing

These sections describe bridge protocol data unit (BPDU) format and processing:

- BPDU Format and Processing Overview, page 28-16
- Processing Superior BPDU Information, page 28-17
- Processing Inferior BPDU Information, page 28-17

BPDU Format and Processing Overview

The RSTP BPDU format is the same as the 802.1D BPDU format except that the protocol version is set to 2. A new 1-byte Version 1 Length field is set to zero, which means that no Version 1 protocol information is present. Table 28-4 describes the RSTP flag fields.

Table 28-4 RSTP BPDU Flags

Bit	Function
0	Topology change (TC)
1	Proposal
2–3:	Port role:
00	Unknown
01	Alternate port or backup port
10	Root port
11	Designated port
4	Learning

Table 28-4 RSTP BPDU Flags (continued)

Bit	Function
5	Forwarding
6	Agreement
7	Topology change acknowledgement (TCA)

The sending switch sets the proposal flag in the RSTP BPDU to propose itself as the designated switch on that LAN. The port role in the proposal message is always set to the designated port.

The sending switch sets the agreement flag in the RSTP BPDU to accept the previous proposal. The port role in the agreement message is always set to the root port.

The RSTP does not have a separate TCN BPDU. It uses the topology change (TC) flag to show the topology changes. However, for interoperability with 802.1D switches, the RSTP switch processes and generates TCN BPDUs.

The learning and forwarding flags are set according to the state of the sending port.

Processing Superior BPDU Information

A superior BPDU is a BPDU with root information (such as lower switch ID or lower path cost) that is superior to what is currently stored for the port.

If a port receives a superior BPDU, the RSTP triggers a reconfiguration. If the port is proposed and is selected as the new root port, RSTP forces all the other ports to synchronize.

If the BPDU received is an RSTP BPDU with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D BPDU, the switch does not set the proposal flag and starts the forward-delay timer for the port. The new root port requires twice the forward-delay time to transition to the forwarding state.

If the superior information received on the port causes the port to become a backup port or an alternate port, RSTP sets the port to the blocking state and sends an agreement message. The designated port continues sending BPDUs with the proposal flag set until the forward-delay timer expires, at which time the port transitions to the forwarding state.

Processing Inferior BPDU Information

An inferior BPDU is a BPDU with root information (such as higher switch ID or higher path cost) that is inferior to what is currently stored for the port.

If a designated port receives an inferior BPDU, it immediately replies with its own information.

Topology Changes

These are the differences between the RSTP and the 802.1D in handling spanning tree topology changes:

- Detection—Unlike 802.1D in which *any* transition between the blocking and the forwarding state causes a topology change, *only* transitions from the blocking to the forwarding state cause a topology change with RSTP (only an increase in connectivity is considered a topology change).

State changes on an edge port do not cause a topology change. When an RSTP switch detects a topology change, it deletes the learned information on all of its nonedge ports except on those from which it received the TC notification.

- Notification—The RSTP does not use TCN BPDUs, unlike 802.1D. However, for 802.1D interoperability, an RSTP switch processes and generates TCN BPDUs.
- Acknowledgement—When an RSTP switch receives a TCN message on a designated port from an 802.1D switch, it replies with an 802.1D configuration BPDU with the TCA bit set. However, if the TC-while timer (the same as the TC timer in 802.1D) is active on a root port connected to an 802.1D switch and a configuration BPDU with the TCA set is received, the TC-while timer is reset.

  This method of operation is only required to support 802.1D switches. The RSTP BPDUs never have the TCA bit set.
- Propagation—When an RSTP switch receives a TC message from another switch through a designated or root port, it propagates the change to all of its nonedge, designated ports and to the root port (excluding the port on which it is received). The switch starts the TC-while timer for all such ports and flushes the information learned on them.
- Protocol migration—For backward compatibility with 802.1D switches, RSTP selectively sends 802.1D configuration BPDUs and TCN BPDUs on a per-port basis.

  When a port is initialized, the migrate-delay timer is started (specifies the minimum time during which RSTP BPDUs are sent), and RSTP BPDUs are sent. While this timer is active, the switch processes all BPDUs received on that port and ignores the protocol type.

  If the switch receives an 802.1D BPDU after the port migration-delay timer has expired, it assumes that it is connected to an 802.1D switch and starts using only 802.1D BPDUs. However, if the RSTP switch is using 802.1D BPDUs on a port and receives an RSTP BPDU after the timer has expired, it restarts the timer and starts using RSTP BPDUs on that port.

Rapid-PVST

Rapid-PVST uses the existing configuration for PVST+; however, Rapid-PVST uses RSTP to provide faster convergence. Independent VLANs run their own RSTP instance.

Dynamic entries are flushed immediately on a per-port basis upon receiving a topology change.

UplinkFast and BackboneFast configurations are ignored in Rapid-PVST mode; both features are included in RSTP.

In Cisco IOS Release 12.2(33)SXI and later releases, Rapid-PVST mode supports unidirectional link failure detection as described in the "Detecting Unidirectional Link Failure" section on page 28-25.

Understanding MST

These sections describe MST:

MST Overview

MST maps multiple VLANs into a spanning tree instance, with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of spanning tree instances required to support a large number of VLANs. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

The most common initial deployment of MST is in the backbone and distribution layers of a Layer 2 switched network. This deployment provides the kind of highly available network that is required in a service-provider environment.

MST provides rapid spanning tree convergence through explicit handshaking, which eliminates the 802.1D forwarding delay and quickly transitions root bridge ports and designated ports to the forwarding state.

MST improves spanning tree operation and maintains backward compatibility with these STP versions:

- Original 802.1D spanning tree
- Existing Cisco-proprietary Multiple Instance STP (MISTP)
- Existing Cisco per-VLAN spanning tree plus (PVST+)
- Rapid per-VLAN spanning tree plus (rapid PVST+)

For information about PVST+ and rapid PVST+, see the preceding sections in this chapter. For information about other spanning tree features such as Port Fast, UplinkFast, and root guard, see Chapter 29, "Configuring Optional STP Features."

Note

- IEEE 802.1w defined the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.
- IEEE 802.1s defined MST and was incorporated into IEEE 802.1Q.

MST Regions

For switches to participate in MST instances, you must consistently configure the switches with the same MST configuration information. A collection of interconnected switches that have the same MST configuration comprises an MST region as shown in Figure 28-10 on page 28-22.

The MST configuration controls to which MST region each switch belongs. The configuration includes the name of the region, the revision number, and the MST VLAN-to-instance assignment map.

A region can have one or multiple members with the same MST configuration; each member must be capable of processing RSTP bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network, but each region can support up to 65 spanning tree instances. Instances can be identified by any number in the range from 0 to 4094. You can assign a VLAN to only one spanning tree instance at a time.

IST, CIST, and CST

These sections describe internal spanning tree (IST), common and internal spanning tree (CIST), and common spanning tree (CST):

IST, CIST, and CST Overview

Unlike other spanning tree protocols, in which all the spanning tree instances are independent, MST establishes and maintains IST, CIST, and CST spanning trees:

- An IST is the spanning tree that runs in an MST region.

  Within each MST region, MST maintains multiple spanning tree instances. Instance 0 is a special instance for a region, known as the IST. All other MST instances are numbered from 1 to 4094.

  The IST is the only spanning tree instance that sends and receives BPDUs. All of the other spanning tree instance information is contained in MSTP records (M-records), which are encapsulated within MST BPDUs. Because the MST BPDU carries information for all instances, the number of BPDUs that need to be processed to support multiple spanning tree instances is significantly reduced.

  All MST instances within the same region share the same protocol timers, but each MST instance has its own topology parameters, such as root bridge ID, root path cost, and so forth. By default, all VLANs are assigned to the IST.

  An MST instance is local to the region; for example, MST instance 1 in region A is independent of MST instance 1 in region B, even if regions A and B are interconnected.

- A CIST is a collection of the ISTs in each MST region.
- The CST interconnects the MST regions and single spanning trees.

The spanning tree computed in a region appears as a subtree in the CST that encompasses the entire switched domain. The CIST is formed by the spanning tree algorithm running among switches that support the 802.1w, 802.1s, and 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

For more information, see the "Spanning Tree Operation Within an MST Region" section on page 28-20 and the "Spanning Tree Operations Between MST Regions" section on page 28-21.

Spanning Tree Operation Within an MST Region

The IST connects all the MST switches in a region. When the IST converges, the root of the IST becomes the CIST regional root (called the *IST master* before the implementation of the 802.1s standard) as shown in Figure 28-10 on page 28-22. The CIST regional root is also the CIST root if there is only one region in the network. If the CIST root is outside the region, one of the MST switches at the boundary of the region is selected as the CIST regional root.

When an MST switch initializes, it sends BPDUs that identify itself as the root of the CIST and the CIST regional root, with both of the path costs to the CIST root and to the CIST regional root set to zero. The switch also initializes all of its MST instances and claims to be the root for all of them. If the switch receives superior MST root information (lower switch ID, lower path cost, and so forth) than currently stored for the port, it relinquishes its claim as the CIST regional root.

During initialization, a region might have many subregions, each with its own CIST regional root. As switches receive superior IST information from a neighbor in the same region, they leave their old subregions and join the new subregion that contains the true CIST regional root, which causes all subregions to shrink except for the one that contains the true CIST regional root.

For correct operation, all switches in the MST region must agree on the same CIST regional root. Therefore, any two switches in the region only synchronize their port roles for an MST instance if they converge to a common CIST regional root.

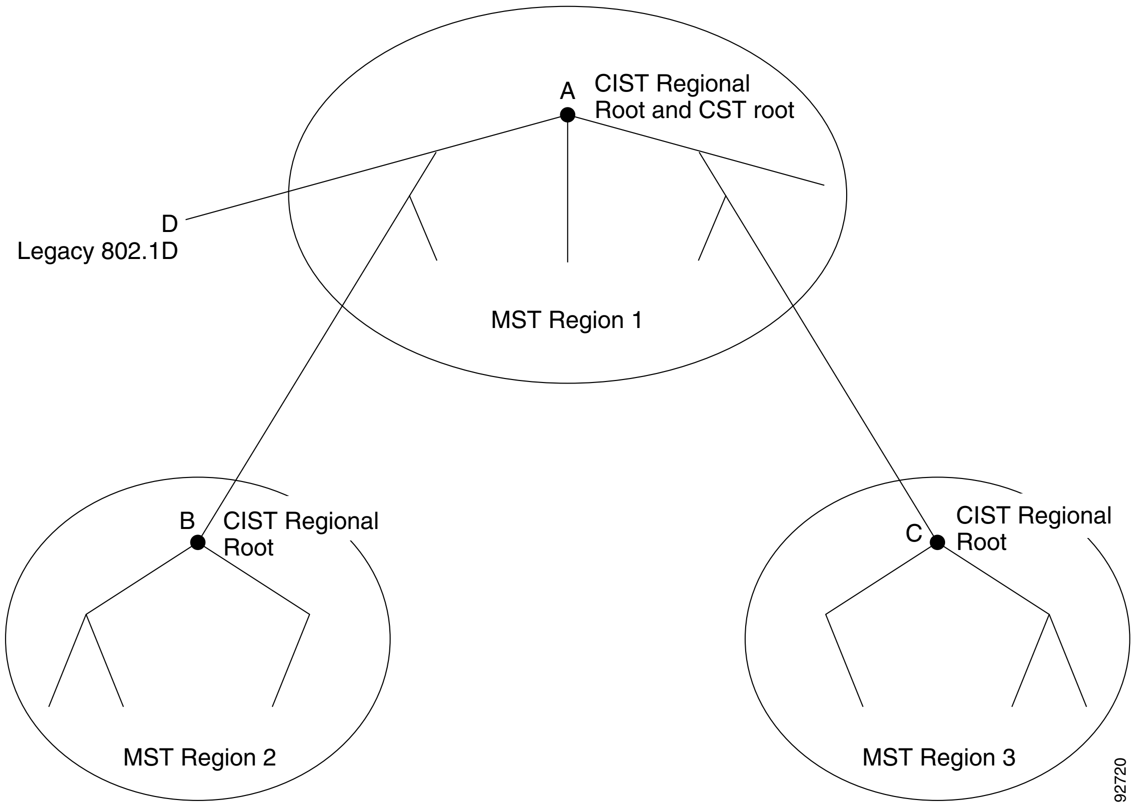
Spanning Tree Operations Between MST Regions

If there are multiple regions or 802.1D switches within the network, MST establishes and maintains the CST, which includes all MST regions and all 802.1D STP switches in the network. The MST instances combine with the IST at the boundary of the region to become the CST.

The IST connects all the MST switches in the region and appears as a subtree in the CIST that encompasses the entire switched domain. The root of the subtree is the CIST regional root. The MST region appears as a virtual switch to adjacent STP switches and MST regions.

Figure 28-10 shows a network with three MST regions and an 802.1D switch (D). The CIST regional root for region 1 (A) is also the CIST root. The CIST regional root for region 2 (B) and the CIST regional root for region 3 (C) are the roots for their respective subtrees within the CIST.

Figure 28-10 MST Regions, CIST Regional Roots, and CST Root



Only the CST instance sends and receives BPDUs, and MST instances add their spanning tree information into the BPDUs to interact with neighboring switches and compute the final spanning tree topology. Because of this, the spanning tree parameters related to BPDU transmission (for example, hello time, forward time, max-age, and max-hops) are configured only on the CST instance but affect all MST instances. Parameters related to the spanning tree topology (for example, switch priority, port VLAN cost, and port VLAN priority) can be configured on both the CST instance and the MST instance.

MST switches use Version 3 BPDUs or 802.1D STP BPDUs to communicate with 802.1D switches. MST switches use MST BPDUs to communicate with MST switches.

IEEE 802.1s Terminology

Some MST naming conventions used in the prestandard implementation have been changed to include identification of some *internal* and *regional* parameters. These parameters are used only within an MST region, compared to external parameters that are used throughout the whole network. Because the CIST is the only spanning tree instance that spans the whole network, only the CIST parameters require the external qualifiers and not the internal or regional qualifiers.

- The CIST root is the root bridge for the CIST, which is the unique instance that spans the whole network.
- The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. Remember that an MST region looks like a single switch to the CIST. The CIST external root path cost is the root path cost calculated between these virtual switches and switches that do not belong to any region.

- The CIST regional root was called the IST master in the prestandard implementation. If the CIST root is in the region, the CIST regional root is the CIST root. Otherwise, the CIST regional root is the closest switch to the CIST root in the region. The CIST regional root acts as a root bridge for the IST.
- The CIST internal root path cost is the cost to the CIST regional root in a region. This cost is only relevant to the IST, instance 0.

Table 28-5 compares the IEEE standard and the Cisco prestandard terminology.

Table 28-5 Prestandard and Standard Terminology

IEEE Standard Definition	Cisco Prestandard Implementation	Cisco Standard Implementation
CIST regional root	IST master	CIST regional root
CIST internal root path cost	IST master path cost	CIST internal path cost
CIST external root path cost	Root path cost	Root path cost
MSTI regional root	Instance root	Instance root
MSTI internal root path cost	Root path cost	Root path cost

Hop Count

MST does not use the message-age and maximum-age information in the configuration BPDU to compute the spanning tree topology. Instead, they use the path cost to the root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism.

By using the **spanning-tree mst max-hops** global configuration command, you can configure the maximum hops inside the region and apply it to the IST and all MST instances in that region. The hop count achieves the same result as the message-age information (triggers a reconfiguration). The root bridge of the instance always sends a BPDU (or M-record) with a cost of 0 and the hop count set to the maximum value. When a switch receives this BPDU, it decrements the received remaining hop count by one and propagates this value as the remaining hop count in the BPDUs it generates. When the count reaches zero, the switch discards the BPDU and ages the information held for the port.

The message-age and maximum-age information in the RSTP portion of the BPDU remain the same throughout the region, and the same values are propagated by the region-designated ports at the boundary.

Boundary Ports

In the Cisco prestandard implementation, a boundary port connects an MST region to one of these STP regions:

- A single spanning tree region running RSTP
- A single spanning tree region running PVST+ or rapid PVST+
- Another MST region with a different MST configuration

A boundary port also connects to a LAN, the designated switch of which is either a single spanning tree switch or a switch with a different MST configuration.

There is no definition of a boundary port in the 802.1s standard. The 802.1Q-2002 standard identifies two kinds of messages that a port can receive: internal (coming from the same region) and external. When a message is external, it is received only by the CIST. If the CIST role is root or alternate, or if the external BPDU is a topology change, it could have an impact on the MST instances. When a message is internal, the CIST part is received by the CIST, and each MST instance receives its respective M-record. The Cisco prestandard implementation treats a port that receives an external message as a boundary port, which means a port cannot receive a mix of internal and external messages.

An MST region includes both switches and LANs. A segment belongs to the region of its designated port. Therefore, a port in a different region from the designated port for a segment is a boundary port. This definition allows two ports internal to a region to share a segment with a port belonging to a different region, creating the possibility of receiving both internal and external messages on a port.

The primary change from the Cisco prestandard implementation is that a designated port is not defined as boundary unless it is running in an STP-compatible mode.

Note If there is an 802.1D STP switch on the segment, messages are always considered external.

The other change from the prestandard implementation is that the CIST regional root bridge ID field is now inserted where an RSTP or legacy 802.1s switch has the sender switch ID. The whole region performs like a single virtual switch by sending a consistent sender switch ID to neighboring switches. In this example, switch C would receive a BPDU with the same consistent sender switch ID of root, whether or not A or B is designated for the segment.

Standard-Compliant MST Implementation

The standard-compliant MST implementation includes features required to meet the standard, as well as some of the desirable prestandard functionality that is not yet incorporated into the published standard. These sections describe the standard-compliant MST implementation:

- Changes in Port-Role Naming, page 28-24
- Spanning Tree Interoperation Between Legacy and Standard-Compliant Switches, page 28-25
- Detecting Unidirectional Link Failure, page 28-25

Changes in Port-Role Naming

The boundary role was deleted from the final MST standard, but this boundary concept is maintained in the standard-compliant implementation. However, an MST instance (MSTI) port at a boundary of the region might not follow the state of the corresponding CIST port. The following two situations currently exist:

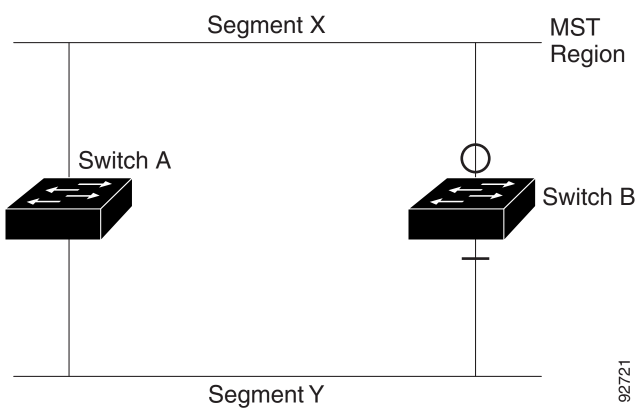
- The boundary port is the root port of the CIST regional root—When the CIST instance port is proposed and is synchronized, it can send back an agreement and move to the forwarding state only after all the corresponding MSTI ports are synchronized (and thus forwarding). The MSTI ports now have a special *master* role.
- The boundary port is not the root port of the CIST regional root—The MSTI ports follow the state and role of the CIST port. The standard provides less information, and it might be difficult to understand why an MSTI port can be alternately blocking when it receives no BPDUs (M-records). In this situation, although the boundary role no longer exists, when you enter the **show** commands, they identify a port as boundary in the *type* column of the output.

Spanning Tree Interoperation Between Legacy and Standard-Compliant Switches

Because automatic detection of prestandard switches can fail, you can use an interface configuration command to identify prestandard ports. A region cannot be formed between a standard and a prestandard switch, but they can interoperate before using the CIST. Only the capability of load balancing over different instances is lost in this specific situation. The CLI displays different flags depending on the port configuration when the port receives prestandard BPDUs. A syslog message also appears the first time a switch receives a prestandard BPDU on a port that has not been configured for prestandard BPDU transmission.

Figure 28-11 illustrates a standard-compliant switch connected to a prestandard switch. Assume that A is the standard-compliant switch and B is a prestandard switch, both configured to be in the same region. A is the root bridge for the CIST, and so B has a root port (BX) on segment X and an alternate port (BY) on segment Y. If segment Y flaps, and the port on BY becomes the alternate before sending out a single prestandard BPDU, AY cannot detect that a prestandard switch is connected to Y and continues to send standard BPDUs. The port BY is fixed in a boundary, and no load balancing is possible between A and B. The same problem exists on segment X, but B might transmit topology changes.

Figure 28-11 Standard-Compliant and Prestandard Switch Interoperation



Note We recommend that you minimize the interaction between standard and prestandard MST implementations.

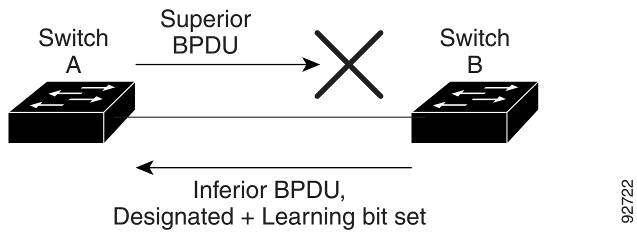
Detecting Unidirectional Link Failure

Using the dispute mechanism included in the IEEE 802.1D-2004 RSTP standard, the switch checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops.

When a designated port detects a conflict, it keeps its role, but reverts to a discarding (blocking) state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

Figure 28-12 illustrates a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to switch B. RSTP and MST BPDUs include the role and state of the sending port. With this information, switch A can detect that switch B does not react to the superior BPDUs it sends and that switch B is the designated, not root bridge. As a result, switch A blocks (or keeps blocking) its port, thus preventing the bridging loop.

Figure 28-12 Detecting Unidirectional Link Failure



Interoperability with IEEE 802.1D-1998 STP

A switch running MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D switches. If this switch receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MST switch also can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an RSTP BPDU (Version 2).

However, the switch does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D switch has been removed from the link unless the 802.1D switch is the designated switch. A switch might also continue to assign a boundary role to a port when the switch to which this switch is connected has joined the region. To restart the protocol migration process (force the renegotiation with neighboring switches), use the **clear spanning-tree detected-protocols** privileged EXEC command.

If all the 802.1D switches on the link are RSTP switches, they can process MST BPDUs as if they are RSTP BPDUs. Therefore, MST switches send either a Version 0 configuration and topology change notification (TCN) BPDUs or Version 3 MST BPDUs on a boundary port. A boundary port connects to a LAN, the designated switch of which is either a single spanning tree switch or a switch with a different MST configuration.

Configuring STP

These sections describe how to configure STP on VLANs:

- Enabling Rapid-PVST, page 28-37

Note The STP commands described in this chapter can be configured on any LAN port, but they are in effect only on LAN ports configured with the **switchport** keyword.

Caution We do not recommend disabling spanning tree, even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN.

Default STP Configuration

Table 28-6 shows the default STP configuration.

Table 28-6 STP Default Configuration

Feature	Default Value	
Enable state	STP enabled for all VLANs	
Mode	PVST	
Bridge priority	32768	
STP port priority (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports)	128	
STP port cost (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports)	10-Gigabit Ethernet:	2
	Gigabit Ethernet:	4
	Fast Ethernet:	19
	Ethernet:	100
STP VLAN port priority (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	128	
STP VLAN port cost (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	10-Gigabit Ethernet:	2
	Gigabit Ethernet:	4
	Fast Ethernet:	19
	Ethernet:	100
Hello time	2 seconds	
Forward delay time	15 seconds	
Maximum aging time	20 seconds	

Enabling STP

Note STP is enabled by default on VLAN 1 and on all newly created VLANs.

You can enable STP on a per-VLAN basis. The switch maintains a separate instance of STP for each VLAN (except on VLANs on which you disable STP).

To enable STP on a per-VLAN basis, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree vlan vlan_ID`	Enables STP on a per-VLAN basis. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 28-6 on page 28-27).
	`Router(config)# default spanning-tree vlan vlan_ID`	Reverts all STP parameters to default values for the specified VLAN.
	`Router(config)# no spanning-tree vlan vlan_ID`	Disables STP on the specified VLAN; see the following Cautions for information regarding this command.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show spanning-tree vlan vlan_ID`	Verifies that STP is enabled.

Caution Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some switches and bridges in a VLAN and leave it enabled on other switches and bridges in the VLAN. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network.

Caution We do not recommend disabling spanning tree, even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN.

This example shows how to enable STP on VLAN 200:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200
Router(config)# end
Router#
```

Note Because STP is enabled by default, entering a **show running** command to view the resulting configuration does not display the command you entered to enable STP.

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200

VLAN0200
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
```

```
                Address     00d0.00b8.14c8
                This bridge is the root
                Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768
                Address     00d0.00b8.14c8
                Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
                Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
Gi1/4            Desg FWD 200000    128.196  P2p
Gi1/5            Back BLK 200000    128.197  P2p

Router#
```

Note You must have at least one interface that is active in VLAN 200 to create a VLAN 200 spanning tree. In this example, two interfaces are active in VLAN 200.

Enabling the Extended System ID

Note
- The extended system ID is enabled permanently on chassis that support 64 MAC addresses.
- In Release 12.2(33)SXJ1 and later releases, the extended system ID is always enabled in VSS mode.

You can enable the extended system ID on chassis that support 1024 MAC addresses (see the "Understanding the Bridge ID" section on page 28-2).

To enable the extended system ID, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree extend system-id`	Enables the extended system ID. **Note** You cannot disable the extended system ID on chassis that support 64 MAC addresses or when you have configured extended range VLANs (see "Table 28-6STP Default Configuration" section on page 28-27).
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show spanning-tree vlan vlan_ID`	Verifies the configuration.

Note When you enable or disable the extended system ID, the bridge IDs of all active STP instances are updated, which might change the spanning tree topology.

This example shows how to enable the extended system ID:

```
Router# configure terminal
Router(config)# spanning-tree extend system-id
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary | include Extended
Extended system ID is enabled.
```

Configuring the Root Bridge

The switches supported by Cisco IOS Release 12.2SX maintain a separate instance of STP for each active VLAN. A bridge ID, consisting of the bridge priority and the bridge MAC address, is associated with each instance. For each VLAN, the network device with the lowest bridge ID becomes the root bridge for that VLAN.

To configure a VLAN instance to become the root bridge, enter the **spanning-tree vlan** *vlan_ID* **root** command to modify the bridge priority from the default value (32768) to a significantly lower value.

When you enter the **spanning-tree vlan** *vlan_ID* **root** command, the switch checks the bridge priority of the current root bridges for each VLAN. With the extended system ID enabled, the switch sets the bridge priority for the specified VLANs to 24576 if this value will cause the switch to become the root for the specified VLANs.

With the extended system ID enabled, if any root bridge for the specified VLANs has a bridge priority lower than 24576, the switch sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority. (4096 is the value of the least significant bit of a 4-bit bridge priority value; see Table 28-1 on page 28-3.)

Note The **spanning-tree vlan** *vlan_ID* **root** command fails if the value required to be the root bridge is less than 1.

With the extended system ID enabled, if all network devices in, for example, VLAN 20 have the default priority of 32768, entering the **spanning-tree vlan 20 root primary** command on the switch sets the bridge priority to 24576, which causes the switch to become the root bridge for VLAN 20.

Caution The root bridge for each instance of STP should be a backbone or distribution switch. Do not configure an access switch as the STP primary root.

Use the **diameter** keyword to specify the Layer 2 network diameter (that is, the maximum number of bridge hops between any two end stations in the Layer 2 network). When you specify the network diameter, the switch automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can use the **hello** keyword to override the automatically calculated hello time.

Note To preserve a stable STP topology, we recommend that you avoid configuring the hello time, forward delay time, and maximum age time manually after configuring the switch as the root bridge.

To configure the switch as the root bridge, perform this task:

	Command	Purpose
Step 1	Router(config)# **spanning-tree vlan** *vlan_ID* **root primary** [**diameter** *hops* [**hello-time** *seconds*]]	Configures the switch as the root bridge. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 28-6 on page 28-27).
	Router(config)# **no spanning-tree vlan** *vlan_ID* **root**	Clears the root bridge configuration.
Step 2	Router(config)# **end**	Exits configuration mode.

This example shows how to configure the switch as the root bridge for VLAN 10, with a network diameter of 4:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# end
Router#
```

Configuring a Secondary Root Bridge

When you configure a switch as the secondary root, the STP bridge priority is modified from the default value (32768) so that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other network devices in the network use the default bridge priority of 32768).

With the extended system ID is enabled, STP sets the bridge priority to 28672.

You can run this command on more than one switch to configure multiple backup root bridges. Use the same network diameter and hello time values as you used when configuring the primary root bridge.

To configure the switch as the secondary root bridge, perform this task:

	Command	Purpose
Step 1	Router(config)# [**no**] **spanning-tree vlan** *vlan_ID* **root secondary** [**diameter** *hops* [**hello-time** *seconds*]]	Configures the switch as the secondary root bridge. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2).
	Router(config)# **no spanning-tree vlan** *vlan_ID* **root**	Clears the root bridge configuring.
Step 2	Router(config)# **end**	Exits configuration mode.

This example shows how to configure the switch as the secondary root bridge for VLAN 10, with a network diameter of 4:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root secondary diameter 4
Router(config)# end
Router#
```

Configuring STP Port Priority

If a loop occurs, STP considers port priority when selecting a LAN port to put into the forwarding state. You can assign higher priority values to LAN ports that you want STP to select first and lower priority values to LAN ports that you want STP to select last. If all LAN ports have the same priority value, STP puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports. The possible priority range is 0 through 240 (default 128), configurable in increments of 16.

Cisco IOS uses the port priority value when the LAN port is configured as an access port and uses VLAN port priority values when the LAN port is configured as a trunk port.

To configure the STP port priority of a Layer 2 LAN interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`gigabitethernet 1`**`/`*`port`*`}	{`**`port-channel`** *`port_channel_number`*`}}`	Selects an interface to configure.
Step 2	`Router(config-if)#` **`spanning-tree port-priority`** *`port_priority`*	Configures the port priority for the LAN interface. The *port_priority* value can be from 1 to 252 in increments of 4.
	`Router(config-if)#` **`no spanning-tree port-priority`**	Reverts to the default port priority value.
Step 3	`Router(config-if)#` **`spanning-tree vlan`** *`vlan_ID`* **`port-priority`** *`port_priority`*	Configures the VLAN port priority for the LAN interface. The *port_priority* value can be from 1 to 252 in increments of 4. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2).
	`Router(config-if)# [`**`no`**`]` **`spanning-tree vlan`** *`vlan_ID`* **`port-priority`**	Reverts to the default VLAN port priority value.
Step 4	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 5	`Router#` **`show spanning-tree interface`** `{`**`gigabitethernet 1`**`/`*`port`*`}	{`**`port-channel`** *`port_channel_number`*`}` `Router#` **`show spanning-tree vlan`** *`vlan_ID`*	Verifies the configuration.

This example shows how to configure the STP port priority of Gigabit Ethernet port 1/4:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/4
Router(config-if)# spanning-tree port-priority 160
Router(config-if)# end
Router#
```

This example shows how to verify the configuration of Gigabit Ethernet port 1/4:

```
Router# show spanning-tree interface gigabitethernet 1/4
Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0001         Back BLK 200000    160.196  P2p
VLAN0006         Back BLK 200000    160.196  P2p
...
VLAN0198         Back BLK 200000    160.196  P2p
VLAN0199         Back BLK 200000    160.196  P2p
VLAN0200         Back BLK 200000    160.196  P2p
Router#
```

Gigabit Ethernet port 1/4 is a trunk. Several VLANs are configured and active as shown in the example. The port priority configuration applies to all VLANs on this interface.

Note The **show spanning-tree interface** command only displays information if the port is connected and operating. If this condition is not met, enter a **show running-config interface** command to verify the configuration.

This example shows how to configure the VLAN port priority of Gigabit Ethernet port 1/4:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/4
Router(config-if)# spanning-tree vlan 200 port-priority 64
Router(config-if)# end
Router#
```

The configuration entered in the example only applies to VLAN 200. All VLANs other than 200 still have a port priority of 160.

This example shows how to verify the configuration:

```
Router# show spanning-tree interface gigabitethernet 1/4
Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0001         Back BLK 200000    160.196  P2p
VLAN0006         Back BLK 200000    160.196  P2p
...
VLAN0199         Back BLK 200000    160.196  P2p
VLAN0200         Desg FWD 200000     64.196  P2p

Router#
```

You also can display spanning tree information for VLAN 200 using the following command:

```
Router# show spanning-tree vlan 200 interface gigabitethernet 1/4
Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
Gi1/4            Desg LRN 200000     64.196  P2p
```

Configuring STP Port Cost

The STP port path cost default value is determined from the media speed of a LAN interface. If a loop occurs, STP considers port cost when selecting a LAN interface to put into the forwarding state. You can assign lower cost values to LAN interfaces that you want STP to select first and higher cost values to LAN interfaces that you want STP to select last. If all LAN interfaces have the same cost value, STP puts the LAN interface with the lowest LAN interface number in the forwarding state and blocks other LAN interfaces. The possible cost range is 0 through 200000000 (the default is media specific).

STP uses the port cost value when the LAN interface is configured as an access port and uses VLAN port cost values when the LAN interface is configured as a trunk port.

To configure the STP port cost of a Layer 2 LAN interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`gigabitethernet 1`**`/`*`port`*`}	{`**`port-channel`** *`port_channel_number`*`}}`	Selects an interface to configure.
Step 2	`Router(config-if)#` **`spanning-tree cost`** *`port_cost`*	Configures the port cost for the LAN interface. The *port_cost* value can be from 1 to 200000000.
	`Router(config-if)#` **`no spanning-tree cost`**	Reverts to the default port cost.
Step 3	`Router(config-if)#` **`spanning-tree vlan`** *`vlan_ID`* **`cost`** *`port_cost`*	Configures the VLAN port cost for the LAN interface. The *port_cost* value can be from 1 to 200000000. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2).
	`Router(config-if)#` **`no spanning-tree vlan`** *`vlan_ID`* **`cost`**	Reverts to the default VLAN port cost.
Step 4	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 5	`Router#` **`show spanning-tree interface`** `{`**`gigabitethernet 1`**`/`*`port`*`}	{`**`port-channel`** *`port_channel_number`*`}` **`show spanning-tree vlan`** *`vlan_ID`*	Verifies the configuration.

This example shows how to change the STP port cost of Gigabit Ethernet port 1/4:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/4
Router(config-if)# spanning-tree cost 1000
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree interface gigabitethernet 1/4
Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0001         Back BLK 1000      160.196   P2p
VLAN0006         Back BLK 1000      160.196   P2p
VLAN0007         Back BLK 1000      160.196   P2p
VLAN0008         Back BLK 1000      160.196   P2p
VLAN0009         Back BLK 1000      160.196   P2p
VLAN0010         Back BLK 1000      160.196   P2p
Router#
```

This example shows how to configure the port priority at an individual port VLAN cost for VLAN 200:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/4
Router(config-if)# spanning-tree vlan 200 cost 2000
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 interface gigabitethernet 1/4
Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
Gi1/4            Desg FWD 2000       64.196   P2p
```

Note In the following output other VLANs (VLAN 1 for example) have not been affected by this configuration.

```
Router# show spanning-tree vlan 1 interface gigabitethernet 1/4
Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
Gi1/4            Back BLK 1000      160.196  P2p
Router#
```

Note The **show spanning-tree** command only displays information for ports that are in link-up operative state and are appropriately configured for DTP. If these conditions are not met, you can enter a **show running-config** command to confirm the configuration.

Configuring the Bridge Priority of a VLAN

Note Be careful when using this command. For most situations, we recommend that you enter the **spanning-tree vlan** *vlan_ID* **root primary** and the **spanning-tree vlan** *vlan_ID* **root secondary** commands to modify the bridge priority.

To configure the STP bridge priority of a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# **spanning-tree vlan** *vlan_ID* **priority** {**0** | **4096** | **8192** | **12288** | **16384** | **20480** | **24576** | **28672** | **32768** | **36864** | **40960** | **45056** | **49152** | **53248** | **57344** | **61440**}	Configures the bridge priority of a VLAN when the extended system ID is enabled. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2).
	Router(config)# **no spanning-tree vlan** *vlan_ID* **priority**	Reverts to the default bridge priority value.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show spanning-tree vlan** *vlan_ID* **bridge** [**detail**]	Verifies the configuration.

This example shows how to configure the bridge priority of VLAN 200 to 33792 when the extended system ID is disabled:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 priority 32768
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge
                                      Hello Max  Fwd
Vlan                      Bridge ID   Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          32768 0050.3e8d.64c8    2   20    15  ieee
Router#
```

Configuring the Hello Time

> **Note** Be careful when using this command. For most situations, we recommend that you use the **spanning-tree vlan** *vlan_ID* **root primary** and **spanning-tree vlan** *vlan_ID* **root secondary** commands to modify the hello time.

To configure the STP hello time of a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree vlan vlan_ID hello-time hello_time`	Configures the hello time of a VLAN. The *hello_time* value can be from 1 to 10 seconds. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2).
	`Router(config)# no spanning-tree vlan vlan_ID hello-time`	Reverts to the default hello time.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show spanning-tree vlan vlan_ID bridge [detail]`	Verifies the configuration.

This example shows how to configure the hello time for VLAN 200 to 7 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 hello-time 7
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge
                                      Hello  Max  Fwd
Vlan                       Bridge ID      Time   Age Delay   Protocol
---------------- -------------------- ---- ---- -----   --------
VLAN200          49152 0050.3e8d.64c8    7   20    15   ieee
Router#
```

Configuring the Forward-Delay Time for a VLAN

To configure the STP forward delay time for a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree vlan vlan_ID forward-time forward_time`	Configures the forward time of a VLAN. The *forward_time* value can be from 4 to 30 seconds. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2).
	`Router(config)# no spanning-tree vlan vlan_ID forward-time`	Reverts to the default forward time.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show spanning-tree vlan vlan_ID bridge [detail]`	Verifies the configuration.

This example shows how to configure the forward delay time for VLAN 200 to 21 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 forward-time 21
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge
                                      Hello Max  Fwd
Vlan                    Bridge ID      Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          49152 0050.3e8d.64c8    2   20    21  ieee
Router#
```

Configuring the Maximum Aging Time for a VLAN

To configure the STP maximum aging time for a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# **spanning-tree vlan** *vlan_ID* **max-age** *max_age*	Configures the maximum aging time of a VLAN. The *max_age* value can be from 6 to 40 seconds. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 23-1 on page 23-2).
	Router(config)# **no spanning-tree vlan** *vlan_ID* **max-age**	Reverts to the default maximum aging time.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show spanning-tree vlan** *vlan_ID* **bridge** [**detail**]	Verifies the configuration.

This example shows how to configure the maximum aging time for VLAN 200 to 36 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 max-age 36
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge
                                      Hello Max  Fwd
Vlan                    Bridge ID      Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          49152 0050.3e8d.64c8    2   36    15  ieee
Router#
```

Enabling Rapid-PVST

Rapid-PVST uses the existing PVST+ framework for configuration and interaction with other features. It also supports some of the PVST+ extensions.

To enable Rapid-PVST mode on the switch, enter the **spanning-tree mode rapid-pvst** command in privileged mode. To configure the switch in Rapid-PVST mode, see the "Configuring STP" section on page 28-26.

Specifying the Link Type

Rapid connectivity is established only on point-to-point links. Spanning tree views a point-to-point link as a segment connecting only two switches running the spanning tree algorithm. Because the switch assumes that all full-duplex links are point-to-point links and that half-duplex links are shared links, you can avoid explicitly configuring the link type. To configure a specific link type, enter the **spanning-tree linktype** command.

Restarting Protocol Migration

A switch running both MSTP and RSTP supports a built-in protocol migration process that enables the switch to interoperate with legacy 802.1D switches. If this switch receives a legacy 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MSTP switch can also detect that a port is at the boundary of a region when it receives a legacy BPDU, or an MST BPDU (version 3) associated with a different region, or an RST BPDU (version 2).

However, the switch does not automatically revert to the MSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. A switch also might continue to assign a boundary role to a port when the switch to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches) on the entire switch, you can use the **clear spanning-tree detected-protocols** privileged EXEC command. To restart the protocol migration process on a specific interface, enter the **clear spanning-tree detected-protocols interface** *interface-id* privileged EXEC command.

Configuring MST

These sections describe how to configure MST:

Default MST Configuration

Table 28-7 shows the default MST configuration.

Table 28-7 Default MST Configuration

Feature	Default Setting	
Spanning tree mode	PVST+ (Rapid PVST+ and MST are disabled)	
Switch priority (configurable on a per-CIST port basis)	32768	
Spanning tree port priority (configurable on a per-MST port basis)	128	
Spanning tree port cost (configurable on a per-MST instance port basis)	10-Gigabit Ethernet:	2,000
	Gigabit Ethernet:	20,000
	Fast Ethernet:	200,000
	Ethernet:	2,000,000
Hello time	2 seconds	
Forward-delay time	15 seconds	
Maximum-aging time	20 seconds	
Maximum hop count	20 hops	

MST Configuration Guidelines and Restrictions

When configuring MST, follow these guidelines and restrictions:

- The 802.1s MST standard allows up to 65 MST instances. You can map an unlimited number of VLANs to an MST instance.
- PVST+, rapid PVST+, and MST are supported, but only one version can be active at any time.
- VTP does not propagate the MST configuration. You must manually configure the MST configuration (region name, revision number, and VLAN-to-instance mapping) on each switch within the MST region through the command-line interface (CLI) or SNMP.
- For load balancing across redundant paths in the network to work, all VLAN-to-instance mapping assignments must match; otherwise, all traffic flows on a single link.
- All MST boundary ports must be forwarding for load balancing between a PVST+ and an MST cloud or between a rapid-PVST+ and an MST cloud. For this to occur, the CIST regional root of the MST cloud must be the root of the CST. If the MST cloud consists of multiple MST regions, one of the MST regions must contain the CST root, and all of the other MST regions must have a better path to the root contained within the MST cloud than a path through the PVST+ or rapid-PVST+ cloud.
- Partitioning the network into a large number of regions is not recommended. However, if this situation is unavoidable, we recommend that you partition the switched LAN into smaller LANs interconnected by non-Layer 2 devices.
- Adding or removing VLANs to an existing MST instance will trigger spanning tree recalculation for that MST instance, and the traffic of all the VLANs for that MST instance will be disrupted.

Specifying the MST Region Configuration and Enabling MST

For two or more switches to be in the same MST region, they must have the same VLAN-to-instance mapping, the same configuration revision number, and the same MST name.

A region can have one member or multiple members with the same MST configuration; each member must be capable of processing RSTP BPDUs. There is no limit to the number of MST regions in a network, but each region can only support up to 65 spanning tree instances. You can assign a VLAN to only one spanning tree instance at a time.

To specify the MST region configuration and enable MST, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# spanning-tree mst configuration`	Enters MST configuration mode.
Step 3	`Router(config-mst)# instance instance_id vlan vlan_range`	Maps VLANs to an MST instance. • For *instance_id*, the range is 0 to 4094. • For **vlan** *vlan_range*, the range is 1 to 4094. When you map VLANs to an MST instance, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped. To specify a VLAN range, use a hyphen; for example, **instance 1 vlan 1-63** maps VLANs 1 through 63 to MST instance 1. To specify a VLAN series, use a comma; for example, **instance 1 vlan 10, 20, 30** maps VLANs 10, 20, and 30 to MST instance 1.
Step 4	`Router(config-mst)# name instance_name`	Specifies the instance name. The *name* string has a maximum length of 32 characters and is case sensitive.
Step 5	`Router(config-mst)# revision version`	Specifies the configuration revision number. The range is 0 to 65535.
Step 6	`Router(config-mst)# show pending`	Verifies your configuration by displaying the pending configuration.
Step 7	`Router(config)# exit`	Applies all changes, and return to global configuration mode.
Step 8	`Router(config)# spanning-tree mode mst`	Enables MST and RSTP. **Caution** Changing the spanning tree mode can disrupt traffic because all spanning tree instances are stopped for the previous mode and restarted in the new mode. You cannot run both MST and PVST+ or both MST and rapid PVST+ at the same time.
Step 9	`Router(config)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 10	`Router# show running-config`	Verifies your entries.
Step 11	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return to defaults, do the following:

- To return to the default MST region configuration, use the **no spanning-tree mst configuration** global configuration command.
- To return to the default VLAN-to-instance map, use the **no instance** *instance_id* [**vlan** *vlan_range*] MST configuration command.
- To return to the default name, use the **no name** MST configuration command.
- To return to the default revision number, use the **no revision** MST configuration command.
- To reenable PVST+, use the **no spanning-tree mode** or the **spanning-tree mode pvst** global configuration command.

This example shows how to enter MST configuration mode, map VLANs 10 to 20 to MST instance 1, name the region *region1*, set the configuration revision to 1, display the pending configuration, apply the changes, and return to global configuration mode:

```
Router(config)# spanning-tree mst configuration
Router(config-mst)# instance 1 vlan 10-20
Router(config-mst)# name region1
Router(config-mst)# revision 1
Router(config-mst)# show pending
Pending MST configuration
Name      [region1]
Revision  1
Instances configured 2
Instance  Vlans Mapped
--------  ---------------------
0         1-9,21-4094
1         10-20
-------------------------------

Router(config-mst)# exit
Router(config)#
```

Configuring the Root Bridge

The switch maintains a spanning tree instance for the group of VLANs mapped to it. A switch ID, consisting of the switch priority and the switch MAC address, is associated with each instance. For a group of VLANs, the switch with the lowest switch ID becomes the root bridge.

To configure a switch to become the root bridge, use the **spanning-tree mst** *instance_id* **root** global configuration command to modify the switch priority from the default value (32768) to a significantly lower value so that the switch becomes the root bridge for the specified spanning tree instance. When you enter this command, the switch checks the switch priorities of the root bridges. Because of extended system ID support, the switch sets its own priority for the specified instance to 24576 if this value will cause this switch to become the root bridge for the specified spanning tree instance.

If any root bridge for the specified instance has a switch priority lower than 24576, the switch sets its own priority to 4096 less than the lowest switch priority. (4096 is the value of the least-significant bit of a 4-bit switch priority value as shown in Table 28-1 on page 28-3.)

If your network consists of switches that both do and do not support the extended system ID, it is unlikely that the switch with the extended system ID support will become the root bridge. The extended system ID increases the switch priority value every time the VLAN number is greater than the priority of the connected switches running older software.

The root bridge for each spanning tree instance should be a backbone or distribution switch. Do not configure an access switch as the spanning tree primary root bridge.

Use the **diameter** keyword, which is available only for MST instance 0, to specify the Layer 2 network diameter (that is, the maximum number of Layer 2 hops between any two end stations in the Layer 2 network). When you specify the network diameter, the switch automatically sets an optimal hello time, forward-delay time, and maximum-age time for a network of that diameter, which can significantly reduce the convergence time. You can use the **hello** keyword to override the automatically calculated hello time.

Note With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time with the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

To configure a switch as the root bridge, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config-config)#` **`spanning-tree mst`** *`instance_id`* **`root primary`** [**`diameter`** *`net_diameter`* [**`hello-time`** *`seconds`*]]	(Optional) Configures a switch as the root bridge. • For *instance_id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is 0 to 4094. • (Optional) For **diameter** *net_diameter*, specify the maximum number of Layer 2 hops between any two end stations. The range is 2 to 7. This keyword is available only for MST instance 0. • (Optional) For **hello-time** *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is 1 to 10 seconds; the default is 2 seconds.
Step 3	`Router(config-config)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Router#` **`show spanning-tree mst`** *`instance_id`*	Verifies your entries.
Step 5	`Router#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst** *instance_id* **root** global configuration command.

Configuring a Secondary Root Bridge

When you configure a switch with the extended system ID support as the secondary root, the switch priority is modified from the default value (32768) to 28672. The switch is then likely to become the root bridge for the specified instance if the primary root bridge fails. This is assuming that the other network switches use the default switch priority of 32768 and therefore are unlikely to become the root bridge.

You can execute this command on more than one switch to configure multiple backup root bridges. Use the same network diameter and hello-time values that you used when you configured the primary root bridge with the **spanning-tree mst** *instance_id* **root primary** global configuration command.

To configure a switch as the secondary root bridge, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`spanning-tree mst`** *`instance_id`* **`root secondary`** [**`diameter`** *`net_diameter`* [**`hello-time`** *`seconds`*]]	(Optional) Configures a switch as the secondary root bridge. • For *instance_id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is 0 to 4094. • (Optional) For **diameter** *net_diameter*, specify the maximum number of switches between any two end stations. The range is 2 to 7. This keyword is available only for MST instance 0. • (Optional) For **hello-time** *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is 1 to 10 seconds; the default is 2 seconds. Use the same network diameter and hello-time values that you used when configuring the primary root bridge. See the "Configuring the Root Bridge" section on page 28-41.
Step 3	`Router(config)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Router#` **`show spanning-tree mst`** *`instance_id`*	Verifies your entries.
Step 5	`Router#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst** *instance_id* **root** global configuration command.

Configuring Port Priority

If a loop occurs, MST uses the port priority when selecting an interface to put into the forwarding state. You can assign higher priority values (lower numerical values) to interfaces that you want selected first and lower priority values (higher numerical values) that you want selected last. If all interfaces have the same priority value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

To configure the MST port priority of an interface, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface`** `{{`**`gigabitethernet 1`**`/`*`port`*`} | {`**`port-channel`** *`number`*`}}`	(Optional) Specifies an interface to configure, and enters interface configuration mode.
Step 3	`Router(config-if)#` **`spanning-tree mst`** *`instance_id`* **`port-priority`** *`priority`*	Configures the port priority. • For *instance_id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is 0 to 4094. • For *priority*, the range is 0 to 240 in increments of 16. The default is 128. The lower the number, the higher the priority. The priority values are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240. All other values are rejected.
Step 4	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Router#` **`show spanning-tree mst interface`** *`interface_id`* `or` `Router#` **`show spanning-tree mst`** *`instance_id`*	Verifies your entries.
Step 6	`Router#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Note The **show spanning-tree mst interface** *interface_id* privileged EXEC command displays information only if the port is in a link-up operative state. Otherwise, you can use the **show running-config interface** privileged EXEC command to confirm the configuration.

To return the interface to its default setting, use the **no spanning-tree mst** *instance_id* **port-priority** interface configuration command.

Configuring Path Cost

The MST path cost default value is derived from the media speed of an interface. If a loop occurs, MST uses cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost values that you want selected last. If all interfaces have the same cost value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

To configure the MST cost of an interface, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface`** `{{`**`gigabitethernet 1`**`/`*`port`*`} | {`**`port-channel`** *`number`*`}}`	(Optional) Specifies an interface to configure, and enters interface configuration mode.
Step 3	`Router(config-if)#` **`spanning-tree mst`** *`instance_id`* **`cost`** *`cost`*	Configures the cost. If a loop occurs, MST uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission. • For *instance_id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is 0 to 4094. • For *cost*, the range is 1 to 200000000; the default value is derived from the media speed of the interface.
Step 4	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Router#` **`show spanning-tree mst interface`** *`interface_id`* or `Router#` **`show spanning-tree mst`** *`instance_id`*	Verifies your entries.
Step 6	`Router#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Note The **show spanning-tree mst interface** *interface_id* privileged EXEC command displays information only for ports that are in a link-up operative state. Otherwise, you can use the **show running-config** privileged EXEC command to confirm the configuration.

To return the interface to its default setting, use the **no spanning-tree mst** *instance_id* **cost** interface configuration command.

Configuring the Switch Priority

You can configure the switch priority so that it is more likely that a switch is chosen as the root bridge.

Note Exercise care when using this command. For most situations, we recommend that you use the **spanning-tree mst** *instance_id* **root primary** and the **spanning-tree mst** *instance_id* **root secondary** global configuration commands to modify the switch priority.

To configure the switch priority, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# spanning-tree mst instance_id priority priority`	(Optional) Configures the switch priority. • For *instance_id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is 0 to 4094. • For *priority*, the range is 0 to 61440 in increments of 4096; the default is 32768. The lower the number, the more likely the switch will be chosen as the root bridge. Priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. All other values are rejected.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show spanning-tree mst instance_id`	Verifies your entries.
Step 5	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst** *instance_id* **priority** global configuration command.

Configuring the Hello Time

You can configure the interval between the generation of configuration messages by the root bridge by changing the hello time.

Note Exercise care when using this command. For most situations, we recommend that you use the **spanning-tree mst** *instance_id* **root primary** and the **spanning-tree mst** *instance_id* **root secondary** global configuration commands to modify the hello time.

To configure the hello time for all MST instances, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# spanning-tree mst hello-time seconds`	(Optional) Configures the hello time for all MST instances. The hello time is the interval between the generation of configuration messages by the root bridge. These messages mean that the switch is alive. For *seconds*, the range is 1 to 10; the default is 2.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 4	`Router# show spanning-tree mst`	Verifies your entries.
Step 5	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst hello-time** global configuration command.

Configuring the Forwarding-Delay Time

To configure the forwarding-delay time for all MST instances, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# spanning-tree mst forward-time` *seconds*	(Optional) Configures the forward time for all MST instances. The forward delay is the number of seconds a port waits before changing from its spanning-tree learning and listening states to the forwarding state. For *seconds*, the range is 4 to 30; the default is 15.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show spanning-tree mst`	Verifies your entries.
Step 5	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst forward-time** global configuration command.

Configuring the Transmit Hold Count

To configure the transmit hold count for all MST instances, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# spanning-tree transmit hold-count` *hold_count_value*	Configures the transmit hold count for all MST instances. For *hold_count_value*, the range is 1 to 20; the default is 6.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show spanning-tree mst`	Verifies your entries.
Step 5	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree transmit hold-count** global configuration command.

Configuring the Maximum-Aging Time

To configure the maximum-aging time for all MST instances, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# spanning-tree mst max-age seconds`	(Optional) Configures the maximum-aging time for all MST instances. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration. For *seconds*, the range is 6 to 40; the default is 20.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show spanning-tree mst`	Verifies your entries.
Step 5	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst max-age** global configuration command.

Configuring the Maximum-Hop Count

To configure the maximum-hop count for all MST instances, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# spanning-tree mst max-hops hop_count`	(Optional) Specifies the number of hops in a region before the BPDU is discarded, and the information held for a port is aged. For *hop_count*, the range is 1 to 255; the default is 20.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show spanning-tree mst`	Verifies your entries.
Step 5	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst max-hops** global configuration command.

Specifying the Link Type to Ensure Rapid Transitions

If you connect a port to another port through a point-to-point link and the local port becomes a designated port, the RSTP negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology as described in the "Rapid Convergence" section on page 28-14.

By default, the link type is controlled from the duplex mode of the interface: a full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection. If you have a half-duplex link physically connected point-to-point to a single port on a remote switch running MST, you can override the default setting of the link type and enable rapid transitions to the forwarding state.

To override the default link-type setting, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface** {{**gigabitethernet 1**/*port*} | {**port-channel** *number*}}	(Optional) Specifies an interface to configure, and enters interface configuration mode.
Step 3	Router(config)# **spanning-tree link-type point-to-point**	Specifies that the link type of a port is point-to-point.
Step 4	Router(config)# **end**	Returns to privileged EXEC mode.
Step 5	Router# **show spanning-tree mst interface** *interface_id*	Verifies your entries.
Step 6	Router# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return the port to its default setting, use the **no spanning-tree link-type** interface configuration command.

Designating the Neighbor Type

A topology could contain both prestandard and 802.1s standard compliant devices. By default, ports can automatically detect prestandard devices, but they can still receive both standard and prestandard BPDUs. When there is a mismatch between a device and its neighbor, only the CIST runs on the interface.

You can choose to set a port to send only prestandard BPDUs. The prestandard flag appears in all the **show** commands, even if the port is in STP compatibility mode.

To override the default link-type setting, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface** {{**gigabitethernet 1**/*port*} | {**port-channel** *number*}}	(Optional) Specifies an interface to configure, and enters interface configuration mode.
Step 3	Router(config)# **spanning-tree mst pre-standard**	Specifies that the port can send only prestandard BPDUs.
Step 4	Router(config)# **end**	Returns to privileged EXEC mode.
Step 5	Router# **show spanning-tree mst interface** *interface_id*	Verifies your entries.
Step 6	Router# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return the port to its default setting, use the **no spanning-tree mst prestandard** interface configuration command.

Restarting the Protocol Migration Process

A switch running MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D switches. If this switch receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MST switch also can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an RST BPDU (Version 2).

However, the switch does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D switch has been removed from the link unless the 802.1D switch is the designated switch. A switch also might continue to assign a boundary role to a port when the switch to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches) on the switch, use the **clear spanning-tree detected-protocols** privileged EXEC command.

To restart the protocol migration process on a specific interface, use the **clear spanning-tree detected-protocols interface** *interface_id* privileged EXEC command.

Displaying the MST Configuration and Status

To display the spanning-tree status, use one or more of the privileged EXEC commands that are described in Table 28-8.

Table 28-8 Commands for Displaying MST Status

Command	Purpose
show spanning-tree mst configuration	Displays the MST region configuration.
show spanning-tree mst configuration digest	Displays the MD5 digest included in the current MSTCI.
show spanning-tree mst *instance_id*	Displays MST information for the specified instance.
show spanning-tree mst interface *interface_id*	Displays MST information for the specified interface.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Optional STP Features

This chapter describes how to configure optional STP features.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Note For information on configuring the Spanning Tree Protocol (STP), see Chapter 28, "Configuring STP and MST."

Understanding the Optional STP Features

The optional features of the Spanning Tree Protocol (STP) enhance loop prevention, protect against some possible user misconfigurations, and provide better control over the protocol parameters. Although similar functionality may be incorporated into the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard, we recommend using these extensions. With the exception of PVST Simulation, all of these extensions can be used with both Rapid PVST+ and MST. PVST Simulation operates only with MST.

These sections describe the optional features:

Understanding STP Port Types

Beginning with Cisco IOS Release 12.2(33)SXI, you can specifically configure a spanning tree port as either an edge port, a network port, or a normal port. The port type determines the behavior of the port with respect to STP extensions.

An edge port, which is connected to a Layer 2 host, can be either an access port or a trunk port.

Note If you configure a port connected to a Layer 2 switch or bridge as an edge port, you might create a bridging loop.

A network port is connected only to a Layer 2 switch or bridge.

Note If you mistakenly configure a port that is connected to a Layer 2 host as a spanning tree network port, the port will automatically move into the blocking state.

The default spanning tree port type is normal, meaning only that its topology is not specified.

In releases earlier than Cisco IOS Release 12.2(33)SXI, configuring a port with PortFast implies that the port is an edge port; otherwise, the port is considered to be a normal port.

Understanding PortFast

STP PortFast causes a Layer 2 LAN port configured as an access port to enter the forwarding state immediately, bypassing the listening and learning states. You can use PortFast on Layer 2 access ports connected to a single workstation or server to allow those devices to connect to the network immediately, instead of waiting for STP to converge. Interfaces connected to a single workstation or server should not receive bridge protocol data units (BPDUs). When configured for PortFast, a port is still running the spanning tree protocol. A PortFast enabled port can immediately transition to the blocking state if necessary (this could happen on receipt of a superior BPDU). PortFast can be enabled on trunk ports. PortFast can have an operational value that is different from the configured value.

Note Beginning with Cisco IOS Release 12.2(33)SXI, PortFast options are edge and normal. There is no difference between edge ports and PortFast ports.

Caution Because the purpose of PortFast is to minimize the time that access ports must wait for STP to converge, it should only be used on access (edge) ports. If you enable PortFast on a port connected to a switch (a network port), you might create a temporary bridging loop.

Understanding Bridge Assurance

With Cisco IOS Release 12.2(33)SXI and later releases, you can use Bridge Assurance to protect against certain problems that can cause bridging loops in the network. Specifically, you use Bridge Assurance to protect against a unidirectional link failure or other software failure and a device that continues to forward data traffic when it is no longer running the spanning tree algorithm.

Note Bridge Assurance is supported only by Rapid PVST+ and MST.

Bridge Assurance is enabled by default and can only be disabled globally. Also, Bridge Assurance is enabled only on spanning tree network ports that are point-to-point links. Finally, both ends of the link must have Bridge Assurance enabled. If the device on one side of the link has Bridge Assurance enabled and the device on the other side either does not support Bridge Assurance or does not have this feature enabled, the connecting port is blocked.

With Bridge Assurance enabled, BPDUs are sent out on all operational network ports, including alternate and backup ports, for each hello time period. If the port does not receive a BPDU for a specified period, the port moves into an inconsistent state (blocking). and is not used in the root port calculation. Once that port receives a BPDU, it resumes the normal spanning tree transitions.

Figure 29-1 shows a normal STP topology, and Figure 29-2 demonstrates a potential network problem when the device fails and you are not running Bridge Assurance.

Figure 29-1 Network with Normal STP Topology

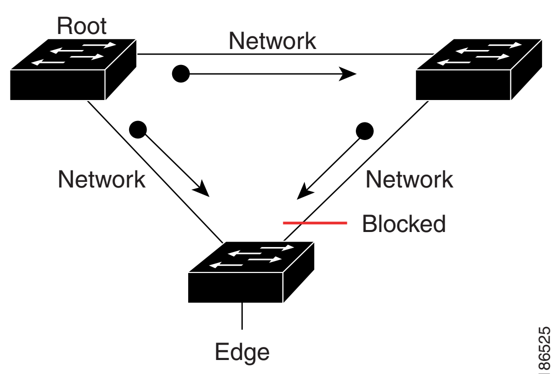


Figure 29-2 Network Problem without Running Bridge Assurance

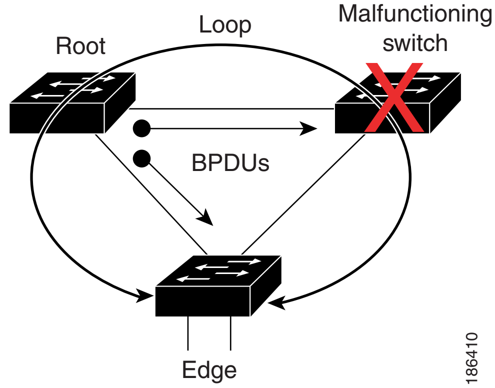


Figure 29-3 shows the network with Bridge Assurance enabled, and the STP topology progressing normally with bidirectional BPDUs issuing from every STP network port. Figure 29-4 shows how the potential network problem shown in Figure 29-2 does not happen when you have Bridge Assurance enabled on your network.

Figure 29-3 Network STP Topology Running Bridge Assurance

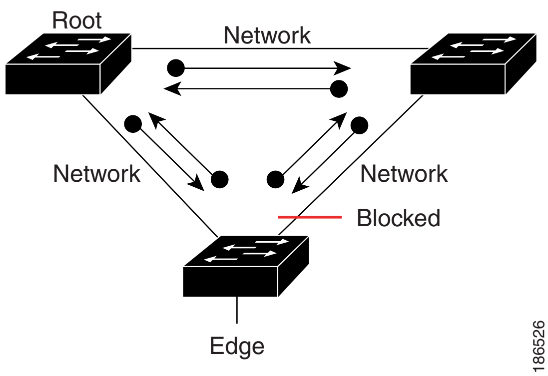
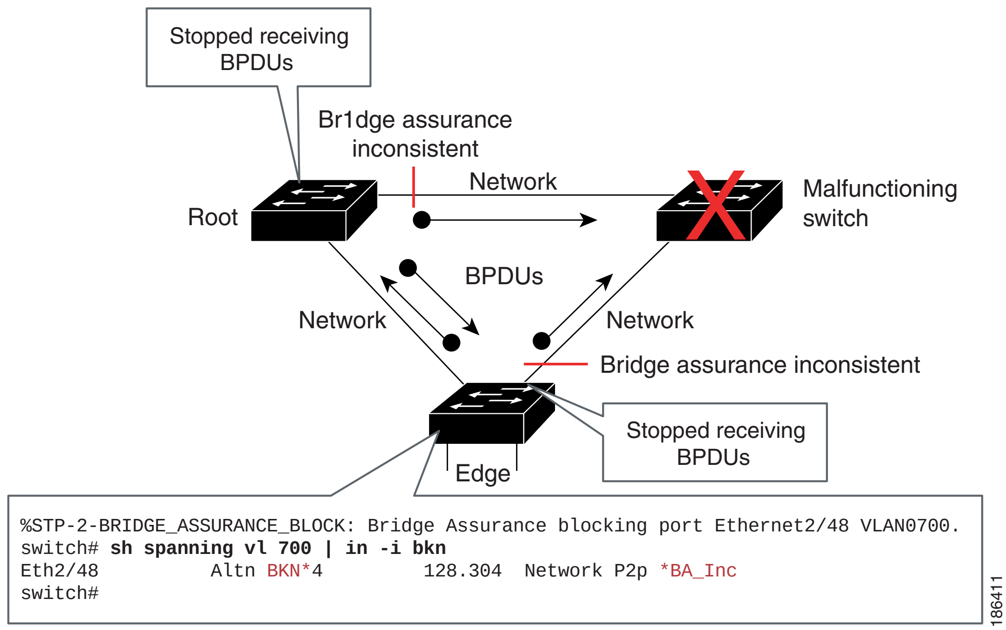


Figure 29-4 Network Problem Averted with Bridge Assurance Enabled



When using Bridge Assurance, follow these guidelines:

- Bridge Assurance runs only on point-to-point spanning tree network ports. You must configure each side of the link for this feature.
- We recommend that you enable Bridge Assurance throughout your network.

Understanding BPDU Guard

BPDU Guard complements the functionality of PortFast. On PortFast-enabled ports, BPDU Guard provides the protection against Layer 2 loops that STP cannot provide when STP PortFast is enabled.

In a valid configuration, PortFast Layer 2 LAN interfaces (edge ports) do not receive BPDUs. When enabled on a port, BPDU Guard shuts down a port that receives a BPDU. When configured globally, BPDU Guard is only effective on ports in the operational PortFast (edge) state. Reception of a BPDU by a PortFast Layer 2 LAN interface signals an invalid configuration, such as connection of an unauthorized device. BPDU Guard provides a secure response to invalid configurations, because the administrator must manually put the Layer 2 LAN interface back in service. BPDU Guard can be configured at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the PortFast configuration.

Note When enabled globally, BPDU Guard applies to all interfaces that are in an operational PortFast (edge) state.

Understanding PortFast BPDU Filtering

PortFast BPDU filtering allows the administrator to prevent the system from sending or even receiving BPDUs on specified ports.

Note Beginning with Cisco IOS Release 12.2(33)SXI, PortFast BPDU Filtering is known as PortFast Edge BPDU Filtering.

When configured globally, PortFast BPDU filtering applies to all operational PortFast (edge) ports. Ports in an operational PortFast state are supposed to be connected to hosts, which typically drop BPDUs. If an operational PortFast port receives a BPDU, it immediately loses its operational PortFast status and becomes a normal port. In that case, PortFast BPDU filtering is disabled on this port and STP resumes sending BPDUs on this port.

PortFast BPDU filtering can also be configured on a per-port basis. When PortFast BPDU filtering is explicitly configured on a port, it does not send any BPDUs and drops all BPDUs it receives.

Caution Explicitly configuring PortFast BPDU filtering on a port that is not connected to a host can result in bridging loops, as the port will ignore any BPDU it receives and will go to a forwarding state.

When you enable PortFast BPDU filtering globally and set the port configuration as the default for PortFast BPDU filtering (see the "Enabling PortFast BPDU Filtering" section on page 29-15), then PortFast enables or disables PortFast BPDU filtering.

If the port configuration is not set to default, then the PortFast configuration will not affect PortFast BPDU filtering. Table 29-1 lists all the possible PortFast BPDU filtering combinations. PortFast BPDU filtering allows access ports to move directly to the forwarding state as soon as the end hosts are connected.

Table 29-1 PortFast BPDU Filtering Port Configurations

Per-Port Configuration	Global Configuration	PortFast State	PortFast BPDU Filtering State
Default	Enable	Enable	Enable[1]
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

1. The port transmits at least 10 BPDUs. If this port receives any BPDUs, then PortFast and PortFast BPDU filtering are disabled.

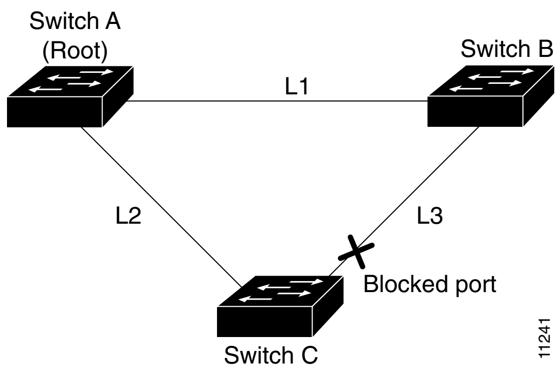
Understanding UplinkFast

UplinkFast provides fast convergence after a direct link failure and achieves load balancing between redundant Layer 2 links using uplink groups. An uplink group is a set of Layer 2 LAN interfaces (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

Note UplinkFast is most useful in wiring-closet switches. This feature may not be useful for other types of applications.

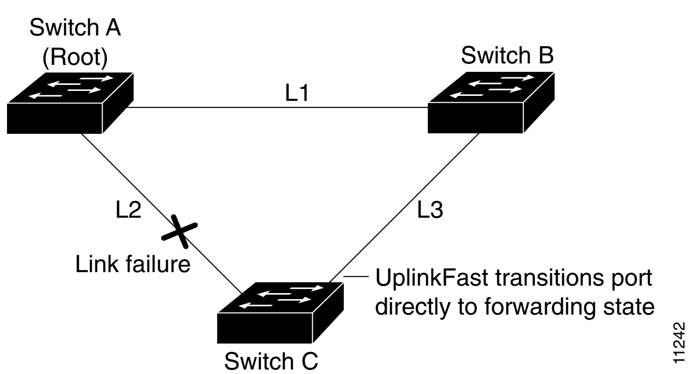
Figure 29-5 shows an example topology with no link failures. Switch A, the root bridge, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 29-5 UplinkFast Example Before Direct Link Failure



If Switch C detects a link failure on the currently active link L2 on the root port (a *direct* link failure), UplinkFast unblocks the blocked port on Switch C and transitions it to the forwarding state without going through the listening and learning states, as shown in Figure 29-6. This switchover takes approximately one to five seconds.

Figure 29-6 UplinkFast Example After Direct Link Failure



Understanding BackboneFast

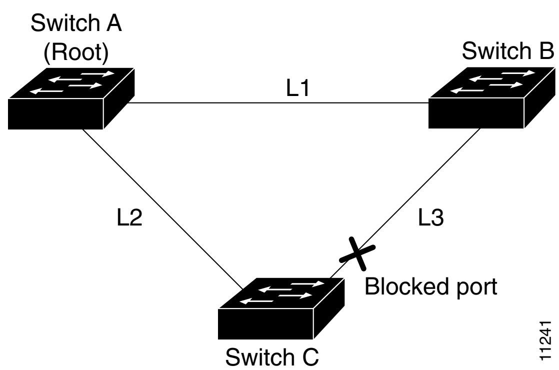
BackboneFast is initiated when a root port or blocked port on a network device receives inferior BPDUs from its designated bridge. An inferior BPDU identifies one network device as both the root bridge and the designated bridge. When a network device receives an inferior BPDU, it indicates that a link to which the network device is not directly connected (an *indirect* link) has failed (that is, the designated bridge has lost its connection to the root bridge). Under normal STP rules, the network device ignores inferior BPDUs for the configured maximum aging time, as specified by the STP **max-age** command.

The network device tries to determine if it has an alternate path to the root bridge. If the inferior BPDU arrives on a blocked port, the root port and other blocked ports on the network device become alternate paths to the root bridge. (Self-looped ports are not considered alternate paths to the root bridge.) If the inferior BPDU arrives on the root port, all blocked ports become alternate paths to the root bridge. If the inferior BPDU arrives on the root port and there are no blocked ports, the network device assumes that it has lost connectivity to the root bridge, causes the maximum aging time on the root to expire, and becomes the root bridge according to normal STP rules.

If the network device has alternate paths to the root bridge, it uses these alternate paths to transmit a new kind of Protocol Data Unit (PDU) called the Root Link Query PDU. The network device sends the Root Link Query PDU out all alternate paths to the root bridge. If the network device determines that it still has an alternate path to the root, it causes the maximum aging time to expire on the ports on which it received the inferior BPDU. If all the alternate paths to the root bridge indicate that the network device has lost connectivity to the root bridge, the network device causes the maximum aging times on the ports on which it received an inferior BPDU to expire. If one or more alternate paths can still connect to the root bridge, the network device makes all ports on which it received an inferior BPDU its designated ports and moves them out of the blocking state (if they were in the blocking state), through the listening and learning states, and into the forwarding state.

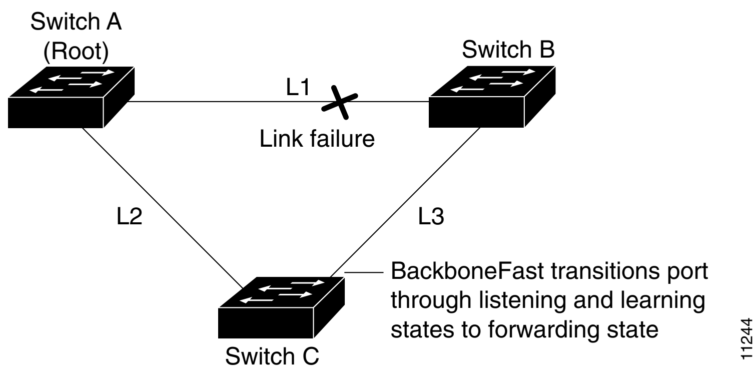
Figure 29-7 shows an example topology with no link failures. Switch A, the root bridge, connects directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that connects directly to Switch B is in the blocking state.

Figure 29-7 BackboneFast Example Before Indirect Link Failure



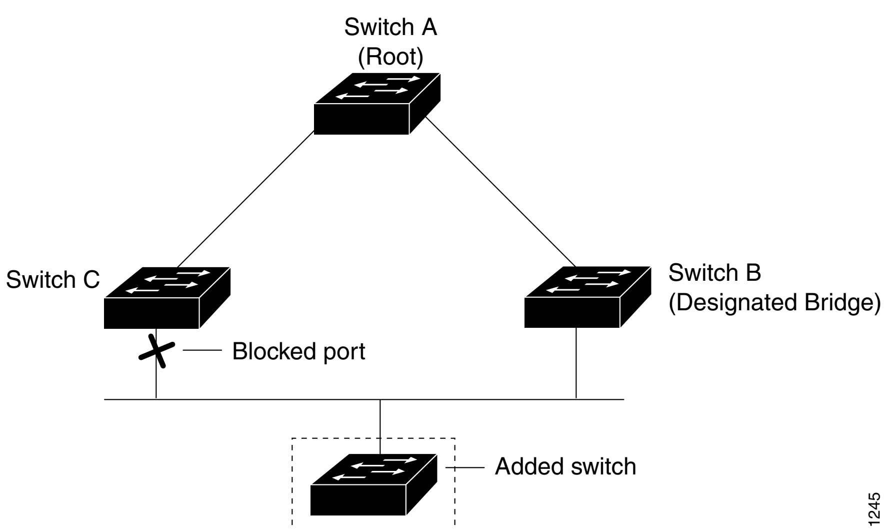
If link L1 fails, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root bridge over L1, it detects the failure and elects itself the root and begins sending BPDUs to Switch C indicating itself as the root. When Switch C receives the inferior BPDUs from Switch B, Switch C infers that an indirect failure has occurred. At that point, BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the maximum aging time for the port to expire. BackboneFast then transitions the Layer 2 LAN interface on Switch C to the forwarding state, providing a path from Switch B to Switch A. This switchover takes approximately 30 seconds, twice the Forward Delay time if the default Forward Delay time of 15 seconds is set. Figure 29-8 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 29-8 BackboneFast Example After Indirect Link Failure



If a new network device is introduced into a shared-medium topology as shown in Figure 29-9, BackboneFast is not activated because the inferior BPDUs did not come from the recognized designated bridge (Switch B). The new network device begins sending inferior BPDUs that indicate that it is the root bridge. However, the other network devices ignore these inferior BPDUs and the new network device learns that Switch B is the designated bridge to Switch A, the root bridge.

Figure 29-9 Adding a Network Device in a Shared-Medium Topology



Understanding EtherChannel Guard

EtherChannel guard detects a misconfigured EtherChannel when interfaces on the switch are configured as an EtherChannel while interfaces on the other device are not or when not all the interfaces on the other device are in the same EtherChannel.

In response to misconfiguration detected on the other device, EtherChannel guard puts interfaces on the switch into the errdisabled state.

Understanding Root Guard

The STP root guard feature prevents a port from becoming root port or blocked port. If a port configured for root guard receives a superior BPDU, the port immediately goes to the root-inconsistent (blocked) state.

Understanding Loop Guard

Loop guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link. When enabled globally, the loop guard applies to all point-to-point ports on the system. Loop guard detects root ports and blocked ports and ensures that they keep receiving BPDUs from their designated port on the segment. If a loop guard enabled root or blocked port stop a receiving BPDUs from its designated port, it transitions to the loop-inconsistent blocking state, assuming there is a physical link error on this port. The port recovers from this loop-inconsistent state as soon as it receives a BPDU.

You can enable loop guard on a per-port basis. When you enable loop guard, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable loop guard, it is disabled for the specified ports. Disabling loop guard moves all loop-inconsistent ports to the listening state.

If you enable loop guard on a channel and the first link becomes unidirectional, loop guard blocks the entire channel until the affected port is removed from the channel. Figure 29-10 shows loop guard in a triangle switch configuration.

Figure 29-10 Triangle Switch Configuration with Loop Guard

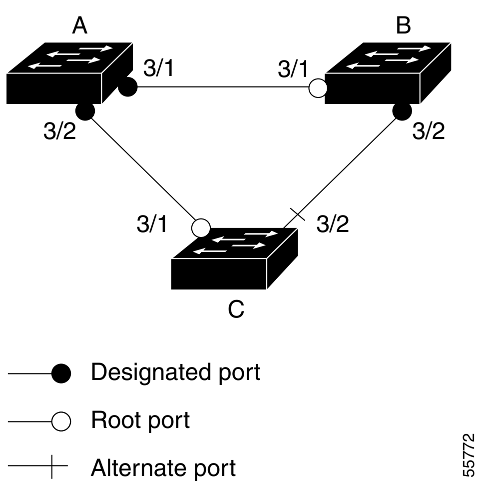


Figure 29-10 illustrates the following configuration:

- Switches A and B are distribution switches.
- Switch C is an access switch.
- Loop guard is enabled on ports 3/1 and 3/2 on Switches A, B, and C.

Enabling loop guard on a root switch has no effect but provides protection when a root switch becomes a nonroot switch.

When using loop guard, follow these guidelines:

- You cannot enable loop guard on PortFast-enabled ports.
- You cannot enable loop guard if root guard is enabled.

Loop guard interacts with other features as follows:

- Loop guard does not affect the functionality of UplinkFast or BackboneFast.
- Enabling loop guard on ports that are not connected to a point-to-point link will not work.
- Root guard forces a port to be always designated as the root port. Loop guard is effective only if the port is a root port or an alternate port. You cannot enable loop guard and root guard on a port at the same time.
- Loop guard uses the ports known to spanning tree. Loop guard can take advantage of logical ports provided by the Port Aggregation Protocol (PAgP). However, to form a channel, all the physical ports grouped in the channel must have compatible configurations. PAgP enforces uniform configurations of root guard or loop guard on all the physical ports to form a channel.

  These caveats apply to loop guard:

  – Spanning tree always chooses the first operational port in the channel to send the BPDUs. If that link becomes unidirectional, loop guard blocks the channel, even if other links in the channel are functioning properly.
  – If a set of ports that are already blocked by loop guard are grouped together to form a channel, spanning tree loses all the state information for those ports and the new channel port may obtain the forwarding state with a designated role.
  – If a channel is blocked by loop guard and the channel breaks, spanning tree loses all the state information. The individual physical ports may obtain the forwarding state with the designated role, even if one or more of the links that formed the channel are unidirectional.

  Note You can enable UniDirectional Link Detection (UDLD) to help isolate the link failure. A loop may occur until UDLD detects the failure, but loop guard will not be able to detect it.

- Loop guard has no effect on a disabled spanning tree instance or a VLAN.

Understanding PVST Simulation

MST interoperates with Rapid PVST+ with no need for user configuration. The PVST simulation feature enables this seamless interoperability.

Note PVST simulation is enabled by default when you enable MST. That is, by default, all interfaces on the device interoperate between MST and Rapid PVST+. The ability to disable PVST simulation is introduced in Cisco IOS Release 12.2(33)SXI.

You may want to control the connection between MST and Rapid PVST+ to protect against accidentally connecting an MST-enabled port to a port enabled to run Rapid PVST+. Because Rapid PVST+ is the default STP mode, you may encounter many Rapid PVST+ connections.

Disabling Rapid PVST+ simulation, which can be done per port or globally for the entire device, moves the MST-enabled port to the PVST peer inconsistent (blocking) state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving Shared Spanning Tree Protocol (SSTP) BPDUs, and then the port resumes the normal STP transition process.

The root bridge for all STP instances must all be in either the MST region or the Rapid PVST+ side. If the root bridge for all STP instances are not on one side or the other, the software moves the port into a PVST simulation-inconsistent state.

Note We recommend that you put the root bridge for all STP instances in the MST region.

Configuring the Optional STP Features

This section includes the following topics:

Enabling PortFast

Caution Use the PortFast feature only when connecting a single end station to a Layer 2 port. Otherwise, you might create a network loop.

Enabling PortFast on a Layer 2 Port

To enable PortFast on a Layer 2 port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}	Selects a port to configure.

	Command	Purpose
Step 2	`Router(config-if)# no spanning-tree portfast`	Configures the port as a normal port.
	Cisco IOS Release 12.2(33)SXI and later releases: `Router(config-if)# spanning-tree portfast edge [trunk]`	Enables edge behavior on a Layer 2 access port connected to a single workstation or server. The **trunk** keyword enables edge behavior on a trunk port.
	`Router(config-if)# spanning-tree portfast network`	Configures the port as a network port. Bridge Assurance, if enabled globally, will be enabled on the port.
	`Router(config-if)# spanning-tree portfast edge disable`	Disables edge behavior on the port.
	Earlier releases: `Router(config-if)# spanning-tree portfast [trunk]`	Enables PortFast on a Layer 2 access port connected to a single workstation or server.
	`Router(config-if)# spanning-tree portfast disable`	Disables edge behavior on the port. Port will behave as a normal port.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show running interface {type[1] slot/port} | {port-channel port_channel_number}`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable and verify PortFast on Fast Ethernet interface 5/8 in Cisco IOS Release 12.2(33)SXI and later releases:

```
Router# configure terminal
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree portfast edge
Router(config-if)# end
Router#
Router# show running-config interface fastethernet 5/8
Building configuration...

Current configuration:
!
interface FastEthernet5/8
 no ip address
 switchport
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast edge
end
Router#
```

This example shows how to enable and verify PortFast on Fast Ethernet interface 5/8 in releases earlier than Cisco IOS Release 12.2(33)SXI:

```
Router# configure terminal
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree portfast
Router(config-if)# end
Router#
Router# show running-config interface fastethernet 5/8
Building configuration...

Current configuration:
!
interface FastEthernet5/8
 no ip address
```

```
 switchport
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast
end
Router#
```

Configuring the PortFast Default State

To set and verify the default PortFast configuration, perform this task:

	Command	Purpose
Step 1	**Cisco IOS Release 12.2(33)SXI and later releases:** Router(config)# **spanning-tree portfast** [**edge** | **network** | **normal**] **default**	Configures the default state for all switch access ports to be edge, network, or normal. Bridge Assurance will be enabled on all network access ports by default.
	Earlier releases: Router(config)# **[no] spanning-tree portfast default**	Configures the default state for all switch access ports to be edge. Use the **no** prefix to configure the default state to be normal.
Step 2	Router(config-if)# **end**	Exits configuration mode.
Step 3	Router# **show spanning-tree summary totals**	Verifies the global configuration.
Step 4	Router# **show spanning-tree interface** {*type*[1] *slot/port*} **detail**	Verifies the effect on a specific port.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure the default switch access port state to be edge in Cisco IOS Release 12.2(33)SXI and later releases:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# spanning-tree portfast edge default
Router(config)# ^Z
```

This example shows how to configure the default switch access port state to be edge in releases earlier than Cisco IOS Release 12.2(33)SXI:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# spanning-tree portfast default
Router(config)# ^Z
```

Enabling Bridge Assurance

Release 12.2(33)SXI and later releases support Bridge Assurance. By default, Bridge Assurance is enabled on all network ports on the switch. Disabling Bridge Assurance causes all configured network ports to behave as normal spanning tree ports. To enable or disable Bridge Assurance globally, perform this task:

Command	Purpose
Router(config)# **spanning-tree bridge assurance**	Enables Bridge Assurance on all network ports on the switch.

This example shows how to enable PortFast Bridge Assurance on all network ports on the switch, and how to configure a network port:

```
Router(config)# spanning-tree bridge assurance
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree portfast network
Router(config-if)# ^Z
```

Enabling PortFast BPDU Filtering

These sections describe how to configure PortFast BPDU filtering.

Enabling PortFast BPDU Filtering Globally

To enable PortFast BPDU filtering globally, perform this task:

	Command	Purpose

Step 1	**Cisco IOS Release 12.2(33)SXI and later releases:**
	`Router(config)# spanning-tree portfast edge bpdufilter default`
	Earlier releases:
	`Router(config)# spanning-tree portfast bpdufilter default`
		Enables BPDU filtering globally by default on all edge ports of the switch. Use the **no** prefix to disable BPDU filtering by default on all edge ports of the switch.

Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show spanning-tree summary totals`	Verifies the configuration.

BPDU filtering is set to default on each edge port. This example shows how to enable PortFast BPDU filtering on the port and verify the configuration in PVST+ mode:

```
Router(config)# spanning-tree portfast edge bpdufilter default
Router(config)# ^Z

Router# show spanning-tree summary totals
Root bridge for: Bridge VLAN0025
EtherChannel misconfiguration guard is enabled
Extended system ID                  is enabled
PortFast Edge BPDU Guard Default    is disabled
Portfast Edge BPDU Filter Default   is enabled
Portfast Default                    is edge
Bridge Assurance                    is enabled
Loopguard                           is disabled
UplinkFast                          is disabled
BackboneFast                        is disabled
Pathcost method used is long

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0         0         0        3          3
Router#
```

Enabling PortFast BPDU Filtering on a Nontrunking Port

To enable or disable PortFast BPDU filtering on a nontrunking port, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {*type*[1] *slot/port*}	Selects the interface to configure.
Step 2	Router(config-if)# **spanning-tree bpdufilter** [**enable** | **disable**]	Enables or disables BPDU filtering on the port.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show spanning-tree interface** {*type* *slot/port*}	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable PortFast BPDU filtering on a nontrunking port:

```
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree bpdufilter enable
Router(config-if)# ^Z

Router# show spanning-tree interface fastEthernet 4/4

Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0010         Desg FWD 1000      160.196  Edge P2p

Router# show spanning-tree interface fastEthernet 4/4 detail
 Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   BPDU:sent 0, received 0
Router#
```

Enabling BPDU Guard

These sections describe how to enable BPDU Guard:

- Enabling BPDU Guard Globally, page 29-17
- Enabling BPDU Guard on a Port, page 29-18

Enabling BPDU Guard Globally

To enable BPDU Guard globally, perform this task:

	Command	Purpose
Step 1	Cisco IOS Release 12.2(33)SXI and later releases: `Router(config)# `**`spanning-tree portfast edge bpduguard default`** Earlier releases: `Router(config)# `**`spanning-tree portfast bpduguard default`**	Enables BPDU Guard globally by default on all edge ports of the switch.
Step 2	`Router(config)# `**`end`**	Exits configuration mode.
Step 3	`Router# `**`show spanning-tree summary totals`**	Verifies the configuration.

This example shows how to enable BPDU Guard:

```
Router# configure terminal
Router(config)# spanning-tree portfast edge bpduguard default
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary totals
Root bridge for: Bridge VLAN0025
EtherChannel misconfiguration guard is enabled
Extended system ID                  is enabled
PortFast Edge BPDU Guard Default    is enabled
Portfast Edge BPDU Filter Default   is disabled
Portfast Default                    is edge
Bridge Assurance                    is enabled
Loopguard                           is disabled
UplinkFast                          is disabled
BackboneFast                        is disabled
Pathcost method used is long

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0        0         0        3          3
Router#
```

Enabling BPDU Guard on a Port

To enable BPDU Guard on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}	Selects a port to configure.
Step 2	Router(config-if)# **spanning-tree bpduguard enable**	Enables BPDU Guard on the port.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show spanning-tree** Router# **show running interface** {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable BPDU Guard:

```
Router# configure terminal
Router(config)# spanning-tree portfast edge bpduguard default
Router(config)# end
Router#
```

Enabling UplinkFast

UplinkFast increases the bridge priority to 49152 and adds 3000 to the STP port cost of all Layer 2 LAN ports on the switch, decreasing the probability that the switch will become the root bridge. UplinkFast cannot be enabled on VLANs that have been configured for bridge priority. To enable UplinkFast on a VLAN with bridge priority configured, restore the bridge priority on the VLAN to the default value by entering a **no spanning-tree vlan** *vlan_ID* **priority** command in global configuration mode.

Note When you enable UplinkFast, it affects all VLANs on the switch. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

	Command	Purpose
Step 1	Router(config)# **spanning-tree uplinkfast** Router(config)# **spanning-tree uplinkfast** [**max-update-rate** *max_update_rate*]	Enables UplinkFast. Enables UplinkFast with an update rate in seconds.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show spanning-tree vlan** *vlan_ID*	Verifies that UplinkFast is enabled.

This example shows how to enable UplinkFast:

```
Router# configure terminal
Router(config)# spanning-tree uplinkfast
Router(config)# exit
Router#
```

This example shows how to enable UplinkFast with an update rate of 400 packets per second:

```
Router# configure terminal
Router(config)# spanning-tree uplinkfast
Router(config)# spanning-tree uplinkfast max-update-rate 400
Router(config)# exit
Router#
```

This example shows how to verify that UplinkFast is enabled:

```
Router# show spanning-tree uplinkfast
UplinkFast is enabled
Router#
```

Enabling BackboneFast

Note BackboneFast operates correctly only when enabled on all network devices in the network. BackboneFast is not supported on Token Ring VLANs. This feature is supported for use with third-party network devices.

To enable BackboneFast, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree backbonefast`	Enables BackboneFast.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show spanning-tree vlan` *vlan_ID*	Verifies that BackboneFast is enabled.

This example shows how to enable BackboneFast:

```
Router# configure terminal
Router(config)# spanning-tree backbonefast
Router(config)# end
Router#
```

This example shows how to verify that BackboneFast is enabled:

```
Router# show spanning-tree backbonefast
BackboneFast is enabled

BackboneFast statistics
-----------------------
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)     : 0
Number of RLQ request PDUs received (all VLANs)   : 0
Number of RLQ response PDUs received (all VLANs)  : 0
Number of RLQ request PDUs sent (all VLANs)       : 0
Number of RLQ response PDUs sent (all VLANs)      : 0
Router#
```

Enabling EtherChannel Guard

To enable EtherChannel guard, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`spanning-tree etherchannel guard misconfig`**	Enables EtherChannel guard.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.
Step 3	`Router#` **`show spanning-tree summary	include EtherChannel`**	Verifies that EtherChannel guard is enabled.

This example shows how to enable EtherChannel guard:

```
Router# configure terminal
Router(config)# spanning-tree etherchannel guard misconfig
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary | include EtherChannel
EtherChannel misconfiguration guard is enabled
```

To display the interfaces that are in the errdisable state, enter the **show interface status err-disable** command.

After the misconfiguration has been cleared, interfaces in the errdisable state might automatically recover. To manually return a port to service, enter a **shutdown** and then a **no shutdown** command for the interface.

Enabling Root Guard

To enable root guard, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{`*`type`*[1] *`slot/port`*`}	{`**`port-channel`** *`port_channel_number`*`}`	Selects a port to configure.
Step 2	`Router(config-if)#` **`spanning-tree guard root`**	Enables root guard.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show spanning-tree`** `Router#` **`show running interface`** `{`*`type`*[1] *`slot/port`*`}	{`**`port-channel`** *`port_channel_number`*`}`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To display ports that are in the root-inconsistent state, enter the **show spanning-tree inconsistentports** command.

Enabling Loop Guard

To enable loop guard globally on the switch, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`spanning-tree loopguard default`**	Enables loop guard globally on the switch.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.
Step 3	`Router#` **`show spanning-tree interface`** {*type*[1] *slot/port*} **detail**	Verifies the configuration impact on a port.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable loop guard globally:

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# spanning-tree loopguard default
Router(config)# ^Z

Router# show spanning-tree interface fastEthernet 4/4 detail
 Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   Loop guard is enabled by default on the port
   BPDU:sent 0, received 0
```

To enable loop guard on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}	Selects a port to configure.
Step 2	`Router(config-if)#` **`spanning-tree guard loop`**	Configures loop guard.
Step 3	`Router(config)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show spanning-tree interface`** {*type* *slot/port*} **detail**	Verifies the configuration impact on that port.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable loop guard:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree guard loop
Router(config-if)# ^Z
```

This example shows how to verify the configuration:

```
Router# show spanning-tree interface fastEthernet 4/4 detail
 Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   Loop guard is enabled on the port
   BPDU:sent 0, received 0
Router#
```

Configuring PVST Simulation

Note PVST simulation is enabled by default so that all interfaces on the device interoperate between MST and Rapid PVST+. The ability to disable PVST simulation is introduced in Cisco IOS Release 12.2(33)SXI.

To prevent an accidental connection to a device that does not run MST as the default STP mode, you can disable PVST simulation. If you disable PVST simulation, the MST-enabled port moves to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving BPDUs, and then the port resumes the normal STP transition process.

To enable or disable PVST simulation globally, enter the command using the **global** keyword, as shown in the following task:

Command	Purpose
`Router(config)#` **`spanning-tree mst simulate pvst global`**	Enables all ports to automatically interoperate with a connected device that is running in Rapid PVST+ mode. The default is enabled; all interfaces will operate seamlessly between Rapid PVST+ and MST.

To override the global PVST simulation setting for a port, enter the command in the interface command mode, as shown in the following task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** {*type*[1] *slot/port*}	Selects a port to configure.
Step 2	`Router(config-if)#` **`spanning-tree mst simulate pvst`**	Enables this interface to automatically interoperate with a connected device that is running in Rapid PVST+ mode.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to prevent the switch from automatically interoperating with a connecting device that is running Rapid PVST+:

```
Router(config)# no spanning-tree mst simulate pvst global
```

This example shows how to prevent a port from automatically interoperating with a connecting device that is running Rapid PVST+:

```
Router(config)# interface gi3/13
Router(config-if)# spanning-tree mst simulate pvst disable
```

Verifying the Optional STP Features

This section includes the following topics:

- Using the show spanning-tree Commands, page 29-23
- Examples in Release 12.2(33)SXI and Later Releases, page 29-23
- Examples in Releases Earlier Than Release 12.2(33)SXI, page 29-26

Using the show spanning-tree Commands

You can view spanning tree status and configuration information, both global and port-level, using the **show spanning-tree** commands described in this section.

To view spanning tree status and configuration information, perform one of the following tasks:

Command	Purpose
Router# **show spanning-tree**	Displays information about the spanning tree, including protocol type and port types.
Router# **show spanning-tree summary**	Displays a summary of the spanning tree feature settings and the spanning tree states of the VLANs.
Router# **show spanning-tree summary totals**	Displays a summary of the spanning tree feature settings and totals of the VLAN states.
Router# **show spanning-tree interface** {*type*[1] *slot/port*} **detail**	Displays the spanning tree status details of an interface.
Router# **show spanning-tree interface** {*type*[1] *slot/port*} **portfast edge**	Displays the spanning tree portfast edge interface operational state for all the instances. In releases earlier than Cisco IOS Release 12.2(33)SXI, the **edge** keyword is not used.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Examples in Release 12.2(33)SXI and Later Releases

Note The output of the **show spanning-tree** commands in releases earlier than Release 12.2(33)SXI differs from the output in Release 12.2(33)SXI and later releases. The differences are shown in bold in the examples in this section.

This example displays the spanning-tree status with Bridge Assurance enabled but in the inconsistent state:

```
Router# show spanning-tree
VLAN0010
  Spanning tree enabled protocol rstp
  Root ID    Priority    32778
             Address     0002.172c.f400
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-ext 10)
             Address     0002.172c.f400
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi3/14           Desg BKN*4         128.270  Network, P2p *BA_Inc
Router#
```

The following inconsistency messages can be appended to the Type field:

- *BA_Inc—Indicates that Bridge Assurance is in the inconsistent state.
- *PVST_Peer_Inc—Indicates that the port is in a peer type Inconsistent state.
- Dispute—Indicates that a dispute condition is detected.

This example shows the spanning-tree configuration summary:

```
Router# show spanning-tree summary

Switch is in rapid-pvst mode
Root bridge for: Bridge VLAN0025
EtherChannel misconfiguration guard is enabled
Extended system ID                  is enabled
PortFast Edge BPDU Guard Default    is disabled
Portfast Edge BPDU Filter Default   is disabled
Portfast Default                    is edge
Bridge Assurance                    is enabled
Loopguard                           is disabled
UplinkFast                          is disabled
BackboneFast                        is disabled
Pathcost method used is short
PVST Simulation Default             is enabled

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0025                  0        0         0        1          1
VLAN0030                  0        0         0        2          2
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0        0         0        3          3
Router#
```

Possible states for the Bridge Assurance field are as follows:

- is enabled
- is disabled
- is enabled but not active in the PVST mode

This example shows the spanning tree summary when PVST simulation is disabled in any STP mode:

```
Router# show spanning-tree summary
```

```
Switch is in mst mode (IEEE Standard)
Root bridge for: MST0
EtherChannel misconfig guard is enabled
Extended system ID           is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Pathcost method used         is long
PVST Simulation Default      is disabled

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
MST0                         2         0        0          0          2
---------------------- -------- --------- -------- ---------- ----------
1 mst                        2         0        0          0          2
```

Possible states for the PVST Simulation Default field are as follows:

- is enabled
- is disabled
- is enabled but not active in rapid-PVST mode

This example shows the spanning tree summary totals:

```
Router# show spanning-tree summary totals
Root bridge for: Bridge VLAN0025
EtherChannel misconfiguration guard is enabled
Extended system ID                  is enabled
PortFast Edge BPDU Guard Default    is enabled
Portfast Edge BPDU Filter Default   is disabled
Portfast Default                    is edge
Bridge Assurance                    is enabled
Loopguard                           is disabled
UplinkFast                          is disabled
BackboneFast                        is disabled
Pathcost method used is long

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0        0         0        3          3
Router#
```

This example shows the spanning-tree configuration details of an edge port:

```
Router# show spanning-tree interface gi3/13 detail
Port 269 (GigabitEthernet3/13) of VLAN0002 is forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.269.
   Designated root has priority 32770, address 0002.172c.f400
   Designated bridge has priority 32770, address 0002.172c.f400
   Designated port id is 128.269, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   Loop guard is enabled by default on the port
   The port is in the portfast edge mode by default
   BPDU: sent 2183, received 0
```

This example shows the spanning-tree configuration details of a trunk port:

```
Router(config-if)# spanning-tree portfast edge trunk
%Warning:portfast should only be enabled on ports connected to a single
```

```
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

Router(config-if)# ^Z

Router# show spanning-tree interface gi3/13 detail
Port 269 (GigabitEthernet3/13) of VLAN0002 is forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.269.
   Designated root has priority 32770, address 0002.172c.f400
   Designated bridge has priority 32770, address 0002.172c.f400
   Designated port id is 128.269, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   Loop guard is enabled by default on the port
   The port is in the portfast edge trunk mode
   BPDU: sent 2183, received 0
Router#
```

This example shows the spanning-tree configuration details of an edge port when a dispute condition has been detected:

```
Router# show spanning-tree interface gi3/13 detail
Port 269 (GigabitEthernet3/13) of VLAN0002 is designated blocking (dispute)
   Port path cost 4, Port priority 128, Port Identifier 128.297.
   Designated root has priority 32769, address 0013.5f20.01c0
   Designated bridge has priority 32769, address 0013.5f20.01c0
   Designated port id is 128.297, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   BPDU: sent 132, received 1
Router#
```

This example shows the spanning tree portfast edge interface operational state for all the instances:

```
Router# show spanning-tree interface gi3/1 portfast edge
MST0             disabled
MST1             disabled
```

Examples in Releases Earlier Than Release 12.2(33)SXI

Note The output of the **show spanning-tree** commands in releases earlier than Release 12.2(33)SXI differs from the output in Release 12.2(33)SXI and later releases. The differences are shown in bold in the examples in this section.

This example displays the spanning-tree status:

```
Router# show spanning-tree
VLAN0010
  Spanning tree enabled protocol rstp
  Root ID    Priority    32778
             Address     0002.172c.f400
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-ext 10)
```

```
             Address     0002.172c.f400
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi3/14           Desg FWD 4         128.270  Edge, P2p
Router#
```

This example shows the spanning-tree configuration summary:

```
Router# show spanning-tree summary

Switch is in rapid-pvst mode
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID   is disabled
Portfast             is enabled by default
PortFast BPDU Guard  is disabled by default
Portfast BPDU Filter is disabled by default
Loopguard            is disabled by default
UplinkFast           is disabled
BackboneFast         is disabled
Pathcost method used is long

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0001                  0        0         0        1          1
VLAN0010                  0        0         0        2          2
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0        0         0        3          3
Router#
```

This example shows the spanning-tree configuration summary totals:

```
Router# show spanning-tree summary totals

Switch is in rapid-pvst mode
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID   is disabled
Portfast             is enabled by default
PortFast BPDU Guard  is disabled by default
Portfast BPDU Filter is disabled by default
Loopguard            is disabled by default
UplinkFast           is disabled
BackboneFast         is disabled
Pathcost method used is long

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0        0         0        3          3
Router#
```

This example shows the spanning-tree configuration details of an edge port:

```
Router# show spanning-tree interface fastEthernet 4/4 detail
 Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by default
```

```
   Link type is point-to-point by default
   BPDU:sent 10, received 0
```

This example shows the spanning-tree configuration details of a trunk port:

```
Router(config-if)# spanning-tree portfast trunk
%Warning:portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

Router(config-if)# ^Z

Router# show spanning-tree interface fastEthernet 4/4 detail
 Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   BPDU:sent 30, received 0
Router#
```

This example shows the spanning tree portfast interface operational state for all the instances:

```
Router# show spanning-tree interface gi3/1 portfast
MST0             disabled
MST1             disabled
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 3

IP Routing Protocols

Configuring Layer 3 Interfaces

This chapter contains information about how to configure Layer 3 interfaces in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Layer 3 Interface Configuration Guidelines and Restrictions

When configuring Layer 3 interfaces, follow these guidelines and restrictions:

- We recommend that you configure no more than 2,000 Layer 3 VLAN interfaces.
- The **ip unnumbered** command is supported on Layer 3 VLAN interfaces.
- To support VLAN interfaces, create and configure VLANs and assign VLAN membership to Layer 2 LAN ports. For more information, see Chapter 23, "Configuring VLANs" and Chapter 22, "Configuring VTP."
- Cisco IOS Release 12.2SX does not support:
  - Integrated routing and bridging (IRB)
  - Concurrent routing and bridging (CRB)
  - Remote source-route bridging (RSRB)
- Use bridge groups on VLAN interfaces, sometimes called fall-back bridging, to bridge nonrouted protocols. Bridge groups on VLAN interfaces are supported in software on the route processor (RP).
- Cisco IOS Release 12.2SX does not support the IEEE bridging protocol for bridge groups. Configure bridge groups to use the VLAN-bridge or the DEC spanning-tree protocol.

Configuring Subinterfaces on Layer 3 Interfaces

When configuring Layer 3 subinterfaces, follow these guidelines and restrictions:

- The PFC3 supports these features on LAN port subinterfaces:
  - IPv4 unicast forwarding, including MPLS VPN
  - IPv4 multicast forwarding, including MPLS VPN
  - 6PE
  - EoMPLS
  - IPv4 unnumbered
  - Counters for subinterfaces in MIBS and with the **show vlans** command
  - iBGP and eBGP
  - OSPF
  - EIGRP
  - RIPv1/v2
  - RIPv2
  - ISIS
  - Static routing
  - Unidirectional link routing (UDLR)
  - IGMPv1, IGMPv2, IGMPv3
  - PIMv1, PIMv2
  - SSM IGMPv3lite and URD
  - Stub IP multicast routing

  - IGMP join
  - IGMP static group
  - Multicast routing monitor (MRM)
  - Multicast source discovery protocol (MSDP)
  - SSM
  - IPv4 Ping
  - IPv6 Ping
- Always use the **native** keyword when the VLAN ID is the ID of the IEEE 802.1Q native VLAN. Do not configure encapsulation on the native VLAN of an IEEE 802.1Q trunk without the **native** keyword.
- Because VLAN IDs are global to the switch, you can use a VLAN internally, on a subinterface, or with a Layer 3 VLAN interface.
  - You cannot configure an internal VLAN on a subinterface or a Layer 3 VLAN interface.
  - You cannot configure a subinterface VLAN on a Layer 3 VLAN interface.
  - You cannot configure a VLAN used with a Layer 3 VLAN interface on a subinterface.

Note You cannot configure a VLAN used on one interface or subinterface on another interface or subinterface.

- With any VTP version, you can configure subinterfaces with any normal range or extended range VLAN ID in VTP transparent mode. Because VLAN IDs 1 to 1005 are global in the VTP domain and can be defined on other network devices in the VTP domain, you can use only extended range VLANs with subinterfaces in VTP client or server mode. In VTP client or server mode, normal range VLANs are excluded from subinterfaces.

Note If you configure normal range VLANs on subinterfaces, you cannot change the VTP mode from transparent.

To configure a subinterface, perform this task:

	Command	Purpose
Step 1	`Router>` **`enable`**	Enters privileged EXEC mode.
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`interface`** `{{`*`type`*[1] *`slot/port.subinterface`*`}	{`**`port-channel`** *`port_channel_number.subinterface`*`}}`	Selects an interface and enters subinterface configuration mode.
Step 4	`Router(config-subif)#` **`encapsulation dot1q`** *`vlan_ID`* `[`**`native`**`]`	Configures 802.1Q encapsulation for the subinterface.
Step 5	`Router(config-if)#` **`exit`**	Returns to global configuration mode.

1. *type* = **fastethernet**, **gigabitethernet**, **tengigabitethernet**, or **ge-wan**

Configuring IPv4 Routing and Addresses

For complete information and procedures, see these publications:

- *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.2, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/fipr_c.html

- *Cisco IOS IP and IP Routing Command Reference*, Release 12.2, at these URLs:

  http://www.cisco.com/en/US/docs/ios/12_2/ipaddr/command/reference/fipras_r.html

  http://www.cisco.com/en/US/docs/ios/12_2/iproute/command/reference/fiprrp_r.html

When configuring IPv4 routing and addresses, follow these guidelines and restrictions:

- See the command reference for information about the **maximum-paths** command.
- The Policy Feature Card (PFC) and any Distributed Feature Cards (DFCs) provide hardware support for policy-based routing (PBR) for route-map sequences that use the **match ip address**, **set ip next-hop**, and **ip default next-hop** PBR keywords.

  When configuring PBR, follow these guidelines and restrictions:

  – The PFC provides hardware support for PBR configured on a tunnel interface.

  – The PFC does not provide hardware support for PBR configured with the **set ip next-hop** keywords if the next hop is a tunnel interface.

  – To avoid high CPU utilization, do not configure an address in the same subnet as the next hop.

  – If the RP address falls within the range of a PBR ACL, traffic addressed to the RP is policy routed in hardware instead of being forwarded to the RP. To prevent policy routing of traffic addressed to the RP, configure PBR ACLs to deny traffic addressed to the RP.

  – Any options in Cisco IOS ACLs that provide filtering in a PBR route-map that would cause flows to be sent to the RP to be switched in software are ignored. For example, logging is not supported in ACEs in Cisco IOS ACLs that provide filtering in PBR route-maps.

  – PBR traffic through switching module ports where PBR is configured is routed in software if the switching module resets. (CSCee92191)

  – Any **permit** route-map sequence with no **set** statement will cause matching traffic to be processed by the RP.

  – In Cisco IOS Release 12.2(33)SXH4 and later releases, for efficient use of hardware resources, enter the **platform ipv4 pbr optimize tcam** command in global configuration mode when configuring multiple PBR sequences (or a single PBR sequence with multiple ACLs) in which more than one PBR ACL contains DENY entries. In earlier releases, we recommend avoiding this type of configuration. (CSCsr45495)

  – In Cisco IOS Release 12.2(33)SXH4 and later releases, the BOOTP/DHCP traffic will be dropped unless explicitly permitted. In Cisco IOS Release 12.2(18)SXF, BOOTP/DHCP packets are not subjected to a PBR configured in the ingress interfaces and the BOOTP/DHCP packets are forwarded to the BOOTP/DHCP server, although they are not explicitly permitted.

  To configure PBR, see the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2, “Classification,” “Configuring Policy-Based Routing,” at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfpbr_ps1835_TSD_Products_Configuration_Guide_Chapter.html

To configure IPv4 routing and an IPv4 address on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip routing`**	Enables IPv4 routing. (Required only if IPv4 routing is disabled.)
Step 2	`Router(config)#` **`router`** *`ip_routing_protocol`*	Specifies an IPv4 routing protocol.
Step 3	`Router(config-router)#` *`ip_routing_protocol_commands`*	Configures the IPv4 routing protocol.
Step 4	`Router(config-router)#` **`exit`**	Exists IPv4 routing protocol configuration mode.
Step 5	`Router(config)#` **`interface`** {**`vlan`** *`vlan_ID`*} | {*`type`*[1] *`slot/port`*} | {**`port-channel`** *`port_channel_number`*}	Selects an interface to configure.
Step 6	`Router(config-if)#` **`ip address`** *`ip_address subnet_mask`*	Configures the IPv4 address and IPv4 subnet.
Step 7	`Router(config-if)#` **`no shutdown`**	Enables the interface.
Step 8	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 9	`Router#` **`show interfaces`** [{**`vlan`** *`vlan_ID`*} | {*`type`*[1] *`slot/port`*} | {**`port-channel`** *`port_channel_number`*}] `Router#` **`show ip interfaces`** [{**`vlan`** *`vlan_ID`*} | {*`type`*[1] *`slot/port`*} | {**`port-channel`** *`port_channel_number`*}] `Router#` **`show running-config interfaces`** [{**`vlan`** *`vlan_ID`*} | {*`type`*[1] *`slot/port`*} | {**`port-channel`** *`port_channel_number`*}]	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable IPv4 Routing Information Protocol (RIP) routing:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip routing
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# end
Router#
```

This example shows how to configure an IPv4 address on Fast Ethernet port 5/4:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/4
Router(config-if)# ip address 172.20.52.106 255.255.255.248
Router(config-if)# no shutdown
Router(config-if)# end
Router#
```

This example uses the **show interfaces** command to display the interface IPv4 address configuration and status of Fast Ethernet port 5/4:

```
Router# show interfaces fastethernet 5/4
FastEthernet5/4 is up, line protocol is up
  Hardware is Cat6K 100Mb Ethernet, address is 0050.f0ac.3058 (bia 0050.f0ac.3058)
  Internet address is 172.20.52.106/29
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
```

```
  Full-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     7 packets input, 871 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     8 packets output, 1658 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Router#
```

This example uses the **show ip interface** command to display the detailed configuration and status of Fast Ethernet port 5/4:

```
Router# show ip interface fastethernet 5/4
FastEthernet5/4 is up, line protocol is up
  Internet address is 172.20.52.106/29
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP Fast switching turbo vector
  IP Normal CEF switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
  IP multicast multilayer switching is disabled
  IP mls switching is enabled
Router#
```

This example uses the **show running-config** command to display the interface IPv4 address configuration of Fast Ethernet port 5/4:

```
Router# show running-config interfaces fastethernet 5/4
Building configuration...

Current configuration:
!
interface FastEthernet5/4
 description "Router port"
 ip address 172.20.52.106 255.255.255.248
 no ip directed-broadcast
!
```

Configuring IPX Routing and Network Numbers

Note The RP supports IPX with fast switching.

For complete information and procedures, see these publications:

- *Cisco IOS AppleTalk and Novell IPX Configuration Guide*, Release 12.2, at this URL:
  http://www.cisco.com/en/US/docs/ios/12_2/atipx/configuration/guide/fatipx_c.html
- *Cisco IOS AppleTalk and Novell IPX Command Reference*, Release 12.2, at this URL:
  http://www.cisco.com/en/US/docs/ios/12_2/atipx/command/reference/fatipx_r.html

To configure routing for Internetwork Packet Exchange (IPX) and configure IPX on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# ipx routing`	Enables IPX routing.
Step 2	`Router(config)# router ipx_routing_protocol`	Specifies an IP routing protocol. This step might include other commands, such as specifying the networks to route with the **network** command.
Step 3	`Router(config)# interface {vlan vlan_ID}	{type[1] slot/port}	{port-channel port_channel_number}`	Selects an interface to configure.
Step 4	`Router(config-if)# ipx network [network	unnumbered] encapsulation encapsulation_type`	Configures the IPX network number. This enables IPX routing on the interface. When you enable IPX routing on the interface, you can also specify an encapsulation type.
Step 5	`Router(config-if)# no shutdown`	Enables the interface.
Step 6	`Router(config-if)# end`	Exits configuration mode.
Step 7	`Router# show interfaces [{vlan vlan_ID}	{type[1] slot/port}	{port-channel port_channel_number}]` `Router# show ipx interfaces [{vlan vlan_ID}	{type[1] slot/port}	{port-channel port_channel_number}]` `Router# show running-config interfaces [{vlan vlan_ID}	{type[1] slot/port}	{port-channel port_channel_number}]`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable IPX routing and assign an IPX network address to interface VLAN 100:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ipx routing
Router(config)# ipx router rip
Router(config-ipx-router)# network all
Router(config-ipx-router)# interface vlan 100
Router(config-if)# ipx network 100 encapsulation snap
Router(config-if)# no shutdown
Router(config-if)# end
Router# copy running-config startup-config
```

Configuring AppleTalk Routing, Cable Ranges, and Zones

For complete information and procedures, see these publications:

- *Cisco IOS AppleTalk and Novell IPX Configuration Guide*, Release 12.2, at this URL:
  http://www.cisco.com/en/US/docs/ios/12_2/atipx/configuration/guide/fatipx_c.html
- *Cisco IOS AppleTalk and Novell IPX Command Reference*, Release 12.2, at this URL:
  http://www.cisco.com/en/US/docs/ios/12_2/atipx/command/reference/fatipx_r.html

To configure routing for AppleTalk, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **appletalk routing**	Enables AppleTalk routing.
Step 2	Router(config)# **interface** {**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}	Selects an interface to configure.
Step 3	Router(config-if)# **appletalk cable-range** *cable_range*	Assigns a cable range to the interface.
Step 4	Router(config-if)# **appletalk zone** *zone_name*	Assigns a zone name to the interface.
Step 5	Router(config-if)# **no shutdown**	Enables the interface.
Step 6	Router(config-if)# **end**	Exits configuration mode.
Step 7	Router# **show interfaces** [{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}] Router# **show appletalk interfaces** [{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}] Router# **show running-config interfaces** [{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}]	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable AppleTalk routing and assign an AppleTalk cable-range and zone name to interface VLAN 100:

```
Router# configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# appletalk routing
Router(config)# interface vlan 100
Router(config-if)# appletalk cable-range 100-100
Router(config-if)# appletalk zone Engineering
Router(config-if)# no shutdown
Router(config-if)# end
Router# copy running-config startup-config
```

Configuring Other Protocols on Layer 3 Interfaces

See these publications for information about configuring other protocols on Layer 3 interfaces:

- *Cisco IOS Apollo Domain, VINES, DECnet, ISO CLNS, and XNS Configuration Guide*, Release 12.2, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/apollo/configuration/guide/fapolo_c.html

- *Cisco IOS Apollo Domain, VINES, DECnet, ISO CLNS, and XNS Command Reference*, Release 12.2, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/apollo/command/reference/fapolo_r.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring UDE and UDLR

This chapter describes how to configure unidirectional Ethernet (UDE) and unidirectional link routing (UDLR).

Note
- Cisco ME 6500 Series Ethernet switches do not support UDE and UDLR.
- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

These sections describe UDE and UDLR:

- Understanding UDE and UDLR, page 31-1
- Configuring UDE and UDLR, page 31-3

Understanding UDE and UDLR

These sections describe UDE and UDLR:

- UDE and UDLR Overview, page 31-2
- Supported Hardware, page 31-2
- Understanding UDE, page 31-2
- Understanding UDLR, page 31-3

UDE and UDLR Overview

Routing protocols support unidirectional links only if the unidirectional links emulate bidirectional links because routing protocols expect to send and receive traffic through the same interface.

Unidirectional links are advantageous because when you transmit mostly unacknowledged unidirectional high-volume traffic (for example, a video broadcast stream) over a high-capacity full-duplex bidirectional link, you use both the link from the source to the receiver and the equally high-capacity reverse-direction link, called the “back channel,” that carries the few acknowledgements from the receiver back to the source.

UDE and UDLR support use of a high-capacity unidirectional link for the high-volume traffic without consuming a similar high-capacity link for the back channel. UDE provides a high-capacity unidirectional link. UDLR provides the back channel through a tunnel that is configured over a regular-capacity link, and also provides bidirectional link emulation by transparently making the back channel appear to be on the same interface as the high-capacity unidirectional link.

Supported Hardware

On Catalyst 6500 series switches, UDE and UDLR are supported on the interfaces of these switching modules:

- WS-X6704-10GE 4-port 10-Gigabit Ethernet
- WS-X6816-GBIC 16-port Gigabit Ethernet
- WS-X6516A-GBIC 16-port Gigabit Ethernet
- WS-X6516-GBIC 16-port Gigabit Ethernet

Understanding UDE

These sections describe UDE:

- UDE Overview, page 31-2
- Understanding Hardware-Based UDE, page 31-3
- Understanding Software-Based UDE, page 31-3

UDE Overview

On Catalyst 6500 series switches, you can implement UDE with hardware or in software. Hardware-based UDE and software-based UDE both use only one strand of fiber instead of the two strands of fiber required by bidirectional traffic.

The unidirectional transceiver determines whether hardware-based UDE is receive-only or transmit-only. You can configure software-based UDE as either transmit-only or receive-only.

You do not need to configure software-based UDE on ports where you implement hardware-based UDE.

Note See the “Supported Hardware” section on page 31-2 for a list of the module with interfaces that support hardware-based UDE and software-based UDE.

Understanding Hardware-Based UDE

You can create a unidirectional link by using a unidirectional transceiver. Unidirectional transceivers are less expensive than bidirectional transceivers. These are the supported unidirectional transceivers:

- Receive-only WDM GBIC (WDM-GBIC-REC=)
- Receive-only XENPAK (WDM-XENPAK-REC=)

Understanding Software-Based UDE

You can create a unidirectional link by configuring ports equipped with bidirectional transceivers to unidirectionally transmit or receive traffic. You can use software-based UDE when there is no appropriate unidirectional transceiver available. For example, with no support for any transmit-only transceivers, you must configure transmit-only links with software-based UDE.

Understanding UDLR

UDLR provides a unidirectional tunnel as the back channel of a unidirectional high-capacity link, and transparently emulates a single bidirectional link for unicast and multicast traffic.

UDLR intercepts packets that need to be sent on receive-only interfaces and sends them on UDLR back-channel tunnels. When routers receive these packets over UDLR back-channel tunnels, UDLR makes the packets appear as if received on send-only interfaces.

UDLR back-channel tunnels support these IPv4 features:

- Address Resolution Protocol (ARP)
- Next Hop Resolution Protocol (NHRP)
- Emulation of a bidirectional link for all IPv4 traffic (as opposed to only broadcast and multicast control traffic)
- IPv4 GRE multipoint at a receive-only tunnels

Note UDLR back-channel tunnels do not support IPv6 or MPLS.

Configuring UDE and UDLR

These sections describe how to configure UDE and UDLR:

- Configuring UDE, page 31-4
- Configuring UDLR, page 31-6

Note This caveat is open in releases that support UDLR: Neighboring ISIS routers are not seen through a UDLR topology. (CSCee56596)

Configuring UDE

These sections describe how to configure UDE:

- UDE Configuration Guidelines, page 31-4
- Configuring Hardware-Based UDE, page 31-5
- Configuring Software-Based UDE, page 31-5

UDE Configuration Guidelines

When configuring UDE, follow these guidelines:

- UDE is supported on the Supervisor Engine 720.
- STP cannot prevent Layer 2 loops in topologies that include unidirectional links.
- Send-only ports always transition to the STP forwarding state, because send-only ports never receive BPDUs.
- Receive-only ports cannot send BPDUs.
- Unidirectional ports do not support any features or protocols that require negotiation with the port at the other end of the link, including these:
  - Speed and duplex mode autonegotiation
  - Link negotiation
  - IEEE 802.3Z flow control
  - Dynamic trunking protocol (DTP)

  You must manually configure the parameters that are typically controlled by Layer 2 protocols.
- A topology that includes unidirectional links only supports the VLAN Trunking Protocol (VTP) when the VTP server can send VTP frames to all switches in the VTP domain.
- Disable VTP pruning on switches that have send-only ports, because VTP pruning depends on a bidirectional exchange of information.
- Unidirectional EtherChannels cannot support PAgP or LACP. To create a unidirectional EtherChannel, you must configure the EtherChannel “on” mode.
- You can configure software-based UDE on the physical ports in an EtherChannel. You cannot configure software-based UDE on any nonphysical interfaces (for example, port-channel interfaces).
- When you implement hardware-based UDE on a port or configure software-based UDE on a port, UDLD is automatically disabled on the port.
- CDP sends CDP frames from send-only ports and receives CDP frames from receive-only ports, which means that the switch on the send-only side of a unidirectional link never receives CDP information.
- SPAN does not restrict configuration of unidirectional ports as sources or destinations.
  - Send-only ports can be SPAN destinations.
  - Receive-only ports can be SPAN sources.
- Unidirectional ports do not support IEEE 802.1X port-based authentication.
- IGMP snooping does not support topologies where there are unidirectional links between the switch and the hosts that are receiving multicast traffic.

- Configure UDLR with UDE to support communication over unidirectional links between IGMP snooping on the switch and a multicast router.
- Unidirectional links do not support ARP.

Configuring Hardware-Based UDE

There are no software configuration procedures required to support hardware-based UDE. Install a unidirectional transceiver to implement hardware-based UDE.

To verify hardware-based UDE on a port, perform this task:

Command	Purpose
Router# **show interfaces** [{**gigabitethernet** | **tengigabitethernet**} *slot/interface*}] **status**	Verifies the configuration.

This example shows how to verify the configuration of Gigabit Ethernet port 1/1:

```
Router# show interfaces gigabitethernet 1/1 status

Port    Name               Status       Vlan       Duplex  Speed Type
Gi1/1                      notconnect   1            full   1000 WDM-RXONLY
```

Configuring Software-Based UDE

To configure software-based UDE on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {{**gigabitethernet** | **tengigabitethernet**} *slot/interface*}	Selects the interface to configure.
Step 2	Router(config-if)# **unidirectional** {**send-only** | **receive-only**}	Configures software-based UDE.
	Router(config-if)# **no unidirectional**	Removes the software-based UDE configuration.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show interface** {{**gigabitethernet** | **tengigabitethernet**} *slot/interface*} **unidirectional**	Verifies the configuration.

This example shows how to configure 10-Gigabit Ethernet port 1/1 as a UDE send-only port:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 1/1
Router(config-if)# unidirectional send-only
Router(config-if)# end

Warning!

Enable port unidirectional mode will automatically disable port udld. You must manually
ensure that the unidirectional link does not create a spanning tree loop in the network.

Enable l3 port unidirectional mode will automatically disable ip routing on the port. You
must manually configure static ip route and arp entry in order to route ip traffic.
```

This example shows how to configure 10-Gigabit Ethernet port 1/2 as a UDE receive-only port:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 1/2
Router(config-if)# unidirectional receive-only
Router(config-if)# end

Warning!

Enable port unidirectional mode will automatically disable port udld. You must manually
ensure that the unidirectional link does not create a spanning tree loop in the network.

Enable l3 port unidirectional mode will automatically disable ip routing on the port. You
must manually configure static ip route and arp entry in order to route ip traffic.
```

This example shows how to verify the configuration:

```
Router> show interface tengigabitethernet 1/1 unidirectional
  Unidirectional configuration mode: send only
  CDP neighbour unidirectional configuration mode: receive only
```

This example shows how to disable UDE on 10-Gigabit Ethernet interface 1/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 1/1
Router(config-if)# no unidirectional
Router(config-if)# end
```

This example shows the result of entering the **show interface** command for a port that does not support unidirectional Ethernet:

```
Router# show interface fastethernet 6/1 unidirectional
Unidirectional Ethernet is not supported on FastEthernet6/1
```

Configuring UDLR

These sections describe how to configure UDLR:

- UDLR Back-Channel Tunnel Configuration Guidelines, page 31-6
- Configuring a Receive-Only Tunnel Interface for a UDE Send-Only Port, page 31-7
- Configuring a Send-Only Tunnel Interface for a UDE Receive-Only Port, page 31-7

UDLR Back-Channel Tunnel Configuration Guidelines

When configuring UDLR back-channel tunnels, follow these guidelines:

- The PFC3 does not provide hardware support for UDLR back-channel tunnels. The route processor (RP) supports UDLR back-channel tunnels in software.
- Configure a UDLR back-channel tunnel for each unidirectional link.
- On UDE send-only interfaces, configure the UDLR back-channel tunnel interface to receive.
- On UDE receive-only interfaces, configure the UDLR back-channel tunnel interface to send.
- You must configure IPv4 addresses on UDLR back-channel tunnel interfaces.
- You must configure source and destination IPv4 addresses on UDLR back-channel tunnel interfaces.
- The UDLR back-channel tunnel default mode is GRE.

- UDLR back-channel tunnels do not support IPv6 or MPLS.

Configuring a Receive-Only Tunnel Interface for a UDE Send-Only Port

To configure a receive-only tunnel interface for a UDE send-only port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface tunnel`** *`number`*	Selects the tunnel interface.
Step 2	`Router(config-if)#` **`tunnel udlr receive-only`** *`ude_send_only_port`*	Associates the tunnel receive-only interface with the UDE send-only port.
Step 3	`Router(config-if)#` **`ip address`** *`ipv4_address`*	Configures the tunnel IPv4 address.
Step 4	`Router(config-if)#` **`tunnel source`** `{`*`ipv4_address`* `|` *`type number`*`}`	Configures the tunnel source.
Step 5	`Router(config-if)#` **`tunnel destination`** `{`*`hostname`* `|` *`ipv4_address`*`}`	Configures the tunnel destination.

Configuring a Send-Only Tunnel Interface for a UDE Receive-Only Port

To configure a send-only tunnel interface for a UDE receive-only port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface tunnel`** *`number`*	Selects the tunnel interface.
Step 2	`Router(config-if)#` **`tunnel udlr send-only`** *`ude_receive_only_port`*	Associates the tunnel send-only interface with the UDE receive-only port.
Step 3	`Router(config-if)#` **`ip address`** *`ipv4_address`*	Configures the tunnel IPv4 address.
Step 4	`Router(config-if)#` **`tunnel source`** `{`*`ipv4_address`* `|` *`type number`*`}`	Configures the tunnel source.
Step 5	`Router(config-if)#` **`tunnel destination`** `{`*`hostname`* `|` *`ipv4_address`*`}`	Configures the tunnel destination.
Step 6	`Router(config-if)#` **`tunnel udlr address-resolution`**	Enables ARP and NHRP.

In the following UDE and UDLR sample configuration:

- On Router A:
  - Open Shortest Path First (OSPF) and PIM are configured.
  - 10-Gigabit Ethernet port 1/1 is a send-only UDE port.
  - The UDLR back-channel tunnel is configured as receive only and is associated with 10-Gigabit Ethernet port 1/1.
- On Router B:
  - OSPF and PIM are configured.
  - 10-Gigabit Ethernet port 1/2 is a receive-only UDE port.
  - The UDLR back-channel tunnel is configured as send-only and is associated with 10-Gigabit Ethernet port 1/2.
  - ARP and NHRP are enabled.

Router A Configuration

```
ip multicast-routing
!
! tengigabitethernet 1/1 is send-only
!
interface tengigabitethernet 1/1
 unidirectional send-only
 ip address 10.1.0.1 255.255.0.0
 ip pim sparse-dense-mode
!
! Configure tunnel as receive-only UDLR tunnel.
!
interface tunnel 0
 tunnel source 11.0.0.1
 tunnel destination 11.0.0.2
 tunnel udlr receive-only tengigabitethernet 1/1
!
! Configure OSPF.
!
router ospf <pid>
 network 10.0.0.0 0.255.255.255 area 0
```

Router B Configuration

```
ip multicast-routing
!
! tengigabitethernet 1/2 is receive-only
!
interface tengigabitethernet 1/2
 unidirectional receive-only
 ip address 10.1.0.2 255.255.0.0
 ip pim sparse-dense-mode
!
! Configure tunnel as send-only UDLR tunnel.
!
interface tunnel 0
 tunnel source 11.0.0.2
 tunnel destination 11.0.0.1
 tunnel udlr send-only tengigabitethernet 1/2
 tunnel udlr address-resolution
!
! Configure OSPF.
!
router ospf <pid>
 network 10.0.0.0 0.255.255.255 area 0
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Multiprotocol Label Switching

Configuring Multiprotocol Label Switching

This chapter describes how to configure Multiprotocol Label Switching (MPLS) in Cisco IOS Release 12.2SX.

Note

- MPLS is not supported in PFC3A mode.
- For complete syntax and usage information for the commands used in this chapter, see these publications:
  - The Cisco IOS Master Command List, at this URL:

    http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html
  - The Release 12.2 publications at this URL:

    http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

MPLS

These sections describe MPLS:

Understanding MPLS

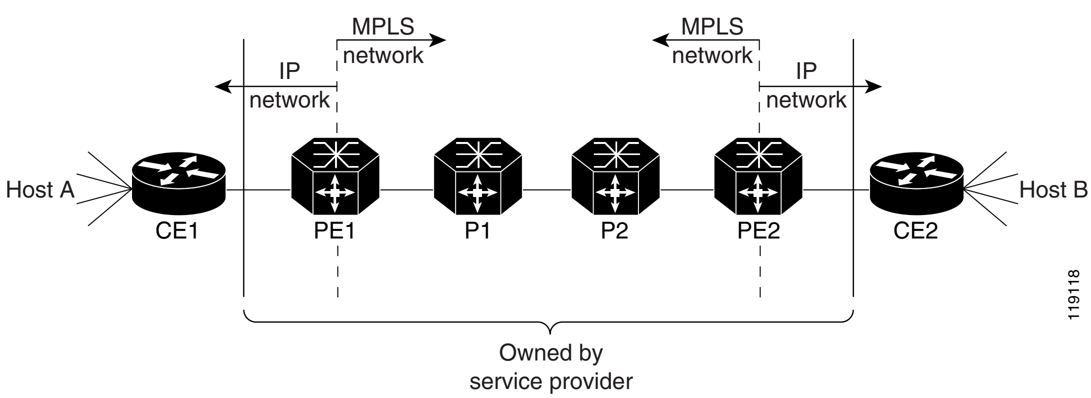
MPLS uses label switching to forward packets over various link-level technologies such as Packet-over-SONET (POS), Frame Relay, ATM, and Ethernet. Labels are assigned to packets based on groupings or forwarding equivalence classes (FECs). The label is added between the Layer 2 and the Layer 3 header.

In an MPLS network, the label edge router (LER) performs a label lookup of the incoming label, swaps the incoming label with an outgoing label, and sends the packet to the next hop at the label switch router (LSR). Labels are imposed (pushed) on packets only at the ingress edge of the MPLS network and are removed (popped) at the egress edge. The core network LSRs (provider, or P routers) read the labels, apply the appropriate services, and forward the packets based on the labels.

Incoming labels are aggregate or nonaggregate. The aggregate label indicates that the arriving MPLS packet must be switched through an IP lookup to find the next hop and the outgoing interface. The nonaggregate label indicates that the packet contains the IP next hop information.

Figure 32-1 shows an MPLS network of a service provider that connects two sites of a customer network.

Figure 32-1 MPLS Network



For additional information on MPLS, see this publication:

http://www.cisco.com/en/US/docs/ios-xml/ios/mpls/config_library/12-2sx/mp-12-2sx-library.html

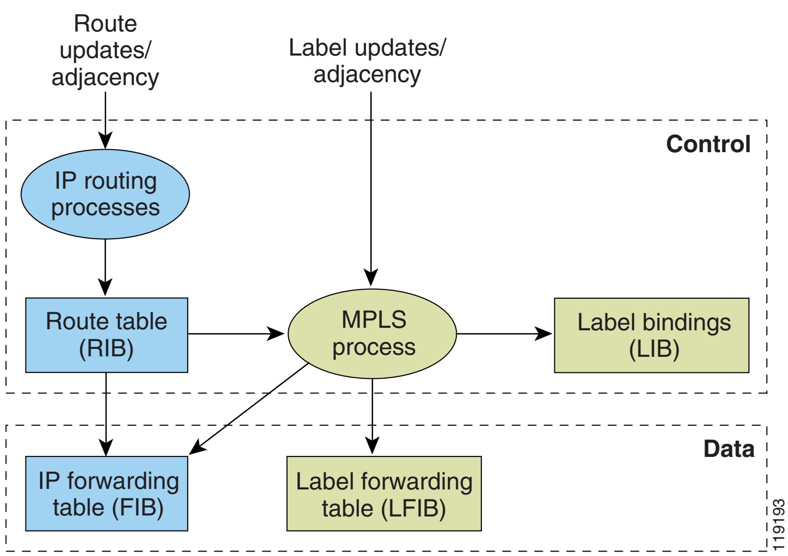
Cisco IOS Release 12.2SX supports Layer 3 Multiprotocol Label Switching (MPLS) virtual private networks (VPNs), and Layer 2 Ethernet over MPLS (EoMPLS), with quality of service (QoS) and security.

The route processor (RP) performs Layer 3 control-plane functions, including address resolution and routing protocols. The RP processes information from the Routing and Label Distribution Protocols and builds the IP forwarding (FIB) table and the label forwarding (LFIB) table. The RP distributes the information in both tables to the PFC3.

The PFC3 receives the information and creates its own copies of the FIB and LFIB tables. Together, these tables comprise the FIB TCAM. The DFC looks up incoming IP packets and labeled packets against the FIB TCAM table. The lookup result is the pointer to a particular adjacency entry. It is the adjacency entry that contains appropriate information for label pushing (for IP to MPLS path), label swapping (for MPLS to MPLS path), label popping (for MPLS to IP path), and encapsulation.

Figure 32-2 shows the various functional blocks that support MPLS. Routing protocol generates a routing information base (RIB) that is used for forwarding IP and MPLS data packets. For Cisco Express Forwarding (CEF), necessary routing information from the RIB is extracted and built into a forwarding information base (FIB). The label distribution protocol (LDP) obtains routes from the RIB and distributes the label across a label switch path to build a label forwarding information base (LFIB) in each of the LSRs and LERs.

Figure 32-2 MPLS Forwarding, Control and Data Planes



IP to MPLS

At the ingress to the MPLS network, the PFC examines the IP packets and performs a route lookup in the FIB TCAM. The lookup result is the pointer to a particular adjacency entry. The adjacency entry contains the appropriate information for label pushing (for IP to MPLS path) and encapsulation. The PFC generates a result containing the imposition label(s) needed to switch the MPLS packet.

Note If MPLS load sharing is configured, the adjacency may point to a load-balanced path. See "Basic MPLS Load Balancing" section on page 32-8.

MPLS to MPLS

At the core of an MPLS network, the PFC uses the topmost label to perform a lookup in the FIB TCAM. The successful lookup points to an adjacency that swaps the top label in the packet with a new label as advertised by the downstream label switch router (LSR). If the router is the penultimate hop LSR router (the upstream LSR next to the egress LER), the adjacency instructs the PFCBXL to pop the topmost label, resulting in either an MPLS packet with the remaining label for any VPN or AToM use or a native IP packet.

MPLS to IP

At the egress of the MPLS network there are several possibilities.

For a native IP packet (when the penultimate router has popped the label), the PFC performs a route lookup in the FIB TCAM.

For a MPLS VPN packet, after the Interior Gateway Protocol (IGP) label is popped at penultimate router, the VPN label remains. The operation that the PFC performs depends on the VPN label type. Packets carrying aggregate labels require a second lookup based on the IP header after popping the aggregate label. For a nonaggregate label, the PFC performs a route lookup in the FIB TCAM to obtain the IP next hop information.

For the case of a packet with an IGP label and a VPN label, when there is no penultimate hop popping (PHP), the packet carries the explicit-null label on top of the VPN label. The PFC looks up the top label in the FIB TCAM and recirculates the packet. Then the PFC handles the remaining label as described in the preceding paragraph, depending on whether it is an aggregate or nonaggregate label.

Packets with the explicit-null label for the cases of EoMPLS, MPLS, and MPLS VPN an MPLS are handled the same way.

MPLS VPN Forwarding

There are two types of VPN labels: aggregate labels for directly connected network or aggregate routes, and nonaggregate labels. Packets carrying aggregate labels require a second lookup based on the IP header after popping the aggregate label. The VPN information (VPN-IPv4 address, extended community, and label) is distributed through the Multiprotocol-Border Gateway Protocol (MP-BGP).

Recirculation

In certain cases, the PFC provides the capability to recirculate the packets. Recirculation can be used to perform additional lookups in the ACL or QoS TCAMs, the NetFlow table, or the FIB TCAM table. Recirculation is necessary in these situations:

- To push more than three labels on imposition
- To pop more than two labels on disposition
- To pop an explicit null top label
- When the VPN Routing and Forwarding (VRF) number is more than 511
- For IP ACL on the egress interface (for nonaggregate (per-prefix) labels only)

Packet recirculation occurs only on a particular packet flow; other packet flows are not affected. The rewrite of the packet occurs on the modules; the packets are then forwarded back to the PFC for additional processing.

Hardware-Supported MPLS Functions

The PFC supports the following MPLS functions in hardware:

- Label operation— Any number of labels can be pushed or popped, although for best results, up to three labels can be pushed, and up to two labels can be popped in the same operation.
- IP to MPLS path—IP packets can be received and sent to the MPLS path.
- MPLS to IP path—Labeled packets can be received and sent to the IP path.
- MPLS to MPLS path—Labeled packets can be received and sent to the label path.
- MPLS Traffic Engineering (MPLS TE)—Enables an MPLS backbone to replicate and expand the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks.
- Time to live (TTL) operation—At the ingress edge of the MPLS network, the TTL value in the MPLS frame header can be received from either the TTL field of the IP packet header or the user-configured value from the adjacency entry. At the egress of the MPLS network, the final TTL equals the minimum (label TTL and IP TTL)-1.

Note With the Uniform mode, the TTL is taken from the IP TTL; with the Pipe mode, a value of 255, taken from the hardware register, is used for the outgoing label.

- QoS—Information on Differentiated Services (DiffServ) and ToS from IP packets can be mapped to MPLS EXP field.
- MPLS/VPN Support—Up to 1024 VRFs can be supported (over 511 VRFs requires recirculation).
- Ethernet over MPLS—The Ethernet frame can be encapsulated at the ingress to the MPLS domain and the Ethernet frame can be decapsulated at the egress.
- Packet recirculation—The PFC provides the capability to recirculate the packets. See the "Recirculation" section on page 32-4.
- Configuration of MPLS switching is supported on VLAN interfaces with the **mpls ip** command.

Supported Cisco IOS Features

The following Cisco IOS software features are supported:

- Multi-VPN Routing and Forwarding (VRF) for CE Routers (VRF Lite)—VRF-lite is a feature that enables a service provider to support two or more VPNs (using only VRF-based IPv4), where IP addresses can be overlapped among the VPNs. See this publication for more information:

  http://www.cisco.com/en/US/products/hw/routers/ps259/prod_bulletin09186a00800921d7.html

  VRF Lite is supported with the following features:

  - IPv4 forwarding between VRFs interfaces
  - IPv4 ACLs
  - IPv4 HSRP
- MPLS on Cisco routers—This feature provides basic MPLS support for imposing and removing labels on IP packets at label edge routers (LERs) and switching labels at label switch routers (LSRs). See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mpls/config_library/12-2sx/mp-12-2sx-library.html

- MPLS TE—MPLS traffic engineering software enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. MPLS traffic engineering thereby makes traditional Layer 2 features available to Layer 3 traffic flows. For more information, see these publications:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mpls/config_library/12-2sx/mp-12-2sx-library.html

  http://www.cisco.com/en/US/tech/tk436/tk428/technologies_configuration_example09186a0080093fcb.shtml

  http://www.cisco.com/en/US/tech/tk436/tk428/technologies_configuration_example09186a0080093fd0.shtml

- MPLS TE DiffServ Aware (DS-TE)—This feature provides extensions made to MPLS TE to make it DiffServ aware, allowing constraint-based routing of guaranteed traffic. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_te_diffserv/configuration/12-2sx/mp-te-diffserv-12-2sx-book.html

- MPLS TE Forwarding Adjacency—This feature allows a network administrator to handle a traffic engineering, label-switched path (LSP) tunnel as a link in an Interior Gateway Protocol (IGP) network based on the Shortest Path First (SPF) algorithm. For information on forwarding adjacency with Intermediate System-to-Intermediate System (IS-IS) routing, see this publication:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_te_path_setup/configuration/12-2sx/mp-te-fwd-adjacency.html

- MPLS TE Interarea Tunnels—This feature allows the router to establish MPLS TE tunnels that span multiple Interior Gateway Protocol (IGP) areas and levels, removing the restriction that had required the tunnel head-end and tail-end routers to be in the same area. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_te_path_setup/configuration/12-2sx/mp-te-interarea-tun.html

- MPLS virtual private networks (VPNs)—This feature allows you to deploy scalable IPv4 Layer 3 VPN backbone services over a Cisco IOS network. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mpls/config_library/12-2sx/mp-12-2sx-library.html

- MPLS VPN Carrier Supporting Carrier (CSC)—This feature enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_ias_and_csc/configuration/12-2sx/mp-carrier-ldp-igp.html

- MPLS VPN Carrier Supporting Carrier IPv4 BGP Label Distribution—This feature allows you to configure your CSC network to enable Border Gateway Protocol (BGP) to transport routes and MPLS labels between the backbone carrier provider edge (PE) routers and the customer carrier customer edge (CE) routers. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_ias_and_csc/configuration/12-2sx/mp-carrier-bgp.html

- MPLS VPN Interautonomous System (InterAS) Support —This feature allows an MPLS VPN to span service providers and autonomous systems. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_ias_and_csc/configuration/12-2sx/mp-ias-and-csc-12-2sx-book.html

- MPLS VPN Inter-AS IPv4 BGP label distribution—This feature enables you to set up a VPN service provider network so that the autonomous system boundary routers (ASBRs) exchange IPv4 routes with MPLS labels of the PE routers. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_ias_and_csc/configuration/12-2sx/mp-carrier-ldp-igp.html
- HSRP Support for MPLS VPNs—This feature ensures that the HSRP virtual IP address is added to the correct IP routing table and not to the global routing table. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/ipapp_fhrp/configuration/12-2sx/fhp-hsrp.html
- OSPF Sham-Link Support for MPLS VPN—This feature allows you to use a sham-link to connect VPN client sites that run the Open Shortest Path First (OSPF) protocol and share OSPF links in a MPLS VPN configuration. See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_ospf/configuration/15-sy/iro-sham-link.html
- Any Transport over MPLS (AToM)—Transports Layer 2 packets over an MPLS backbone. See the "Any Transport over MPLS" section on page 32-17.

MPLS Guidelines and Restrictions

When configuring MPLS, follow these guidelines and restrictions:

- The PFC3 and DFC3s support up to 16 load-shared paths (Cisco IOS releases for other platforms support only 8 load-shared paths).
- MTU size checking is supported in hardware.
- Fragmentation is supported in software, including traffic that ingresses as IP and egresses as MPLS. To prevent excessive CPU utilization, you can rate-limit the traffic being sent to the RP for fragmentation with the **mls rate-limit all mtu-failure** command.

Note For information on other limitations and restrictions, see the "MPLS VPN Guidelines and Restrictions" section on page 32-14 and the "EoMPLS Guidelines and Restrictions" section on page 32-18.

Supported MPLS Commands

Release 12.2SX supports these MPLS commands:

- **mpls ip default route**
- **mpls ip propagate-ttl**
- **mpls ip ttl-expiration pop**
- **mpls label protocol**
- **mpls label range**
- **mpls ip**
- **mpls label protocol**
- **mpls mtu**

For information about these commands, see this publication:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Configuring MPLS

For information about configuring MPLS, see the *Multiprotocol Label Switching on Cisco Routers* publication at the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcftagc.html

MPLS Per-Label Load Balancing

The following sections provide information on basic MPLS, MLPS Layer 2 VPN, and MPLS Layer 3 VPN load balancing.

Basic MPLS Load Balancing

The maximum number of load balancing paths is 8. The PFC forwards MPLS labeled packets without explicit configuration. If the packet has three labels or less and the underlying packet is IPv4, then the PFC uses the source and destination IPv4 address. If the underlying packet is not IPv4, or if more than three labels are present, the PFC parses down as deep as the fifth or lowest label and uses it for hashing.

MPLS Layer 2 VPN Load Balancing

Load balancing is based on the VC label in the MPLS core if the first nibble of the MAC address in the customer Ethernet frame is not 4.

Note Load balancing is not supported at the ingress PE for Layer 2 VPNs.

MPLS Layer 3 VPN Load Balancing

MPLS Layer 3 VPN load balancing is similar to basic MPLS load balancing. For more information, see the "Basic MPLS Load Balancing" section on page 32-8.

MPLS Configuration Examples

The following is an example of a basic MPLS configuration:

```
*****
Basic MPLS
*****


IP ingress interface:
```

```
Router# mpls label protocol ldp

interface GigabitEthernet6/2
 ip address 75.0.77.1 255.255.255.0
 media-type rj45
 speed 1000
end

Label egress interface:

interface GigabitEthernet7/15
 mtu 9216
 ip address 75.0.67.2 255.255.255.0
 logging event link-status
 mpls ip


Router# show ip route 188.0.0.0
Routing entry for 188.0.0.0/24, 1 known subnets

O IA    188.0.0.0 [110/1] via 75.0.77.2, 00:00:10, GigabitEthernet6/2


Router# show ip routing 88.0.0.0
Routing entry for 88.0.0.0/24, 1 known subnets

O E2    88.0.0.0 [110/0] via 75.0.67.1, 00:00:24, GigabitEthernet7/15
                 [110/0] via 75.0.21.2, 00:00:24, GigabitEthernet7/16
Router# show mpls forwarding-table 88.0.0.0
Local  Outgoing    Prefix             Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id       switched   interface
30     50          88.0.0.0/24       0          Gi7/15     75.0.67.1
       50          88.0.0.0/24       0          Gi7/16     75.0.21.2


Router# show mls cef 88.0.0.0 detail

Codes: M - mask entry, V - value entry, A - adjacency index, P - priority bit
       D - full don't switch, m - load balancing modnumber, B - BGP Bucket sel
       V0 - Vlan 0,C0 - don't comp bit 0,V1 - Vlan 1,C1 - don't comp bit 1
       RVTEN - RPF Vlan table enable, RVTSEL - RPF Vlan table select
Format: IPV4_DA - (8 | xtag vpn pi cr recirc tos prefix)
Format: IPV4_SA - (9 | xtag vpn pi cr recirc prefix)
M(3223   ): E | 1 FFF  0 0 0 0   255.255.255.0
V(3223   ): 8 | 1 0    0 0 0 0   88.0.0.0           (A:344105 ,P:1,D:0,m:1 ,B:0 )
M(3223   ): E | 1 FFF  0 0 0 255.255.255.0
V(3223   ): 9 | 1 0    0 0 0 88.0.0.0 (V0:0    ,C0:0 ,V1:0    ,C1:0 ,RVTEN:0 ,RVTSEL:0  )
Router# show mls cef adj ent 344105

Index: 344105  smac: 0005.9a39.a480, dmac: 000a.8ad8.2340
               mtu: 9234, vlan: 1031, dindex: 0x0, l3rw_vld: 1
               packets: 109478260, bytes: 7006608640

Router# show mls cef adj ent 344105 de

Index: 344105  smac: 0005.9a39.a480, dmac: 000a.8ad8.2340
               mtu: 9234, vlan: 1031, dindex: 0x0, l3rw_vld: 1
               format: MPLS, flags: 0x1000008418
               label0: 0, exp: 0, ovr: 0
               label1: 0, exp: 0, ovr: 0
               label2: 50, exp: 0, ovr: 0
               op: PUSH_LABEL2
               packets: 112344419, bytes: 7190042816
```

DE/CLP and EXP Mapping on FR/ATMoMPLS VC

Note Cisco IOS Release 12.2(33)SXH and later releases do not support any Frame Relay interfaces.

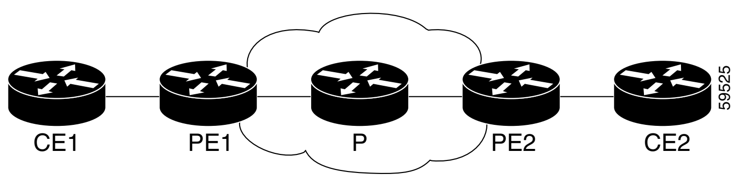
The DE/CLP and EXP Mapping on FR/ATMoMPLS VC feature allows you to map the ATM Congestion Loss Priority (CLP) bit to the MPLS EXP value at the ingress to an MPLS AToM network and to map the MPLS EXP value to the ATM CLP bit at the egress of an MPLS AToM network.

The ATM CLP bit indicates whether the cell may be discarded if it encounters extreme congestion as it moves through the network.

In the figure below, the PE1 tags the incoming packet with the MPLS EXP value and sends the packet to the next hop. At each hop, matching is on the EXP value. At the PE2 egress, however, the packet is no longer MPLS but IP, so matching cannot occur on the EXP value.

Internally, the OSM preserves the EXP value in the QoS group so matching on the QoS group at the PE2 egress provides the same effect as matching on the EXP value.

Figure 32-3 CLP and EXP Mapping



See the following sections:

- Match on ATM CLP Bit, page 32-10
- Set on ATM CLP Bit, page 32-12

Match on ATM CLP Bit

Use Match on ATM CLP Bit at the ingress to an MPLS AToM network to map the ATM cell loss priority (CLP) of the packet arriving at an interface to the EXP value, and then apply the desired QoS functionality and actions (for example, traffic policing) to those packets.

Restrictions for Match on ATM CLP Bit

This feature is supported on policy maps attached to ATM permanent virtual circuits (PVCs) only.

Configuring Match on ATM CLP Bit for Ingress Policy

Perform the following steps to configure Match on ATM CLP Bit for the ingress policy:

	Command or Action	Purpose
Step 1	Router# **enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Router(config)# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **class-map** *class-name*	Specifies the user-defined name of the traffic class.
Step 4	Router(config-cmap)# **match atm clp**	Enables packet matching on the basis of the ATM CLP bit set to 1.
Step 5	Router(config-cmap)# **policy-map** *policy-name*	Specifies the name of the traffic policy to configure.
Step 6	Router(config-pmap)# **class** *class-name*	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 7	Router(config-pmap-c)# **set mpls experimental** *value*	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
Step 8	Router(config)# **interface atm** *interface-number*	Enters interface configuration mode.
Step 9	Router(config-if)# **pvc** [**name**] *vpi/vci* [**l2transport**]	Entesr ATM virtual circuit configuration mode.
Step 10	Router(config-if)# **service-policy input** *policy-name*	Attaches a traffic policy to an interface.

The following is an example of a Match on ATM CLP Bit configuration:

```
Router# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#class-map CLP
Router(config-cmap)#match atm clp
Router(config-cmap)#exit
Router(config)#policy-map CLP2EXP
Router(config-pmap)#class CLP
Router(config-pmap-c)#set mpls experimental 1
Router(config-pmap-c)#exit
Router(config-pmap)#interface ATM3/0
Router(config-if)#pvc 1/100
Router(cfg-if-atm-l2trans-pvc)#service-policy input CLP2EXP
Router(cfg-if-atm-l2trans-pvc)#end
Router#
```

Use the **show policy-map interface** command to verify the Match on ATM CLP bit as in the following example:

```
CFLOW_PE1# show policy-map interface a3/0
ATM3/0/0: VC 1/100 -

Service-policy input: CLP2EXP

Class-map: CLP (match-all)
200 packets, 22400 bytes
5 minute offered rate 2000 bps, drop rate 0 bps
Match: atm clp
QoS Set
mpls experimental imposition 1
```

```
Packets marked 200

Class-map: class-default (match-any)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: any
CFLOW_PE1#
```

Set on ATM CLP Bit

Use Set on ATM CLP Bit at the egress of an MPLS AToM network to map the EXP value to the ATM CLP bit.

Restrictions for Set on ATM CLP Bit

This feature is supported on policy maps attached to ATM permanent virtual circuits (PVCs) only.

Configuring Set on ATM CLP Bit for Egress Policy

Perform the following steps to configure Set on ATM CLP Bit for the ingress policy:

	Command or Action	Purpose
Step 1	`Router#` **`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router(config)#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`class-map`** *`class-name`*	Specifies the user-defined name of the traffic class.
Step 4	`Router(config-cmap)#` **`match qos-group`** *`qos-group-value`*	Identifies a specific quality of service (QoS) group value as a match criterion. The QoS group value has no mathematical significance. **Note** The QoS group concept is directly derived from the incoming MPLS EXP value and is valid only with AToM. You cannot use MQC to set QoS group value.
Step 5	`Router(config-cmap)#` **`policy-map`** *`policy-name`*	Specifies the name of the traffic policy to configure.
Step 6	`Router(config-pmap)#` **`class`** *`class-name`*	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 7	`Router(config-pmap-c)#` **`set atm-clp`**	Sets the cell loss priority (CLP) bit when a policy map is configured.
Step 8	`Router(config)#` **`interface`** *`slot/port`*	Enters the interface and enters interface configuration mode.
Step 9	`Router(config-if)#` **`service-policy input`** *`policy-name`*	Attaches a traffic policy to an interface.

The following shows how to configure Set on ATM CLP Bit:

```
arthos# show policy-map qg2clp
```

```
 Policy Map qg2clp
   Class qg1
     set atm-clp
arthos# show class-map qg1
Class Map match-all qg1 (id 23)
  Match qos-group 1

arthos# show run int a9/1
interface ATM9/1
no ip address
atm clock INTERNAL
atm mtu-reject-call
mls qos trust dscp
pvc 1/100 l2transport
  encapsulation aal5
  mpls l2transport route 101.101.101.101 1000
  service-policy out qg2clp
```

Verify the configuration with the **show policy-map interface** command.

```
arthos# show policy interface ATM9/1
ATM9/1: VC 1/100 -

 Service-policy output: qg2clp

   Class-map: qg1 (match-all)
     1000 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: qos-group 1
     QoS Set
       atm-clp
         Packets marked 1000
```

VPN Switching

These sections describe VPN switching:

- VPN Switching Operation, page 32-13
- MPLS VPN Guidelines and Restrictions, page 32-14
- MPLS VPN Supported Commands, page 32-14
- MPLS VPN Sample Configuration, page 32-15

VPN Switching Operation

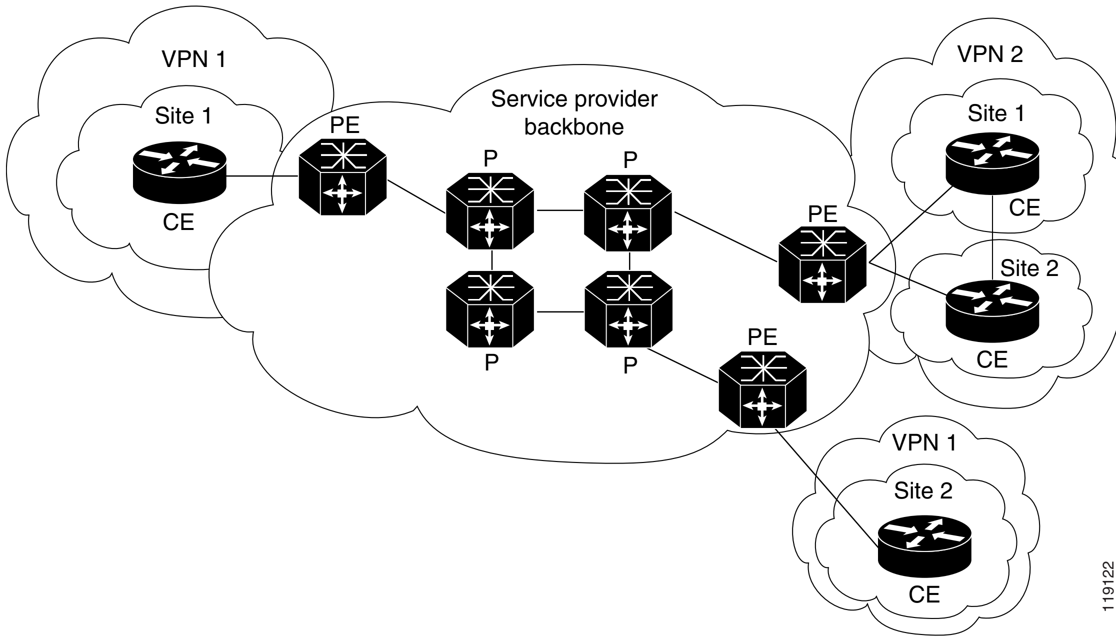
The IP VPN feature for MPLS allows a Cisco IOS network to deploy scalable IP Layer 3 VPN backbone services to multiple sites deployed on a shared infrastructure while also providing the same access or security policies as a private network. VPN based on MPLS technology provides the benefits of routing isolation and security, as well as simplified routing and better scalability.

See the Cisco IOS software documentation for a conceptual MPLS VPN overview and configuration details at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcftagov_ps1835_TSD_Products_Configuration_Guide_Chapter.html

A typical MPLS VPN network topology is shown in Figure 32-4.

Figure 32-4 VPNs with Service Provider Backbone



At the ingress PE, the PFC makes a forwarding decision based on the packet headers. The PFC contains a table that maps VLANs to VPNs. In the switch architecture, all physical ingress interfaces in the system are associated with a specific VPN. The PFC looks up the IP destination address in the CEF table but only against prefixes that are in the specific VPN. (The table entry points to a specific set of adjacencies and one is chosen as part of the load-balancing decision if multiple parallel paths exist.)

The table entry contains the information on the Layer 2 header that the packet needs, as well as the specific MPLS labels to be pushed onto the frame. The information to rewrite the packet goes back to the ingress module where it is rewritten and forwarded to the egress line interface.

VPN traffic is handled at the egress from the PE based upon the per-prefix labels or aggregate labels. If per-prefix labels are used, then each VPN prefix has a unique label association; this allows the PE to forward the packet to the final destination based upon a label lookup in the FIB.

Note The PFC allocates only one aggregate label per VRF.

If aggregate labels are used for disposition in an egress PE, many prefixes on the multiple interfaces may be associated with the label. In this case, the PFC must perform an IP lookup to determine the final destination. The IP lookup may require recirculation.

MPLS VPN Guidelines and Restrictions

When configuring MPLS VPN, note that VPNs are recirculated when the number of VPNs is over 511.

MPLS VPN Supported Commands

MPLS VPN supports these commands:

- **address-family**
- **exit-address-family**
- **import map**
- **ip route vrf**
- **ip route forwarding**
- **ip vrf**
- **neighbor activate**
- **rd**
- **route-target**

For information about these commands, see this publication:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Configuring MPLS VPN

For information on configuring MPLS VPN, see the *MPLS Virtual Private Networks* feature module at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcftagc_ps1835_TSD_Products_Configuration_Guide_Chapter.html#Configuring_MPLS_Virtual_Private_Networks

Note If you use a Layer 3 VLAN interface as the MPLS uplink through a Layer 2 port peering with another MPLS device, then you can use another Layer 3 VLAN interface as the VRF interface.

MPLS VPN Sample Configuration

This sample configuration shows LAN and FlexWAN CE-facing interfaces. MPLS switching configuration in Cisco IOS Release 12.2SX is identical to configuration in other releases.

```
!ip vrf blues
 rd 100:10
 route-target export 100:1
 route-target import 100:1
!
mpls label protocol ldp
mpls ldp logging neighbor-changes
mls mpls tunnel-recir
!
interface Loopback0
 ip address 10.4.4.4 255.255.255.255
!
interface GigabitEthernet4/2
 description Catalyst link to P2
 no ip address
 mls qos trust dscp
!
interface GigabitEthernet4/2.42
 encapsulation dot1Q 42
 ip address 10.0.3.2 255.255.255.0
 tag-switching ip
!
```

```
interface GigabitEthernet7/3
 description Catalyst link to CE2
 no ip address
 mls qos trust dscp
!
interface GigabitEthernet7/3.73
 encapsulation dot1Q 73
 ip vrf forwarding blues
 ip address 10.19.7.1 255.255.255.0
!
interface POS8/1
 description OSM link to CE3
 ip vrf forwarding blues
 ip address 10.19.8.1 255.255.255.252
 encapsulation ppp
 mls qos trust dscp
 pos scramble-atm
 pos flag c2 22
!
interface POS9/0/0
 description FlexWAN link to CE1
 ip vrf forwarding blues
 ip address 10.19.9.1 255.255.255.252
 encapsulation ppp
 pos scramble-atm
 pos flag c2 22
!
router ospf 100
 log-adjacency-changes
 network 10.4.4.4 0.0.0.0 area 0
 network 10.0.0.0 0.0.255.255 area 0
!
router ospf 65000 vrf blues
 log-adjacency-changes
 redistribute bgp 100 subnets
 network 10.19.0.0 0.0.255.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.3.3.3 remote-as 100
 neighbor 10.3.3.3 description MP-BGP to PE1
 neighbor 10.3.3.3 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 10.3.3.3 activate
 neighbor 10.3.3.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf blues
 redistribute connected
 redistribute ospf 65000 match internal external 1 external 2
 no auto-summary
 no synchronization
 exit-address-family
!
```

Any Transport over MPLS

Any Transport over MPLS (AToM) transports Layer 2 packets over an MPLS backbone. AToM uses a directed Label Distribution Protocol (LDP) session between edge routers for setting up and maintaining connections. Forwarding occurs through the use of two level labels that provide switching between the edge routers. The external label (tunnel label) routes the packet over the MPLS backbone to the egress PE at the ingress PE. The VC label is a demuxing label that determines the connection at the tunnel endpoint (the particular egress interface on the egress PE as well as the VLAN identifier for an Ethernet frame).

AToM supports the following like-to-like transport types in Cisco IOS Release 12.2SX:

- Ethernet over MPLS (EoMPLS) (VLAN mode and port mode)
- Frame Relay over MPLS with DLCI-to-DLCI connections
- ATM AAL5 over MPLS
- ATM Cell Relay over MPLS

Note Additional AToM types are planned in future releases.

Cisco IOS Release 12.2SX supports both PFC-accelerated EoMPLS as well as FlexWAN2-based EoMPLS. For more information, see this publication:

http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SX_OSM_config/mpls.html#Ethernet_over_MPLS

For information on other AToM implementations (ATM AAL5 over MPLS, ATM Cell Relay over MPLS, Frame Relay over MPLS), see this publication:

http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SX_OSM_config/mpls.html#Any_Transport_over_MPLS

These sections describe AToM:

AToM Load Balancing

EoMPLS in Cisco IOS Release 12.2SX does not support load balancing at the tunnel ingress; only one Interior Gateway Protocol (IGP) path is selected even if multiple IGP paths are available, but load balancing is available at the MPLS core.

Understanding EoMPLS

EoMPLS is one of the AToM transport types. AToM transports Layer 2 packets over a MPLS backbone using a directed LDP session between edge routers for setting up and maintaining connections. Forwarding occurs through the use of two level labels that provide switching between the edge routers.

The external label (tunnel label) routes the packet over the MPLS backbone to the egress PE at the ingress PE. The VC label is a demuxing label that determines the connection at the tunnel endpoint (the particular egress interface on the egress PE as well as the VLAN identifier for an Ethernet frame).

EoMPLS works by encapsulating Ethernet PDUs in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet.

Note Use FlexWAN-based EoMPLS when you want local Layer 2 switching and EoMPLS on the same VLAN. You need to configure EoMPLS on the SVI; the core-facing card must be a FlexWAN module. When local Layer 2 switching is not required, use PFC-based EoMPLS configured on the subinterface or physical interface.

EoMPLS Guidelines and Restrictions

When configuring EoMPLS, follow these guidelines and restrictions:

- Ensure that the maximum transmission unit (MTU) of all intermediate links between endpoints is sufficient to carry the largest Layer 2 packet received.
- EoMPLS supports VLAN packets that conform to the IEEE 802.1Q standard. The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames.
- If QoS is disabled globally, both the 802.1p and IP precedence bits are preserved. When the QoS is enabled on a Layer 2 port, either 802.1q P bits or IP precedence bits can be preserved with the trusted configuration. However, by default the unpreserved bits are overwritten by the value of preserved bits. For instance, if you preserve the P bits, the IP precedence bits are overwritten with the value of the P bits. Cisco IOS Release 12.2SX provides a new command that allows you to trust the P bits while preserving the IP precedence bits. To preserve the IP precedence bits, use the **no mls qos rewrite ip dscp** command.
  - The **no mls qos rewrite ip dscp** command is not compatible with the MPLS and MPLS VPN features. See Chapter 43, "Configuring PFC QoS."
  - Do not use the **no mls qos rewrite ip dscp** command if you have PFC-based EoMPLS and PXF-based EoMPLS services in the same system.
- EoMPLS is not supported with private VLANs.
- The following restrictions apply to using trunks with EoMPLS:
  - To support Ethernet spanning tree bridge protocol data units (BPDUs) across an EoMPLS cloud, you must disable spanning tree for the Ethernet-over-MPLS VLAN. This ensures that the EoMPLS VLANs are carried only on the trunk to the customer switch. Otherwise, the BPDUs are not directed to the EoMPLS cloud.
  - The native VLAN of a trunk must not be configured as an EoMPLS VLAN.
- In Cisco IOS Release 12.2SX, all protocols (for example, CDP, VTP, BPDUs) are tunneled across the MPLS cloud without conditions.
- ISL encapsulation is not supported for the interface that receives EoMPLS packets.
- Unique VLANs are required across interfaces. You cannot use the same VLAN ID on different interfaces.
- EoMPLS tunnel destination route in the routing table and the CEF table must be a /32 address (host address where the mask is 255.255.255.255) to ensure that there is a label-switched path (LSP) from PE to PE.

- For a particular EoMPLS connection, both the ingress EoMPLS interface on the ingress PE and the egress EoMPLS interface on the egress PE have to be subinterfaces with dot1Q encapsulation or neither is a subinterface.
- 802.1Q in 802.1Q over EoMPLS is supported if the outgoing interface connecting to MPLS network is a port on an Layer 2 card.
- Shaping EoMPLS traffic is not supported if the egress interface connecting to an MPLS network is a Layer 2 LAN port (a mode known as PFC-based EoMPLS).
- EoMPLS based on a PFC does not perform any Layer 2 lookup to determine if the destination MAC address resides on the local or remote segment and does not perform any Layer 2 address learning (as traditional LAN bridging does). This functionality (local switching) is available only when using FlexWAN modules as uplinks.
- In previous releases of AToM, the command used to configure AToM circuits was **mpls l2 transport route**. This command has been replaced with the **xconnect** command. You can use the **xconnect** command to configure EoMPLS circuits.
- The AToM control word is not supported.
- EoMPLS is not supported on Layer 3 VLAN interfaces.
- Point-to-point EoMPLS works with a physical interface and subinterfaces.

Configuring EoMPLS

These sections describe how to configure EoMPLS:

- Prerequisites, page 32-19
- Configuring VLAN-Based EoMPLS, page 32-20
- Configuring Port-Based EoMPLS, page 32-23

Prerequisites

Before you configure EoMPLS, ensure that the network is configured as follows:

- Configure IP routing in the core so that the PE routers can reach each other through IP.
- Configure MPLS in the core so that a label switched path (LSP) exists between the PE routers.

EoMPLS works by encapsulating Ethernet PDUs in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet. Two methods are available to configure EoMPLS in Cisco IOS Release 12.2SX:

- VLAN mode—Transports Ethernet traffic from a source 802.1Q VLAN to a destination 802.1Q VLAN through a single VC over an MPLS network. VLAN mode uses VC type 5 as default (no dot1q tag) and VC type 4 (transport dot1 tag) if the remote PE does not support VC type 5 for subinterface (VLAN) based EoMPLS.
- Port mode—Allows all traffic on a port to share a single VC across an MPLS network. Port mode uses VC type 5.

Note
- For both VLAN mode and port mode, EoMPLS in Cisco IOS Release 12.2SX does not allow local switching of packets between interfaces unless you use loopback ports.

- A system can have both FlexWAN configuration and PFC configuration enabled at the same time. Cisco supports this configuration but does not recommend it. Unless the uplinks to the MPLS core are through FlexWAN interfaces, FlexWAN-based EoMPLS connections will not be active; this causes packets for FlexWAN-based EoMPLS arriving on non-WAN interfaces to be dropped. For information on FlexWAN EoMPLS, see this publication:

  http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SX_OSM_config/mpls.html#Ethernet_over_MPLS

In Cisco IOS Release 12.2SX, LAN ports can receive Layer 2 traffic, impose labels, and switch the frames into the MPLS core without using a FlexWAN module.

With Cisco IOS Release 12.2SX, you can configure a FlexWAN module to face the core of MPLS network and use either the FlexWAN configuration or the PFC configuration. For more information, see this publication:

http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SX_OSM_config/mpls.html#Ethernet_over_MPLS

Configuring VLAN-Based EoMPLS

When configuring VLAN-based EoMPLS, follow these guidelines and restrictions:

- The AToM control word is not supported.
- Ethernet packets with hardware-level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.
- You must configure VLAN-based EoMPLS on subinterfaces.

To configure VLAN-based EoMPLS, perform this task on the provider edge (PE) routers:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface gigabitethernet***slot/interface.subinterface*	Specifies the Gigabit Ethernet subinterface. Make sure that the subinterface on the adjoining CE router is on the same VLAN as this PE router.
Step 3	Router(config-if)# **encapsulation dot1q** *vlan_id*	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers do not need to be on the same subnet.
Step 4	Router(config-if)# **xconnect** *peer_router_id vcid* **encapsulation mpls**	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

This is a VLAN-based EoMPLS configuration sample:

```
!
interface GigabitEthernet7/4.2
encapsulation dot1Q 3
xconnect 13.13.13.13 3 encapsulation mpls
no shut
```

Note The IP address is configured on subinterfaces of the CE devices.

Verifying the Configuration

To verify and display the configuration of Layer 2 VLAN transport over MPLS tunnels, perform the following:

- To display a single line for each VLAN, naming the VLAN, status, and ports, enter the **show vlan brief** command.

```
Router# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------
1    default                          active
2    VLAN0002                         active
3    VLAN0003                         active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

- To make sure that the PE router endpoints have discovered each other, enter the **show mpls ldp discovery** command. When an PE router receives an LDP hello message from another PE router, it considers that router and the specified label space to be "discovered."

```
Router# show mpls ldp discovery
 Local LDP Identifier:
    13.13.13.13:0
    Discovery Sources:
    Interfaces:
        GE-WAN3/3 (ldp): xmit/recv
            LDP Id: 12.12.12.12:0
    Targeted Hellos:
        13.13.13.13 -> 11.11.11.11 (ldp): active/passive, xmit/recv
            LDP Id: 11.11.11.11:0
```

- To make sure that the label distribution session has been established, enter the **show mpls ldp neighbor** command. The third line of the output shows that the state of the LDP session is operational and shows that messages are being sent and received.

```
Router# show mpls ldp neighbor
    Peer LDP Ident: 12.12.12.12:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 12.12.12.12.646 - 13.13.13.13.11010
        State: Oper; Msgs sent/rcvd: 1649/1640; Downstream
        Up time: 23:42:45
        LDP discovery sources:
          GE-WAN3/3, Src IP addr: 34.0.0.2
        Addresses bound to peer LDP Ident:
          23.2.1.14       37.0.0.2        12.12.12.12     34.0.0.2
          99.0.0.1
    Peer LDP Ident: 11.11.11.11:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 11.11.11.11.646 - 13.13.13.13.11013
        State: Oper; Msgs sent/rcvd: 1650/1653; Downstream
        Up time: 23:42:29
        LDP discovery sources:
          Targeted Hello 13.13.13.13 -> 11.11.11.11, active, passive
        Addresses bound to peer LDP Ident:
          11.11.11.11     37.0.0.1        23.2.1.13
```

- To ensure that the label forwarding table is built correctly, enter the **show mpls forwarding-table** command to verify that a label has been learned for the remote PE and that the label is going from the correct interface to the correct next-hop.

```
Router# show mpls forwarding-table
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Untagged    223.255.254.254/32   \
                                     0          Gi2/1      23.2.0.1
20     Untagged    l2ckt(2)          133093     Vl2        point2point
21     Untagged    l2ckt(3)          185497     Vl3        point2point
24     Pop tag     37.0.0.0/8        0          GE3/3      34.0.0.2
25     17          11.11.11.11/32    0          GE3/3      34.0.0.2
26     Pop tag     12.12.12.12/32    0          GE3/3      34.0.0.2
Router#
```

The output shows the following data:

  - Local tag—Label assigned by this router.
  - Outgoing tag or VC—Label assigned by next hop.
  - Prefix or Tunnel Id—Address or tunnel to which packets with this label are going.
  - Bytes tag switched— Number of bytes switched out with this incoming label.
  - Outgoing interface—Interface through which packets with this label are sent.
  - Next Hop—IP address of neighbor that assigned the outgoing label.

- To view the state of the currently routed VCs, enter the **show mpls l2transport vc** command.

```
Router# show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Vl2            Eth VLAN 2           11.11.11.11     2          UP
Vl3            Eth VLAN 3           11.11.11.11     3          UP
```

To see detailed information about each VC, add the keyword **detail.**

```
Router# show mpls l2transport vc detail
Local interface: Vl2 up, line protocol up, Eth VLAN 2 up
  Destination address: 11.11.11.11, VC ID: 2, VC status: up
    Tunnel label: 17, next hop 34.0.0.2
    Output interface: GE3/3, imposed label stack {17 18}
  Create time: 01:24:44, last status change time: 00:10:55
  Signaling protocol: LDP, peer 11.11.11.11:0 up
    MPLS VC labels: local 20, remote 18
    Group ID: local 71, remote 89
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 1009, send 1019
    byte totals:   receive 133093, send 138089
    packet drops:  receive 0, send 0

Local interface: Vl3 up, line protocol up, Eth VLAN 3 up
  Destination address: 11.11.11.11, VC ID: 3, VC status: up
    Tunnel label: 17, next hop 34.0.0.2
    Output interface: GE3/3, imposed label stack {17 19}
  Create time: 01:24:38, last status change time: 00:10:55
  Signaling protocol: LDP, peer 11.11.11.11:0 up
    MPLS VC labels: local 21, remote 19
    Group ID: local 72, remote 90
```

```
  MTU: local 1500, remote 1500
  Remote interface description:
Sequencing: receive disabled, send disabled
VC statistics:
  packet totals: receive 1406, send 1414
  byte totals:   receive 185497, send 191917
  packet drops:  receive 0, send 0
```

Configuring Port-Based EoMPLS

When configuring port-based EoMPLS in Cisco IOS Release 12.2SX, follow these guidelines and restrictions:

- The AToM control word is not supported.
- Ethernet packets with hardware-level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.
- Port-based EoMPLS and VLAN-based EoMPLS are mutually exclusive. If you enable a main interface for port-to-port transport, you also cannot enter commands on a subinterface.

To support 802.1Q-in-802.1Q traffic and Ethernet traffic over EoMPLS in Cisco IOS Release 12.2SX, configure port-based EoMPLS by performing this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface gigabitethernet***slot/interface*	Specifies the Gigabit Ethernet interface. Make sure that the interface on the adjoining CE router is on the same VLAN as this PE router.
Step 3	Router(config-if)# **xconnect** *peer_router_id vcid* **encapsulation mpls**	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

The following is an example of a port-based configuration:

```
!
EoMPLS:

router# show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Fa8/48         Ethernet             75.0.78.1       1          UP
Gi7/11.2000    Eth VLAN 2000        75.0.78.1       2000       UP


Port-Based EoMPLS Config:

router# show run interface f8/48
Building configuration...

Current configuration : 86 bytes
!
interface FastEthernet8/48
 no ip address
 xconnect 75.0.78.1 1 encapsulation mpls
end

Sub-Interface Based Mode:
```

```
router# show run interface g7/11
Building configuration...

Current configuration : 118 bytes
!
interface GigabitEthernet7/11
 description Traffic-Generator
 no ip address
 logging event link-status
 speed nonegotiate
end

router# show run int g7/11.2000
Building configuration...

Current configuration : 112 bytes
!
interface GigabitEthernet7/11.2000
 encapsulation dot1Q 2000
 xconnect 75.0.78.1 2000 encapsulation mpls
end

kb7606# show mpls l2transport vc 1 detail
Local interface: Gi7/47 up, line protocol up, Ethernet up
  Destination address: 75.0.80.1, VC ID: 1, VC status: up
    Tunnel label: 5704, next hop 75.0.83.1
    Output interface: Te8/3, imposed label stack {5704 10038}
  Create time: 00:30:33, last status change time: 00:00:43
  Signaling protocol: LDP, peer 75.0.80.1:0 up
    MPLS VC labels: local 10579, remote 10038
    Group ID: local 155, remote 116
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 26, send 0
    byte totals:   receive 13546, send 0
    packet drops:  receive 0, send 0
```

To obtain the VC type:

```
kb7606# remote command switch show mpls l2transport vc 1 de

Local interface: GigabitEthernet7/47, Ethernet
  Destination address: 75.0.80.1, VC ID: 1
  VC status: receive UP, send DOWN
  VC type: receive 5, send 5
    Tunnel label: not ready, destination not in LFIB
    Output interface: unknown, imposed label stack {}
    MPLS VC label: local 10579, remote 10038
  Linecard VC statistics:
    packet totals:  receive: 0  send: 0
    byte totals:  receive: 0  send: 0
    packet drops:  receive: 0  send: 0
  Control flags:
    receive 1,  send: 31
!
```

Verifying the Configuration

To verify and display the configuration of Layer 2 VLAN transport over MPLS tunnels, perform the following:

- To display a single line for each VLAN, naming the VLAN, status, and ports, enter the **show vlan brief** command.

```
Router# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active
2    VLAN0002                         active    Gi1/4
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

- To make sure the PE router endpoints have discovered each other, enter the **show mpls ldp discovery** command. When an PE router receives an LDP Hello message from another PE router, it considers that router and the specified label space to be "discovered."

```
Router# show mpls ldp discovery
 Local LDP Identifier:
    13.13.13.13:0
    Discovery Sources:
    Interfaces:
        GE-WAN3/3 (ldp): xmit/recv
            LDP Id: 12.12.12.12:0
    Targeted Hellos:
        13.13.13.13 -> 11.11.11.11 (ldp): active/passive, xmit/recv
            LDP Id: 11.11.11.11:0
```

- To make sure the label distribution session has been established, enter the **show mpls ldp neighbor** command. The third line of the output shows that the state of the LDP session is operational and shows that messages are being sent and received.

```
Router# show mpls ldp neighbor
    Peer LDP Ident: 12.12.12.12:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 12.12.12.12.646 - 13.13.13.13.11010
        State: Oper; Msgs sent/rcvd: 1715/1706; Downstream
        Up time: 1d00h
        LDP discovery sources:
          GE-WAN3/3, Src IP addr: 34.0.0.2
        Addresses bound to peer LDP Ident:
          23.2.1.14       37.0.0.2        12.12.12.12     34.0.0.2
          99.0.0.1
    Peer LDP Ident: 11.11.11.11:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 11.11.11.11.646 - 13.13.13.13.11013
        State: Oper; Msgs sent/rcvd: 1724/1730; Downstream
        Up time: 1d00h
        LDP discovery sources:
          Targeted Hello 13.13.13.13 -> 11.11.11.11, active, passive
        Addresses bound to peer LDP Ident:
          11.11.11.11     37.0.0.1        23.2.1.13
```

- To make sure the label forwarding table is built correctly, enter the **show mpls forwarding-table** command.

```
Router# show mpls forwarding-table
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Untagged    223.255.254.254/32   \
```

```
                                    0          Gi2/1      23.2.0.1
20     Untagged    l2ckt(2)         55146580   Vl2        point2point
24     Pop tag     37.0.0.0/8       0          GE3/3      34.0.0.2
25     17          11.11.11.11/32   0          GE3/3      34.0.0.2
26     Pop tag     12.12.12.12/32   0          GE3/3      34.0.0.2
```

- The output shows the following data:
  - Local tag—Label assigned by this router.
  - Outgoing tag or VC—Label assigned by next hop.
  - Prefix or Tunnel Id—Address or tunnel to which packets with this label are going.
  - Bytes tag switched— Number of bytes switched out with this incoming label.
  - Outgoing interface—Interface through which packets with this label are sent.
  - Next Hop—IP address of neighbor that assigned the outgoing label.
- To view the state of the currently routed VCs, enter the **show mpls l2transport vc** command:

```
Router# show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Vl2            Eth VLAN 2           11.11.11.11     2          UP
```

Configuring MUX-UNI Support on LAN Cards

A User Network Interface (UNI) is the point where the customer edge (CE) equipment connects to the ingress PE and an attachment VLAN is a VLAN on a UNI port.

The MUX-UNI support on LAN cards feature provides the ability to partition a physical port on an attachment VLAN to provide multiple Layer 2 and Layer 3 services over a single UNI.

When configuring MUX-UNI support on LAN cards, follow these guidelines and restrictions:

- Encapsulation on main interface has to be dot1Q and not ISL.
- With dot1q encapsulation on the main interface, you cannot configure ISL on the subinterfaces; Layer 3 interfaces are unaffected.

To configure MUX-UNI support on LAN cards, perform this task on the provider edge (PE) routers.

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface type number`	Selects an interface to configure and enters interface configuration mode; valid only for Ethernet ports.
Step 3	`Router(config-if)# switchport`	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 4	`Router(config-if)# switchport trunk encapsulation {isl | dot1q}`	Configures the port to support 802.1Q encapsulation. You must configure each end of the link with the same encapsulation type. **Note** The valid choice for MUX-UNI support is dot1Q.
Step 5	`Router(config-if)# switchport mode trunk`	Configures the port as a VLAN trunk.

Step 6	`Router(config-if)#` **`switchport trunk allowed vlan`** *`vlan-list`*	By default, all VLANs are allowed. Use this command to explicitly allow VLANs; valid values for *vlan-list* are from 1 to 4094. **Note** Avoid overlapping VLAN assignments between main and subinterfaces. VLAN assignments between the main interface and subinterfaces must be mutually exclusive.
Step 7	`Router(config-if)#` **`exit`**	Exits interface configuration mode.
Step 8	`Router(config)#` **`interface`** *`type slot/port.subinterface-number`*	Selects a subinterface to configure and enters interface configuration mode; valid only for Ethernet ports.
Step 9	`Router(config-if)#` **`encapsulation dot1q`** *`vlan_id`*	Enables the subinterface to accept 802.1Q VLAN packets. The subinterfaces between the CE and PE routers that are running Ethernet over MPLS must be in the same subnet. All other subinterfaces and backbone routers do not need to be on the same subnet.
Step 10	`Router(config-if)#` **`xconnect`** *`peer_router_id vcid`* **`encapsulation mpls`**	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

This example shows a physical trunk port used as UNI:

```
Router(config)# interface Fastethernet3/1
Router(config-if)# switchport
Router(config-if)# switchport encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed VLAN 200-250
Router(config-if)# exit

Router(config)# interface Fastethernet3/1.10
Router(config-if)# encap dot1q 3000
Router(config-if)# xconnect 10.0.0.1 3000 encapsulation mpls
Router(config-if)# exit
```

This example shows a Layer 2 port channel used as UNI:

```
Router(config)# interface Port-channel100
Router(config-if)# switchport
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport trunk allowed VLAN 100-200
Router(config-if)# switchport mode trunk
Router(config-if)# no ip address
Router(config-if)# exit

Router(config)# interface Port-channel100.1
Router(config-if)# encapsulation dot1Q 3100
Router(config-if)# xconnect 10.0.0.30 100 encapsulation mpls
Router(config-if)# exit
```

This example shows Layer 3 termination and VRF for muxed UNI ports:

```
Router(config)# vlan 200, 300, 400
Router(config)# interface Fastethernet3/1
Router(config-if)# switchport
```

```
Router(config-if)# switchport encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# switchport trunk allowed VLAN 200-500
Router(config-if)# exit

Router(config)# interface Fastethernet3/1.10
Router(config-if)# encap dot1q 3000
Router(config-if)# xconnect 10.0.0.1 3000 encapsulation mpls
Router(config-if)# exit

Router(config)# interface Vlan 200
Router(config-if)# ip address 1.1.1.3
Router(config-if)# exit

Router(config)# interface Vlan 300
Router(config-if)# ip vpn VRF A
Router(config-if)# ip address 3.3.3.1
Router(config-if)# exit

Router(config)# interface Vlan 400
Router(config-if)# ip address 4.4.4.1
Router(config-if)# ip ospf network broadcast
Router(config-if)# mpls label protocol ldp
Router(config-if)# mpls ip
Router(config-if)# exit
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring A-VPLS

This chapter describes how to configure Layer 2 Virtual Private Networks (L2VPN) Advanced Virtual Private LAN Services (A-VPLS). Release 12.2(33)SXI4 and later releases support A-VPLS.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

This chapter consists of these sections:

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Understanding A-VPLS

A-VPLS introduces the following enhancements to VPLS:

- Ability to load-balance traffic at the provider edge (PE) among multiple equal-cost core-facing paths and at core interfaces using flow labels.
- Support for redundant PE routers.

A-VPLS uses the Flow Aware Transport (FAT) Pseudowire feature to achieve PE redundancy and load-balancing on both PE and core routers. FAT pseudowires are used to load-balance traffic in the core when equal cost multipaths are used. The PE router adds an additional MPLS Label to the each packet (the flow label). Each flow has a unique flow label. For more information about FAT pseudowires, see PWE3 Internet-Draft *Flow Aware Transport of MPLS Pseudowires* (draft-bryant-filsfils-fat-pw).

Restrictions for A-VPLS

- Release 12.2(33)SXJ1 and later releases support configuration of ES+ module ports as MPLS PE core-facing ports that carry A-VPLS traffic.
- Release 12.2(33)SXI4a and later releases support A-VPLS on these core facing port types in a 7600-SIP-400:
  - Gigabit and 10-Gigabit Ethernet SPAs (2X1GE-V1, 2X1GE-V2 and 1X10GE-V2 SPA)
  - Packet over Sonet (POS) SPAs (2XOC3, 4XOC3, 1XOC12 and 1XOC48)
- Release 12.2(33)SXI4a and later releases support these types of configurations:
  - MPLS core with configuration of PE routers through the **neighbor** command under transport vpls mode.
  - MPLS core with configuration of PE routers through MPLS traffic engineering tunnels using explicit paths.
  - IP core with configuration of PE routers through MPLS over GRE tunnels.

  Other configuration methods, including use of the **route-via** command, BGP autodiscovery, or explicit VLAN assignment to a PE egress port, are not supported.
- A-VPLS supports the following:
  - In switches without an ES+ line card:

    —Up to 32 EtherChannel port-channel interfaces.
    (ES+ line cards do not support port-channel interfaces)

    —Up to 60 VPLS neighbors, minus the number of neighbors configured with the **load-balance flow** command.
  - In switches with an ES+ line card (with or without a 7600-SIP-400):

    —Up to 30 EtherChannel port-channel interfaces.
    (ES+ line cards do not support port-channel nterfaces)

    —Up to 30 VPLS neighbors, minus the number of neighbors configured with the **load-balance flow** command.
- A-VPLS requires nonstop forwarding and stateful switchover.
- A-VPLS works with following:
  - MPLS Traffic Engineering tunnels that are configured with explicit paths.
  - Generic Routing Encapsulation (GRE tunnels) that are configured with static routes to the tunnel destination.

  For information about MPLS traffic engineering and GRE tunnels, see the following documents:
  - *MPLS Traffic Engineering and Enhancements*
  - *Implementing Tunnels*
- The **ping** and **traceroute** commands that support the Any Transport over MPLS Virtual Circuit Connection Verification (VCCV) feature are not supported over FAT pseudowires.
- The VPLS Autodiscovery feature is not supported with A-VPLS.
- Load-balancing is not supported in the core routers when the core uses IP to transport packets.

Configuring A-VPLS

The following sections explain how to configure A-VPLS:

- Enabling Load-Balancing with ECMP and FAT Pseudowires, page 33-3 (Required)
- Enabling Port-Channel Load-Balancing, page 33-4 (Required)
- Explicitly Specifying the PE Routers As Part of Virtual Ethernet Interface Configuration, page 33-4 (Optional)
- Configuring an MPLS Traffic Engineering Tunnel, page 33-5 (Optional)
- Configuring a GRE Tunnel, page 33-6 (Optional)

Enabling Load-Balancing with ECMP and FAT Pseudowires

The following steps explain how to configure load-balancing on the provider edge (PE) routers, which enables it on the core P routers. No configuration is required on the core P routers.

To enable load-balancing on the edge routers, issue the **load-balance flow** command. The load-balancing rules are configured through the **port-channel load-balance** command parameters (see the "Enabling Port-Channel Load-Balancing" section on page 33-4).

To enable core load-balancing, issue the **flow-label enable** command on both PE routers. You must issue the **load-balance flow** command with the **flow-label enable** command.

	Command	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# pseudowire-class name`	Establishes a pseudowire class with a name that you specify and enters pseudowire class configuration mode.
Step 4	`Router(config-pw)# encapsulation mpls`	Specifies the MPLS tunneling encapsulation type.
Step 5	`Router(config-pw)# load-balance flow`	Enables load-balancing on ECMPs.
Step 6	`Router(config-pw)# flow-label enable`	Enables the imposition and disposition of flow labels for the pseudowire.
Step 7	`Router(config-pw)# end`	Exits pseudowire class configuration mode and enters privileged EXEC mode.

Enabling Port-Channel Load-Balancing

The following task explains how to enable port channel load-balancing, which sets the load-distribution method among the ports in the bundle. If the **port-channel load-balance** command is not configured, load-balancing occurs with default parameters.

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`port-channel load-balance`** *`method`*	Specifies the load distribution method among the ports in a bundle.
Step 4	`Router(config)#` **`exit`**	Exits global configuration mode and enters privileged EXEC mode.

Explicitly Specifying the PE Routers As Part of Virtual Ethernet Interface Configuration

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE routers as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to explicitly specify the PE routers as part of the virtual Ethernet interface configuration.

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`interface virtual-ethernet`** *`num`*	Creates a virtual Ethernet interface and enters interface configuration mode.
Step 4	`Router(config-if)#` **`transport vpls mesh`**	Create a full mesh of pseudowires and enters VPLS transport mode.
Step 5	`Router(config-if-transport)#` **`neighbor`** *`remote-router-id`* `[`**`pw-class`** `pw-class-name]`	Specifies the PE routers to be used in the pseudowire.
Step 6	`Router(config-if-transport)#` **`exit`**	Exits VPLS transport configuration mode and enters interface configuration mode.
Step 7	`Router(config-if)#` **`switchport`**	Configures the port for Layer 2 switching.
Step 8	`Router(config-if)#` **`switchport mode trunk`**	Enables permanent trunking mode and negotiates to convert the link into a trunk link.

	Command	Purpose
Step 9	Router(config-if)# **switchport trunk allowed vlan** {**add** | **except** | **none** | **remove**} *vlan* [,*vlan*[,*vlan*[,...]]	Configures the list of VLANs allowed on the trunk.
Step 10	Router(config)# **exit**	Exits interface configuration mode and enters privileged EXEC mode.

Configuring an MPLS Traffic Engineering Tunnel

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE routers as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to configure an MPLS Traffic Engineering tunnel. For more information about MPLS Traffic Engineering tunnels, see *MPLS Traffic Engineering and Enhancements.*

	Command	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **interface tunnel** *number*	Configures an interface type and enters interface configuration mode.
Step 4	Router(config-if)# **ip unnumbered** *type number*	Assigns an IP address to the tunnel interface. An MPLS traffic engineering tunnel interface should be unnumbered because it represents a unidirectional link.
Step 5	Router(config-if)# **tunnel destination** *ip-address*	Specifies the destination for a tunnel. The *ip-address* keyword is the IP address of the host destination expressed in dotted decimal notation.
Step 6	Router(config-if)# **tunnel mode mpls traffic-eng**	Configures the tunnel encapsulation mode to MPLS traffic engineering.
Step 7	Router(config-if)# **tunnel mpls traffic-eng autoroute announce**	Configures the IGP to use the tunnel in its enhanced SPF calculation.
Step 8	Router(config-if)# **tunnel mpls traffic-eng path-option** *number* {**dynamic** | **explicit** {**name** *path-name*} | **identifier** *path-number*} [**lockdown**]	Configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. A dynamic path is used if an explicit path is currently unavailable.
Step 9	Router(config-if)# **exit**	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a GRE Tunnel

There are several ways to specify the route through which traffic should pass.

- Explicitly specify the PE routers as part of the virtual Ethernet interface configuration
- Configure an MPLS Traffic Engineering tunnel
- Configure a GRE tunnel

The following task explains how to configure a GRE tunnel. For more information on GRE tunnels, see *Implementing Tunnels*.

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`interface`** *`type number`*	Specifies the interface type and number and enters interface configuration mode. To configure a tunnel, use **tunnel** for the type argument.
Step 4	`Router(config-if)#` **`tunnel mode`** `{`**`gre ip`** `|` **`gre multipoint`**`}`	Specifies the encapsulation protocol to be used in the tunnel.
Step 5	`Router(config-if)#` **`mpls ip`**	Enables MPLS on the tunnel.
Step 6	`outer(config-if)#` **`tunnel source`** `{`*`ip-address`* `|` *`interface-type interface-number`*`}`	Configures the tunnel source. • Use the *ip-address* argument to specify the source IP address. • Use the *interface-type* and *interface-number* arguments to specify the interface to use. **Note** The tunnel source and destination IP addresses must be defined on both PE routers.
Step 7	`Router(config-if)#` **`tunnel destination`** `{`*`hostname`* `|` *`ip-address`*`}`	Configures the tunnel destination. • Use the hostname argument to specify the name of the host destination. • Use the ip-address argument to specify the IP address of the host destination. **Note** The tunnel source and destination IP addresses must be defined on both PE routers.
Step 8	`Router(config-if)#` **`exit`**	Exits interface configuration mode and returns to privileged EXEC mode.
Step 9	`Router(config)#` **`ip route`** *`ip-address tunnel num`*	Creates a static route.

These examples show the three supported methods of configuring A-VPLS.

Explicitly Specifying Peer PE Routers

The following example shows how to create two VPLS domains under VLANs 10 and 20. Each VPLS domain includes two pseudowires to peer PE routers 10.2.2.2 and 10.3.3.3. Load-balancing is enabled through the l**oad-balance flow** and **flow-label enable** commands.

```
pseudowire-class cl1
   encap mpls
   load-balance flow
   flow-label enable
!
port-channel load-balance src-mac
!
interface virtual-ethernet 1
   transport vpls mesh
      neighbor 10.2.2.2 pw-class cl1
      neighbor 10.3.3.3 pw-class cl1
   switchport
   switchport mode trunk
   switchport trunk allowed vlan 10, 20
```

Using MPLS Traffic Engineering Tunnels

The following example shows the creation of two VPLS domains and uses MPLS Traffic Engineering tunnels to specify the explicit path.

```
pseudowire-class cl1
   encap mpls
   load-balance flow
   flow-label enable
!
port-channel load-balance src-mac
!
interface Tunnel1
   ip unnumbered Loopback0
   tunnel mode mpls traffic-eng
   tunnel destination 192.168.1.1
   tunnel mpls traffic-eng autoroute announce
   tunnel mpls traffic-eng path-option 1 explicit name LSP1
!
ip explicit-path name LSP1 enable
   next-address 192.168.2.2
   next-address loose 192.168.1.1
!
interface Tunnel2
   ip unnumbered Loopback0
   tunnel mode mpls traffic-eng
   tunnel destination 172.16.1.1
   tunnel mpls traffic-eng autoroute announce
   tunnel mpls traffic-eng path-option 1 explicit name LSP2
!
ip explicit-path name LSP2 enable
   next-address 172.16.2.2
   next-address loose 172.16.1.1
!
interface virtual-ethernet 1
    transport vpls mesh
      neighbor 10.2.2.2 pw-class cl1
      neighbor 10.3.3.3 pw-class cl1
   switchport
   switchport mode trunk
   switchport trunk allowed vlan 10,20
```

Using MPLS over GRE Tunnels

The following example shows the creation of two VPLS domains under VLANs 10 and 20. Each VPLS domain includes two pseudowires to peer PEs 10.2.2.2 and 10.3.3.3. The pseudowires are MPLS over GRE tunnels because the core is IP.

```
pseudowire-class cl1
   encap mpls
   load-balance flow
!
port-channel load-balance src-mac
!
interface tunnel 1
   tunnel mode gre ip
   mpls ip
   tunnel source 10.1.1.1
   tunnel destination 10.2.2.2
!
interface tunnel 2
   tunnel mode gre ip
   mpls ip
   tunnel source 10.1.1.1
   tunnel destination 10.3.3.3
!
interface virtual-ethernet 1
   transport vpls mesh
      neighbor 10.2.2.2 pw-class cl1
      neighbor 10.3.3.3 pw-class cl1
   switchport
   switchport mode trunk
   switchport trunk allowed vlan 10, 20

ip route 10.2.2.2 255.255.255.255 Tunnel1
ip route 10.3.3.3 255.255.255.255 Tunnel2
```

Routed Pseudo-Wire (RPW) and Routed VPLS

RPW and Routed VPLS can route Layer 3 traffic as well as switch Layer 2 frames for pseudowire connections between provider edge (PE) devices. Both point-to-point PE connections, in the form of Ethernet over MPLS (EoMPLS), and Virtual Private LAN Services (VPLS) multipoint PE connections are supported. The ability to route frames to and from these interfaces supports termination of a pseudowire into a Layer 3 network (VPN or global) on the same switch, or to tunnel Layer 3 frames over a Layer 2 tunnel (EoMPLS or VPLS). The feature supports faster network convergence in the event of a physical interface or device failure through the MPLS Traffic Engineering (MPLS-TE) and Fast Reroute (FRR) features. In particular, the feature enables MPLS TE-FRR protection for Layer 3 multicast over a VPLS domain.

Note When the RPW is configured in A-VPLS mode, TE/FRR is not supported because A-VPLS runs over ECMP and the ECMP convergence is comparable to TE/FRR.

To configure routing support for the pseudowire, configure an IP address and other Layer 3 features for the Layer 3 domain (VPN or global) in the virtual LAN (VLAN) interface configuration. The following example assigns the IP address 10.10.10.1 to the VLAN 100 interface, and enables Multicast PIM. (Layer 2 forwarding is defined by the VFI VFI100.)

```
interface vlan 100
  xconnect vfi VFI100
  ip address 10.10.10.1 255.255.255.0
  ip pim sparse-mode
```

The following example assigns an IP address 20.20.20.1 of the VPN domain VFI200. (Layer 2 forwarding is defined by the VFI VFI200.)

```
interface vlan 200
  xconnect vfi VFI200
  ip vrf forwarding VFI200
  ip address 20.20.20.1 255.255.255.0
```

PART 5

IP Switching

Configuring IP Unicast Layer 3 Switching

This chapter describes how to configure IP unicast Layer 3 switching in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Note

- IPX traffic is fast switched on the route proccessor (RP). For more information, see this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/atipx/configuration/guide/fatipx_c.html

- For information about IP multicast Layer 3 switching, see Chapter 37, "Configuring IPv4 Multicast Layer 3 Switching."

Understanding Layer 3 Switching

These sections describe Layer 3 switching:

- Understanding Hardware Layer 3 Switching, page 34-2
- Understanding Layer 3-Switched Packet Rewrite, page 34-2

Understanding Hardware Layer 3 Switching

Hardware Layer 3 switching allows the PFC and DFCs, instead of the RP, to forward IP unicast traffic between subnets. Hardware Layer 3 switching provides wire-speed forwarding on the PFC and DFCs, instead of in software on the RP. Hardware Layer 3 switching requires minimal support from the RP. The RP routes any traffic that cannot be hardware Layer 3 switched.

Hardware Layer 3 switching supports the routing protocols configured on the RP. Hardware Layer 3 switching does not replace the routing protocols configured on the RP.

Hardware Layer 3 switching runs equally on the PF3 and DFCs to provide IP unicast Layer 3 switching locally on each module. Hardware Layer 3 switching provides the following functions:

- Hardware access control list (ACL) switching for policy-based routing (PBR)
- Hardware flow-based switching for TCP intercept and reflexive ACL forwarding decisions
- Hardware Cisco Express Forwarding (CEF) switching for all other IP unicast traffic

Hardware Layer 3 switching on the PFC supports modules that do not have a DFC. The RP forwards traffic that cannot be Layer 3 switched.

Traffic is hardware Layer 3 switched after being processed by access lists and quality of service (QoS).

Hardware Layer 3 switching makes a forwarding decision locally on the ingress-port module for each packet and sends the rewrite information for each packet to the egress port, where the rewrite occurs when the packet is transmitted from the switch.

Hardware Layer 3 switching generates flow statistics for Layer 3-switched traffic. Hardware Layer 3 flow statistics can be used for NetFlow Data Export (NDE). (See Chapter 64, "Configuring NDE".)

Understanding Layer 3-Switched Packet Rewrite

When a packet is Layer 3 switched from a source in one subnet to a destination in another subnet, the switch performs a packet rewrite at the egress port based on information learned from the RP so that the packets appear to have been routed by the RP.

Packet rewrite alters five fields:

- Layer 2 (MAC) destination address
- Layer 2 (MAC) source address
- Layer 3 IP Time to Live (TTL)
- Layer 3 checksum
- Layer 2 (MAC) checksum (also called the frame checksum or FCS)

Note Packets are rewritten with the encapsulation appropriate for the next-hop subnet.

If Source A and Destination B are in different subnets and Source A sends a packet to the RP to be routed to Destination B, the switch recognizes that the packet was sent to the Layer 2 (MAC) address of the RP.

To perform Layer 3 switching, the switch rewrites the Layer 2 frame header, changing the Layer 2 destination address to the Layer 2 address of Destination B and the Layer 2 source address to the Layer 2 address of the RP. The Layer 3 addresses remain the same.

In IP unicast and IP multicast traffic, the switch decrements the Layer 3 TTL value by 1 and recomputes the Layer 3 packet checksum. The switch recomputes the Layer 2 frame checksum and forwards (or, for multicast packets, replicates as necessary) the rewritten packet to Destination B's subnet.

A received IP unicast packet is formatted (conceptually) as follows:

Layer 2 Frame Header		**Layer 3 IP Header**				**Data**	**FCS**
Destination	Source	Destination	Source	TTL	Checksum		
RP MAC	*Source A MAC*	*Destination B IP*	*Source A IP*	*n*	*calculation1*		

After the switch rewrites an IP unicast packet, it is formatted (conceptually) as follows:

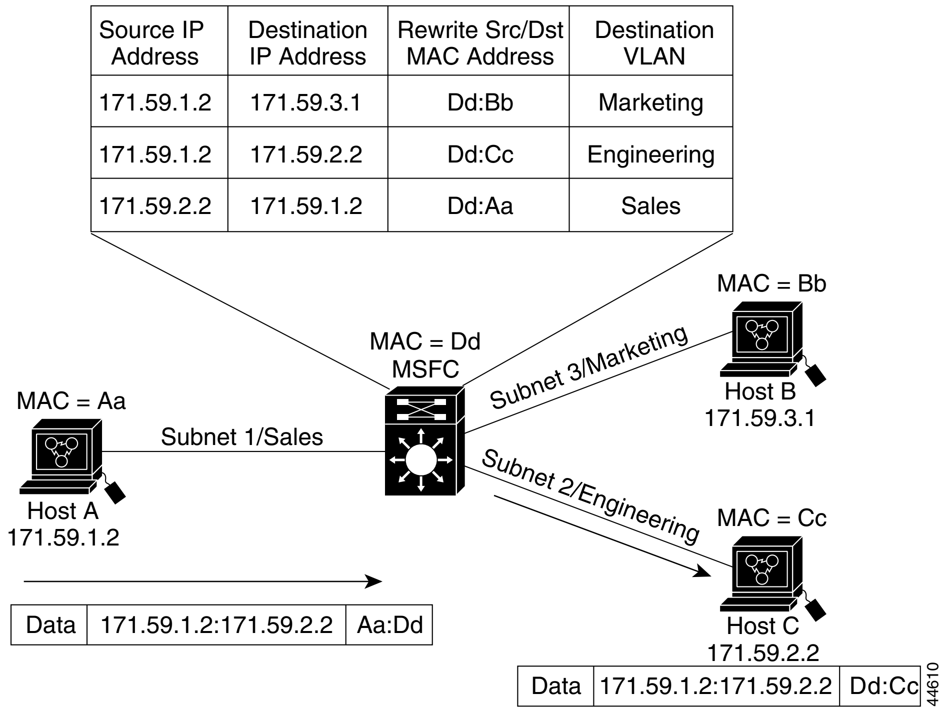
Layer 2 Frame Header		**Layer 3 IP Header**				**Data**	**FCS**
Destination	Source	Destination	Source	TTL	Checksum		
Destination B MAC	*RP MAC*	*Destination B IP*	*Source A IP*	*n-1*	*calculation2*		

Hardware Layer 3 Switching Examples

Figure 34-1 on page 34-4 shows a simple network topology. In this example, Host A is on the Sales VLAN (IP subnet 171.59.1.0), Host B is on the Marketing VLAN (IP subnet 171.59.3.0), and Host C is on the Engineering VLAN (IP subnet 171.59.2.0).

When Host A initiates an HTTP file transfer to Host C, Hardware Layer 3 switching uses the information in the local forwarding information base (FIB) and adjacency table to forward packets from Host A to Host C.

Figure 34-1 Hardware Layer 3 Switching Example Topology



Default Hardware Layer 3 Switching Configuration

Table 34-1 shows the default hardware Layer 3 switching configuration.

Table 34-1 Default Hardware Layer 3 Switching Configuration

Feature	Default Value
Hardware Layer 3 switching enable state	Enabled (cannot be disabled)
Cisco IOS CEF enable state on RP	Enabled (cannot be disabled)
Cisco IOS dCEF[1] enable state on RP	Enabled (cannot be disabled)

1. dCEF = Distributed Cisco Express Forwarding

Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring hardware Layer 3 switching:

- Hardware Layer 3 switching supports the following ingress and egress encapsulations:
  - Ethernet V2.0 (ARPA)
  - 802.3 with 802.2 with 1 byte control (SAP1)

Configuring Hardware Layer 3 Switching

> **Note** For information on configuring unicast routing on the RP, see Chapter 30, "Configuring Layer 3 Interfaces."

Hardware Layer 3 switching is permanently enabled. No configuration is required.

To display information about Layer 3-switched traffic, perform this task:

Command	Purpose
Router# **show interface** {{*type*[1] *slot/port*} | {**port-channel** *number*}} **| begin L3**	Displays a summary of Layer 3-switched traffic.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to display information about hardware Layer 3-switched traffic on Fast Ethernet port 3/3:

```
Router# show interface fastethernet 3/3 | begin L3
  L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 12 pkt, 778 bytes mcast
  L3 out Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
     4046399 packets input, 349370039 bytes, 0 no buffer
     Received 3795255 broadcasts, 2 runts, 0 giants, 0 throttles
<...output truncated...>
Router#
```

> **Note** The Layer 3 switching packet count is updated approximately every five seconds.

Cisco IOS CEF and dCEF are permanently enabled. No configuration is required to support hardware Layer 3 switching.

With a PFC (and DFCs, if present), hardware Layer 3 switching uses per-flow load balancing based on IP source and destination addresses. Per-flow load balancing avoids the packet reordering that can be necessary with per-packet load balancing. For any given flow, all PFC- and DFC-equipped switches make exactly the same load-balancing decision, which can result in nonrandom load balancing.

The Cisco IOS CEF **ip load-sharing per-packet**, **ip cef accounting per-prefix**, and **ip cef accounting non-recursive** commands on the RP apply only to traffic that is CEF-switched in software on the RP. The commands do not affect traffic that is hardware Layer 3 switched on the PFC or on DFC-equipped switching modules.

For information about Cisco IOS CEF and dCEF on the RP, see these publications:

- The "Cisco Express Forwarding" sections at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcfcef.html

- The *Cisco IOS Switching Services Command Reference* publication at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/switch/command/reference/fswtch_r.html

Displaying Hardware Layer 3 Switching Statistics

Hardware Layer 3 switching statistics are obtained on a per-VLAN basis.

To display hardware Layer 3 switching statistics, perform this task:

Command	Purpose
Router# **show interfaces** {{*type*[1] *slot/port*} | {**port-channel** *number*}}	Displays hardware Layer 3 switching statistics.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to display hardware Layer 3 switching statistics:

```
Router# show interfaces gigabitethernet 9/5 | include Switched
L2 Switched: ucast: 8199 pkt, 1362060 bytes - mcast: 6980 pkt, 371952 bytes
L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes mcast
L3 out Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
```

To display adjacency table information, perform this task:

Command	Purpose
Router# **show adjacency** [{{*type*[1] *slot/port*} | {**port-channel** *number*}} | **detail** | **internal** | **summary**]	Displays adjacency table information. The optional **detail** keyword displays detailed adjacency information, including Layer 2 information.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to display adjacency statistics:

```
Router# show adjacency gigabitethernet 9/5 detail
Protocol Interface                 Address
IP       GigabitEthernet9/5        172.20.53.206(11)
                                   504 packets, 6110 bytes
                                   00605C865B82
                                   000164F83FA50800
                                   ARP        03:49:31
```

Note Adjacency statistics are updated approximately every 60 seconds.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 6

IPv6

Configuring IPv6 Multicast PFC3 and DFC3 Layer 3 Switching

The PFC3 and DFC3 provide hardware support for IPv6 multicast traffic. Use these publications to configure IPv6 multicast in Cisco IOS Release 12.2SX:

- The *Cisco IOS IPv6 Configuration Library*, “Implementing IPv6 Multicast”:

  http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/configuration/12-2sy/ipv6-12-2sy-book.html

- The *Cisco IOS IPv6 Command Reference*:

  http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

These sections provide additional information about IPv6 multicast support in Cisco IOS Release 12.2SX:

Features that Support IPv6 Multicast

These features support IPv6 multicast:

- RPR redundancy mode—See Chapter 7, "Configuring RPR Supervisor Engine Redundancy."
- Multicast Listener Discovery version 2 (MLDv2) snooping—See Chapter 36, "Configuring MLD Snooping for IPv6 Multicast Traffic."

Note MLDv1 snooping is not supported.

- IPv6 Multicast rate limiters—See Chapter 52, "Configuring Denial of Service Protection."
- IPv6 Multicast: Bootstrap Router (BSR)—See the BSR information in the *Cisco IOS IPv6 Configuration Library* and the *Cisco IOS IPv6 Command Reference*.
- IPv6 Access Services—See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/config_library/15-sy/ipv6-15-sy-library.html
- SSM mapping for IPv6—See this publication for more information:

  http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/config_library/15-sy/ipv6-15-sy-library.html

IPv6 Multicast Guidelines and Restrictions

These guidelines and restrictions apply to IPv6 multicast support in Cisco IOS Release 12.2SX:

- The PFC3 and DFC3 provide hardware support for the following:
  - Completely switched IPv6 multicast flows
  - IPv6 PIM-Sparse Mode (PIM-SM) (S,G) and (*,G) forwarding

Note Release 12.2(33)SXH provides initial support for IPv6 PIM-SM (*,G) forwarding.

  - Multicast RPF check for IPv6 PIM-SM (S,G) traffic using the NetFlow table
  - Rate limiting of IPv6 PIM-SM (S,G) traffic that fails the multicast RPF check
  - Static IPv6 multicast routes
  - SSM Mapping for IPv6 (PIM-SSM)
  - IPv6 multicast forwarding information base (MFIB) using the NetFlow table
  - IPv6 distributed MFIB (dMFIB) using the NetFlow table
  - Link-local and link-global IPv6 multicast scopes
  - Egress multicast replication with the **ipv6 mfib hardware-switching** command
  - Ingress interface statistics for multicast routes (egress interface statistics not available)
  - RPR redundancy mode (see Chapter 7, "Configuring RPR Supervisor Engine Redundancy")
  - Ingress and egress PFC QoS (see Chapter 43, "Configuring PFC QoS")
  - Input and output Cisco access-control lists (ACLs)

- The PFC3 and DFC3 do not provide hardware support for the following:
  - Partially switched IPv6 multicast flows
  - Multicast RPF check for PIM-SM (*,G) traffic
  - Multicast helper maps
  - Site-local multicast scopes
  - Manually configured IPv6 over IPv4 tunnels
  - IPv6 multicast 6to4 tunnels
  - IPv6 multicast automatic tunnels
  - IPv6 over GRE tunnels
  - IPv6-in-IPv6 PIM register tunnels
  - IPv6 multicast basic ISATAP tunnels
  - ISATAP tunnels with embedded 6to4 tunnels

New or Changed IPv6 Multicast Commands

See the Cisco IOS Master Command List for information about these IPv6 multicast commands:

- **ipv6 mfib hardware-switching**
- **mls rate-limit multicast ipv6** (see Chapter 52, "Configuring Denial of Service Protection")
- **show ipv6 mfib**
- **show mls rate-limit** (see Chapter 52, "Configuring Denial of Service Protection")
- **show platform software ipv6-multicast**
- **show tcam interface**

Configuring IPv6 Multicast Layer 3 Switching

To configure IPv6 multicast Layer 3 switching, perform this task:

	Command	Purpose
Step 1	`Router(config)# ipv6 unicast-routing`	Enables unicast routing on all Layer 3 interfaces.
Step 2	`Router(config)# ipv6 multicast-routing`	Enables PIM-SM on all Layer 3 interfaces.
Step 3	`Router(config)# ipv6 mfib hardware-switching`	Enables MFIB hardware switching globally.

Using show Commands to Verify IPv6 Multicast Layer 3 Switching

These sections describe how to use **show** commands to verify IPv6 multicast Layer 3 switching:

- Verifying MFIB Clients, page 35-4

Note The **show** commands in the following sections are for a switch with a DFC3-equipped switching module in slot 1 and a Supervisor Engine 720 with a PFC3 in slot 6.

Verifying MFIB Clients

This example shows the complete output of the **show ipv6 mrib client** command:

```
Router# show ipv6 mrib client
IP MRIB client-connections
mfib ipv6:81     (connection id 0)
igmp:124         (connection id 1)
pim:281 (connection id 2)
slot 1  mfib ipv6 rp agent:15   (connection id 3)
slot 6  mfib ipv6 rp agent:15   (connection id 4)
```

This example shows how to display the MFIB client running on the route processor (RP):

```
Router# show ipv6 mrib client | include ^mfib ipv6
mfib ipv6:81     (connection id 0)
```

This example shows how to display the MFIB clients running on the PFC3 and any DFC3s:

```
Router# show ipv6 mrib client | include slot
slot 1  mfib ipv6 rp agent:15   (connection id 3)
slot 6  mfib ipv6 rp agent:15   (connection id 4)
```

Displaying the Switching Capability

This example displays the complete output of the **show platform software ipv6-multicast capability** command:

```
Router# show platform software ipv6-multicast capability

 Hardware switching for IPv6 is enabled
 (S,G) forwarding for IPv6 supported using Netflow
 (*,G) bridging for IPv6 is supported using FIB
 Directly-connected entries for IPv6 is supported using ACL-TCAM.
```

```
Current System HW Replication Mode : Ingress
Auto-detection of Replication Mode : ON

Slot Replication-Capability Replication-Mode
   1 Ingress                Ingress
   2 Egress                 Ingress
   6 Egress                 Ingress
   8 Ingress                Ingress
```

Verifying the (S,G) Forwarding Capability

This example shows how to verify the (S,G) forwarding:

```
Router# show platform software ipv6-multicast capability | include (S,G)
 (S,G) forwarding for IPv6 supported using Netflow
```

Verifying the (*,G) Forwarding Capability

This example shows how to verify the (*,G) forwarding:

```
Router# show platform software ipv6-multicast capability | include (\*,G)
 (*,G) bridging for IPv6 is supported using FIB
```

Verifying the Subnet Entry Support Status

This example shows how to verify the subnet entry support status:

```
Router# show platform software ipv6-multicast capability | include entries
 Directly-connected entries for IPv6 is supported using ACL-TCAM.
```

Verifying the Current Replication Mode

This example shows how to verify the current replication mode:

```
Router# show platform software ipv6-multicast capability | include Current
Current System HW Replication Mode : Ingress
```

Note Enter the **no ipv6 mfib hardware-switching replication-mode ingress** command to enable replication mode auto-detection.

Displaying the Replication Mode Auto-Detection Status

This example shows how to display the replication mode auto-detection status:

```
Router# show platform software ipv6-multicast capability | include detection
Auto-detection of Replication Mode : ON
```

Displaying the Replication Mode Capabilities

This example shows how to display the replication mode capabilities of the installed modules:

```
Router# show platform software ipv6-multicast capability | begin ^Slot
Slot Replication-Capability Replication-Mode
   1 Ingress                Ingress
   2 Egress                 Ingress
   6 Egress                 Ingress
   8 Ingress                Ingress
```

Displaying Subnet Entries

This example shows how to display subnet entries:

```
Router# show platform software ipv6-multicast connected
IPv6 Multicast Subnet entries
Flags : H - Installed in ACL-TCAM
        X - Not installed in ACL-TCAM due to
            label-full exception
Interface: Vlan20 [ H ]
         S:20::1 G:FF00::
Interface: Vlan10 [ H ]
         S:10::1 G:FF00::
```

Note In this example, there are subnet entries for VLAN 10 and VLAN 20.

Displaying the IPv6 Multicast Summary

This example shows how to display the IPv6 multicast summary:

```
Router# show platform software ipv6-multicast summary
IPv6 Multicast Netflow SC summary on Slot[1]:
Shortcut Type               Shortcut count
--------------------------+--------------
(S, G)                      100
(*, G)                      0
IPv6 Multicast FIB SC summary on Slot[1]:
Shortcut Type               Shortcut count
--------------------------+--------------
(*, G/128)                  10
(*, G/m)                    47

IPv6 Multicast Netflow SC summary on Slot[6]:
Shortcut Type               Shortcut count
--------------------------+--------------
(S, G)                      100
(*, G)                      0
IPv6 Multicast FIB SC summary on Slot[6]:
Shortcut Type               Shortcut count
--------------------------+--------------
(*, G/128)                  10
(*, G/m)                    47
```

Displaying the NetFlow Hardware Forwarding Count

This example shows how to display the NetFlow hardware forwarding count:

```
Router# show platform software ipv6-multicast summary
IPv6 Multicast Netflow SC summary on Slot[1]:
Shortcut Type              Shortcut count
---------------------------+--------------
(S, G)                     100
(*, G)                     0

<...Output deleted...>

IPv6 Multicast Netflow SC summary on Slot[6]:
Shortcut Type              Shortcut count
---------------------------+--------------
(S, G)                     100
(*, G)                     0
<...Output truncated...>
```

Note The NetFlow (*, G) count is always zero because PIM-SM (*,G) forwarding is supported in software on the RP.

Displaying the FIB Hardware Bridging and Drop Counts

This example shows how to display the FIB hardware bridging and drop hardware counts:

```
Router# show platform software ipv6-multicast summary | begin FIB
IPv6 Multicast FIB SC summary on Slot[1]:
Shortcut Type              Shortcut count
---------------------------+--------------
(*, G/128)                 10
(*, G/m)                   47

<...Output deleted...>

IPv6 Multicast FIB SC summary on Slot[6]:
Shortcut Type              Shortcut count
---------------------------+--------------
(*, G/128)                 10
(*, G/m)                   47
```

Note
- The (*, G/128) value is a hardware bridge entry count.
- The (*, G/m) value is a hardware bridge/drop entry count.

Displaying the Shared and Well-Known Hardware Adjacency Counters

The **show platform software ipv6-multicast shared-adjacencies** command displays the shared and well-known hardware adjacency counters used for IPv6 multicast by entries in FIB and ACL-TCAM.

```
Router# show platform software ipv6-multicast shared-adjacencies
```

```
---- SLOT [1] ----

Shared IPv6 Mcast Adjacencies Index  Packets       Bytes
----------------------------- ------ ------------- ------------------
Subnet bridge adjacency       0x7F802  0             0
Control bridge adjacency      0x7      0             0
StarG_M bridge adjacency      0x8      0             0
S_G bridge adjacency          0x9      0             0
Default drop adjacency        0xA      0             0
StarG (spt == INF) adjacency  0xB      0             0
StarG (spt != INF) adjacency  0xC      0             0

---- SLOT [6] ----

Shared IPv6 Mcast Adjacencies Index  Packets       Bytes
----------------------------- ------ ------------- ------------------
Subnet bridge adjacency       0x7F802  0             0
Control bridge adjacency      0x7      0             0
StarG_M bridge adjacency      0x8      0             0
S_G bridge adjacency          0x9      0             0
Default drop adjacency        0xA      28237         3146058
StarG (spt == INF) adjacency  0xB      0             0
StarG (spt != INF) adjacency  0xC      0             0
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring MLD Snooping for IPv6 Multicast Traffic

This chapter describes how to configure Multicast Listener Discovery (MLD) snooping for IPv6 multicast traffic in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- To constrain IPv4 multicast traffic, see Chapter 38, "Configuring IGMP Snooping for IPv4 Multicast Traffic."
- PFC3C and PFC3CXL modes support MLD version 1 (MLDv1) and MLD version 2 (MLDv2).
- These modes support only MLD version 2 (MLDv2):
  - PFC3A
  - PFC3B
  - PFC3BXL

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding MLD Snooping

These sections describe MLD snooping:

MLD Snooping Overview

MLD snooping allows the switch to examine MLD packets and make forwarding decisions based on their content.

You can configure the switch to use MLD snooping in subnets that receive MLD queries from either MLD or the MLD snooping querier. MLD snooping constrains IPv6 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv6 multicast traffic only to those ports that want to receive it.

MLD, which runs at Layer 3 on a multicast router, generates Layer 3 MLD queries in subnets where the multicast traffic needs to be routed. For information about MLD, see this publication:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/configuration/12-2sx/ipv6-12-2sx-book.html

You can configure the MLD snooping querier on the switch to support MLD snooping in subnets that do not have any multicast router interfaces. For more information about the MLD snooping querier, see the "Enabling the MLD Snooping Querier" section on page 36-9.

MLD (on a multicast router) or, locally, the MLD snooping querier, sends out periodic general MLD queries that the switch forwards through all ports in the VLAN, and to which hosts respond. MLD snooping monitors the Layer 3 MLD traffic.

Note PFC/DFC 3C/3CXL supports source-only Layer 2 entries, but PFC/DFC 3B/3BXL does not support source-only Layer 2 entries and therefore IPv6 multicast flooding cannot be prevented in a source-only network.

Note If a multicast group has only sources and no receivers in a VLAN, MLD snooping constrains the multicast traffic to only the multicast router ports.

MLD Messages

These are the MLD messages:

- Multicast listener queries
  - General query—Sent by a multicast router to learn which multicast addresses have listeners.
  - Multicast address specific query—Sent by a multicast router to learn if a particular multicast address has any listeners.
  - Multicast address and source specific query—Sent by a multicast router to learn if any of the sources from the specified list for the particular multicast address has any listeners.
- Multicast listener reports
  - Current state record (solicited)—Sent by a host in response to a query to specify the INCLUDE or EXCLUDE mode for every multicast group in which the host is interested.
  - Filter mode change record (unsolicited)—Sent by a host to change the INCLUDE or EXCLUDE mode of one or more multicast groups.
  - Source list change record (unsolicited)—Sent by a host to change information about multicast sources.

Source-Based Filtering

MLD uses source-based filtering, which enables hosts and routers to specify which multicast sources should be allowed or blocked for a specific multicast group. Source-based filtering either allows or blocks traffic based on the following information in MLD messages:

- Source lists
- INCLUDE or EXCLUDE mode

Because the Layer 2 table is (MAC-group, VLAN) based, with MLD hosts it is preferable to have only a single multicast source per MAC-group.

Note Source-based filtering is not supported in hardware. The states are maintained only in software and used for explicit host tracking and statistics collection.

Explicit Host Tracking

MLD supports explicit tracking of membership information on any port. The explicit-tracking database is used for fast-leave processing, proxy reporting, and statistics collection. When explicit tracking is enabled on a VLAN, the MLD snooping software processes the MLD report it receives from a host and builds an explicit-tracking database that contains the following information:

- The port connected to the host
- The channels reported by the host
- The filter mode for each group reported by the host
- The list of sources for each group reported by the hosts
- The router filter mode of each group
- For each group, the list of hosts requesting the source

Note

- Disabling explicit host tracking disables fast-leave processing and proxy reporting.
- When explicit tracking is enabled and the switch is in report-suppression mode, the multicast router might not be able to track all the hosts accessed through a VLAN interface.

MLD Snooping Proxy Reporting

Because MLD does not have report suppression, all the hosts send their complete multicast group membership information to the multicast router in response to queries. The switch snoops these responses, updates the database and forwards the reports to the multicast router. To prevent the multicast router from becoming overloaded with reports, MLD snooping does proxy reporting.

Proxy reporting forwards only the first report for a multicast group to the router and suppresses all other reports for the same multicast group.

Proxy reporting processes solicited and unsolicited reports. Proxy reporting is enabled and cannot be disabled.

Note Disabling explicit host tracking disables fast-leave processing and proxy reporting.

Joining an IPv6 Multicast Group

Hosts join IPv6 multicast groups either by sending an unsolicited MLD report or by sending an MLD report in response to a general query from an IPv6 multicast router (the switch forwards general queries from IPv6 multicast routers to all ports in a VLAN). The switch snoops these reports.

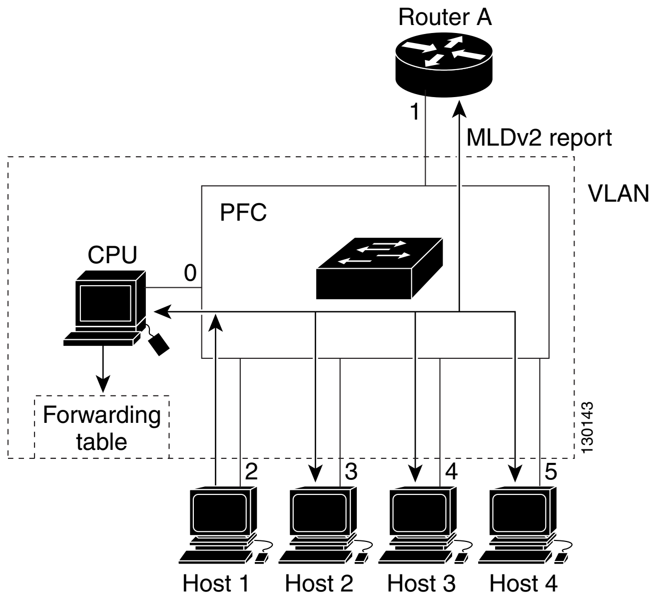
In response to a snooped MLD report, the switch creates an entry in its Layer 2 forwarding table for the VLAN on which the report was received. When other hosts that are interested in this multicast traffic send MLD reports, the switch snoops their reports and adds them to the existing Layer 2 forwarding table entry. The switch creates only one entry per VLAN in the Layer 2 forwarding table for each multicast group for which it snoops an MLD report.

MLD snooping suppresses all but one of the host reports per multicast group and forwards this one report to the IPv6 multicast router.

The switch forwards multicast traffic for the multicast group specified in the report to the interfaces where reports were received (see Figure 36-1).

Layer 2 multicast groups learned through MLD snooping are dynamic. However, you can statically configure Layer 2 multicast groups using the **mac-address-table static** command. When you specify group membership for a multicast group address statically, the static setting supersedes any MLD snooping learning. Multicast group membership lists can consist of both static and MLD snooping-learned settings.

Figure 36-1 Initial MLD Listener Report



Multicast router A sends an MLD general query to the switch, which forwards the query to ports 2 through 5 (all members of the same VLAN). Host 1 wants to join an IPv6 multicast group and multicasts an MLD report to the group with the equivalent MAC destination address of 0x0100.5E01.0203. When the switch snoops the MLD report multicast by Host 1, the switch uses the information in the MLD report to create a forwarding-table entry. Table 36-1 shows the forwarding table, which includes the port numbers of Host 1, the multicast router, and the switch.

Table 36-1 MLD Snooping Forwarding Table

Destination MAC Address	Type of Packet	Ports
0100.5exx.xxxx	MLD	0
0100.5e01.0203	!MLD	1, 2

The switch hardware can distinguish MLD information packets from other packets for the multicast group. The first entry in the table indicates that only MLD packets should be sent to the CPU, which prevents the switch from becoming overloaded with multicast frames. The second entry indicates that frames addressed to the 0x0100.5E01.0203 multicast MAC address that are not MLD packets (!MLD) should be sent to the multicast router and to the host that has joined the group.

If another host (for example, Host 4) sends an unsolicited MLD report for the same group (Figure 36-2), the switch snoops that message and adds the port number of Host 4 to the forwarding table as shown in Table 36-2. Because the forwarding table directs MLD messages only to the switch, the message is not flooded to other ports. Any known multicast traffic is forwarded to the group and not to the switch.

Figure 36-2 Second Host Joining a Multicast Group

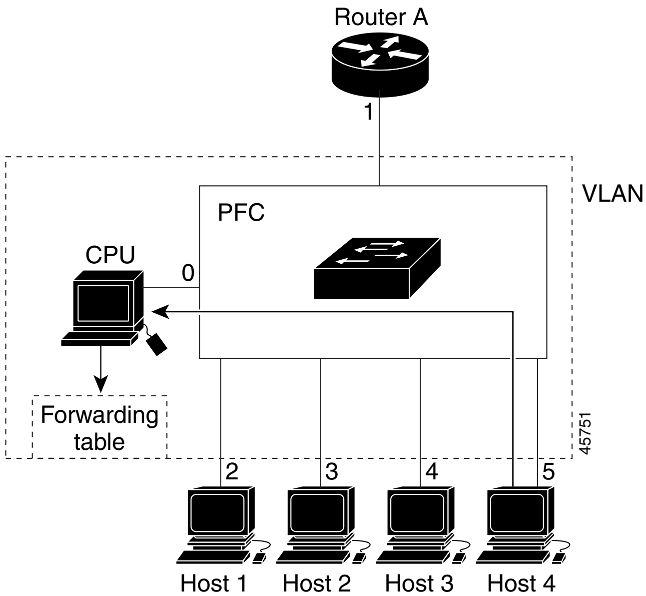


Table 36-2 Updated MLD Snooping Forwarding Table

Destination MAC Address	Type of Packet	Ports
0100.5exx.xxxx	MLD	0
0100.5e01.0203	!MLD	1, 2, 5

Leaving a Multicast Group

These sections describe leaving a multicast group:

- Normal Leave Processing, page 36-6
- Fast-Leave Processing, page 36-7

Normal Leave Processing

Interested hosts must continue to respond to the periodic MLD general queries. As long as at least one host in the VLAN responds to the periodic MLD general queries, the multicast router continues forwarding the multicast traffic to the VLAN. When hosts want to leave a multicast group, they can either ignore the periodic MLD general queries (called a “silent leave”), or they can send an MLD filter mode change record.

When MLD snooping receives a filter mode change record from a host that configures the EXCLUDE mode for a group, MLD snooping sends out a MAC-addressed general query to determine if any other hosts connected to that interface are interested in traffic for the specified multicast group.

If MLD snooping does not receive an MLD report in response to the general query, MLD snooping assumes that no other hosts connected to the interface are interested in receiving traffic for the specified multicast group, and MLD snooping removes the interface from its Layer 2 forwarding table entry for the specified multicast group.

If the filter mode change record was from the only remaining interface with hosts interested in the group, and MLD snooping does not receive an MLD report in response to the general query, MLD snooping removes the group entry and relays the MLD filter mode change record to the multicast router. If the multicast router receives no reports from a VLAN, the multicast router removes the group for the VLAN from its MLD cache.

The interval for which the switch waits before updating the table entry is called the "last member query interval." To configure the interval, enter the **ipv6 mld snooping last-member-query-interval** *interval* command.

Fast-Leave Processing

Fast-leave processing is enabled by default. To disable fast-leave processing, turn off explicit-host tracking.

Fast-leave processing is implemented by maintaining source-group based membership information in software while also allocating LTL indexes on a MAC GDA basis.

When fast-leave processing is enabled, hosts send BLOCK_OLD_SOURCES{src-list} messages for a specific group when they no longer want to receive traffic from that source. When the switch receives such a message from a host, it parses the list of sources for that host for the given group. If this source list is exactly the same as the source list received in the leave message, the switch removes the host from the LTL index and stops forwarding this multicast group traffic to this host.

If the source lists do not match, the switch does not remove the host from the LTL index until the host is no longer interested in receiving traffic from any source.

Note Disabling explicit host tracking disables fast-leave processing and proxy reporting.

Understanding the MLD Snooping Querier

Use the MLD snooping querier to support MLD snooping in a VLAN where PIM and MLD are not configured because the multicast traffic does not need to be routed.

In a network where IP multicast routing is configured, the IP multicast router acts as the MLD querier. If the IP-multicast traffic in a VLAN only needs to be Layer 2 switched, an IP-multicast router is not required, but without an IP-multicast router on the VLAN, you must configure another switch as the MLD querier so that it can send queries.

When enabled, the MLD snooping querier sends out periodic MLD queries that trigger MLD report messages from the switch that wants to receive IP multicast traffic. MLD snooping listens to these MLD reports to establish appropriate forwarding.

You can enable the MLD snooping querier on all the switches in the VLAN, but for each VLAN that is connected to switches that use MLD to report interest in IP multicast traffic, you must configure at least one switch as the MLD snooping querier.

You can configure a switch to generate MLD queries on a VLAN regardless of whether or not IP multicast routing is enabled.

Default MLD Snooping Configuration

Table 36-3 shows the default MLD snooping configuration.

Table 36-3 MLD Snooping Default Configuration

Feature	Default Values
MLD snooping querier	Disabled
MLD snooping	Enabled
Multicast routers	None configured
MLD report suppression	Enabled
MLD snooping router learning method	Learned automatically through PIM or MLD packets
Fast-Leave Processing	Enabled
MLD Explicit Host Tracking	Enabled

MLD Snooping Configuration Guidelines and Restrictions

When configuring MLD snooping, follow these guidelines and restrictions:

- Only PFC3C and PFC3CXL modes support MLD version 1 (MLDv1) and MLD version 2 (MLDv2).
- These modes support only MLD version 2 (MLDv2):
  - PFC3A
  - PFC3B
  - PFC3BXL
- MLD is derived from Internet Group Management Protocol version 3 (IGMPv3). MLD protocol operations and state transitions, host and router behavior, query and report message processing, message forwarding rules, and timer operations are exactly same as IGMPv3. See draft-vida-mld-.02.txt for detailed information on MLD protocol.
- MLD protocol messages are Internet Control Message Protocol version 6 (ICMPv6) messages.
- MLD message formats are almost identical to IGMPv3 messages.
- IPv6 multicast for Cisco IOS software uses MLD version 2. This version of MLD is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD version 1 interoperate with a router running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are supported.
- MLD snooping supports private VLANs. Private VLANs do not impose any restrictions on MLD snooping.
- MLD snooping constrains traffic in MAC multicast groups 0100.5e00.0001 to 0100.5eff.ffff.
- MLD snooping does not constrain Layer 2 multicasts generated by routing protocols.

MLD Snooping Querier Configuration Guidelines and Restrictions

When configuring the MLD snooping querier, follow these guidelines and restrictions:

- Configure the VLAN in global configuration mode (see Chapter 23, "Configuring VLANs").
- Configure an IPv6 address on the VLAN interface (see Chapter 30, "Configuring Layer 3 Interfaces"). When enabled, the MLD snooping querier uses the IPv6 address as the query source address.
- If there is no IPv6 address configured on the VLAN interface, the MLD snooping querier does not start. The MLD snooping querier disables itself if the IPv6 address is cleared. When enabled, the MLD snooping querier restarts if you configure an IPv6 address.
- When enabled, the MLD snooping querier does not start if it detects MLD traffic from an IPv6 multicast router.
- When enabled, the MLD snooping querier starts after 60 seconds with no MLD traffic detected from an IPv6 multicast router.
- When enabled, the MLD snooping querier disables itself if it detects MLD traffic from an IPv6 multicast router.
- QoS does not support MLD packets when MLD snooping is enabled.
- You can enable the MLD snooping querier on all the switches in the VLAN that support it. One switch is elected as the querier.

Enabling the MLD Snooping Querier

Use the MLD snooping querier to support MLD snooping in a VLAN where PIM and MLD are not configured because the multicast traffic does not need to be routed.

To enable the MLD snooping querier in a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface vlan** *vlan_ID*	Selects the VLAN interface.
Step 2	Router(config-if)# **ipv6 address** *prefix*/*prefix_length*	Configures the IPv6 address and subnet.
Step 3	Router(config-if)# **ipv6 mld snooping querier**	Enables the MLD snooping querier.
	Router(config-if)# **no ipv6 mld snooping querier**	Disables the MLD snooping querier.
Step 4	Router(config-if)# **end**	Exits configuration mode.
Step 5	Router# **show ipv6 mld interface vlan** *vlan_ID* | **include querier**	Verifies the configuration.

This example shows how to enable the MLD snooping querier on VLAN 200 and verify the configuration:

```
Router# interface vlan 200
Router(config-if)# ipv6 address 2001:0DB8:0:1::/64 eui-64
Router(config-if)# ipv6 mld snooping querier
Router(config-if)# end
Router# show ipv6 mld interface vlan 200 | include querier
  MLD snooping fast-leave is enabled and querier is enabled
```

Configuring MLD Snooping

Note To use MLD snooping, configure a Layer 3 interface in the subnet for IPv6 multicast routing or enable the MLD snooping querier in the subnet (see the "Enabling the MLD Snooping Querier" section on page 36-9).

These sections describe how to configure MLD snooping:

Note Except for the global enable command, all MLD snooping commands are supported only on VLAN interfaces.

Enabling MLD Snooping

To enable MLD snooping globally, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ipv6 mld snooping`**	Enables MLD snooping.
	`Router(config)#` **`no ipv6 mld snooping`**	Disables MLD snooping.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.
Step 3	`Router#` **`show ipv6 mld interface vlan`** *`vlan_ID`* **`| include globally`**	Verifies the configuration.

This example shows how to enable MLD snooping globally and verify the configuration:

```
Router(config)# ipv6 mld snooping
Router(config)# end
Router# show ipv6 mld interface vlan 200 | include globally
  MLD snooping is globally enabled
Router#
```

To enable MLD snooping in a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface vlan`** *`vlan_ID`*	Selects a VLAN interface.
Step 2	`Router(config-if)#` **`ipv6 mld snooping`**	Enables MLD snooping.
	`Router(config-if)#` **`no ipv6 mld snooping`**	Disables MLD snooping.

	Command	Purpose
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show ipv6 mld interface vlan** *vlan_ID* | **include snooping**	Verifies the configuration.

This example shows how to enable MLD snooping on VLAN 25 and verify the configuration:

```
Router# interface vlan 25
Router(config-if)# ipv6 mld snooping
Router(config-if)# end
Router# show ipv6 mld interface vlan 25 | include snooping
  MLD snooping is globally enabled
  MLD snooping is enabled on this interface
  MLD snooping fast-leave is enabled and querier is enabled
  MLD snooping explicit-tracking is enabled
  MLD snooping last member query response interval is 1000 ms
  MLD snooping report-suppression is disabled
Router#
```

Configuring a Static Connection to a Multicast Receiver

To configure a static connection to a multicast receiver, perform this task:

	Command	Purpose
Step 1	Router(config)# **mac-address-table static** *mac_addr* **vlan** *vlan_id* **interface** *type*[1] *slot/port* [**disable-snooping**]	Configures a static connection to a multicast receiver.
Step 2	Router(config-if)# **end**	Exits configuration mode.
Step 3	Router# **show mac-address-table address** *mac_addr*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When you configure a static connection, enter the **disable-snooping** keyword to prevent multicast traffic addressed to the statically configured multicast MAC address from also being sent to other ports in the same VLAN.

This example shows how to configure a static connection to a multicast receiver:

```
Router(config)# mac-address-table static 0050.3e8d.6400 vlan 12 interface fastethernet 5/7
```

Configuring a Multicast Router Port Statically

To configure a static connection to a multicast router, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface vlan** *vlan_ID*	Selects the VLAN interface.
Step 2	Router(config-if)# **ipv6 mld snooping mrouter interface** *type*[1] *slot/port*	Configures a static connection to a multicast router.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show ipv6 mld snooping mrouter**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

The interface to the router must be in the VLAN where you are entering the command, the interface must be administratively up, and the line protocol must be up.

This example shows how to configure a static connection to a multicast router:

```
Router(config-if)# ipv6 mld snooping mrouter interface fastethernet 5/6
Router(config-if)#
```

Configuring the MLD Snooping Query Interval

You can configure the interval for which the switch waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group.

Note When both MLD snooping fast-leave processing and the MLD snooping query interval are configured, fast-leave processing takes precedence.

To configure the interval for the MLD snooping queries sent by the switch, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ipv6 mld snooping last-member-query-interval interval`	Configures the interval for the IGMP queries sent by the switch. Default is 1 second. Valid range is 1000 to 9990 milliseconds.
Step 3	`Router# show ipv6 mld interface vlan vlan_ID | include last`	Verifies the configuration.

This example shows how to configure the MLD snooping query interval:

```
Router(config-if)# ipv6 mld snooping last-member-query-interval 1000
Router(config-if)# exit
Router# show ipv6 mld interface vlan 200 | include last
  MLD snooping last member query response interval is 1000 ms
```

Enabling Fast-Leave Processing

To enable fast-leave processing in a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ipv6 mld snooping fast-leave`	Enables fast-leave processing in the VLAN.
Step 3	`Router# show ipv6 mld interface vlan vlan_ID | include fast-leave`	Verifies the configuration.

This example shows how to enable fast-leave processing on the VLAN 200 interface and verify the configuration:

```
Router# interface vlan 200
Router(config-if)# ipv6 mld snooping fast-leave
Configuring fast leave on vlan 200
Router(config-if)# end
Router# show ipv6 mld interface vlan 200 | include fast-leave
  MLD snooping fast-leave is enabled and querier is enabled
Router#
```

Enabling SSM Safe Reporting

To enable source-specific multicast (SSM) safe reporting, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface vlan** *vlan_ID*	Selects a VLAN interface.
Step 2	Router(config-if)# **ipv6 mld snooping ssm-safe-reporting**	Enables SSM safe reporting.
	Router(config-if)# **no ipv6 mld snooping ssm-safe-reporting**	Clears the configuration.

This example shows how to SSM safe reporting:

```
Router(config)# interface vlan 10
Router(config-if)# ipv6 mld snooping ssm-safe-reporting
```

Configuring Explicit Host Tracking

Note Disabling explicit host tracking disables fast-leave processing and proxy reporting.

To enable explicit host tracking on a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface vlan** *vlan_ID*	Selects a VLAN interface.
Step 2	Router(config-if)# **ipv6 mld snooping explicit-tracking**	Enables explicit host tracking.
	Router(config-if)# **no ipv6 mld snooping explicit-tracking**	Clears the explicit host tracking configuration.
Step 3	Router# **show ipv6 mld snooping explicit-tracking vlan** *vlan_ID*	Displays the status of explicit host tracking.

This example shows how to enable explicit host tracking:

```
Router(config)# interface vlan 25
Router(config-if)# ipv6 mld snooping explicit-tracking
Router(config-if)# end
Router# show ipv6 mld snooping explicit-tracking vlan 25
Source/Group                     Interface    Reporter        Filter_mode
------------------------------------------------------------------------
```

```
10.1.1.1/226.2.2.2              Vl25:1/2     16.27.2.3       INCLUDE
10.2.2.2/226.2.2.2              Vl25:1/2     16.27.2.3       INCLUDE
```

Configuring Report Suppression

To enable report suppression on a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface vlan** *vlan_ID*	Selects a VLAN interface.
Step 2	Router(config-if)# **ipv6 mld snooping report-suppression**	Enables report suppression.
	Router(config-if)# **no ipv6 mld snooping report-suppression**	Clears the report suppression configuration.
Step 3	Router# **show ipv6 mld interface** *vlan_ID* **| include report-suppression**	Displays the status of report suppression.

This example shows how to enable explicit host tracking:

```
Router(config)# interface vlan 25
Router(config-if)# ipv6 mld snooping report-suppression
Router(config-if)# end
Router# Router# show ipv6 mld interface vlan 25 | include report-suppression
  MLD snooping report-suppression is enabled
```

Displaying MLD Snooping Information

These sections describe displaying MLD snooping information:

- Displaying Multicast Router Interfaces, page 36-14
- Displaying MAC Address Multicast Entries, page 36-15
- Displaying MLD Snooping Information for a VLAN Interface, page 36-15

Displaying Multicast Router Interfaces

When you enable IGMP snooping, the switch automatically learns to which interface the multicast routers are connected.

To display multicast router interfaces, perform this task:

Command	Purpose
Router# **show ipv6 mld snooping mrouter** *vlan_ID*	Displays multicast router interfaces.

This example shows how to display the multicast router interfaces in VLAN 1:

```
Router# show ipv6 mld snooping mrouter vlan 1
vlan            ports
-----+----------------------------------------
  1          Gi1/1,Gi2/1,Fa3/48,Router
Router#
```

Displaying MAC Address Multicast Entries

To display MAC address multicast entries for a VLAN, perform this task:

Command	Purpose
Router# **show mac-address-table multicast** *vlan_ID* [**count**]	Displays MAC address multicast entries for a VLAN.

This example shows how to display MAC address multicast entries for VLAN 1:

```
Router# show mac-address-table multicast vlan 1
vlan   mac address     type    qos             ports
-----+---------------+--------+---+--------------------------------
  1  0100.5e02.0203  static   --  Gi1/1,Gi2/1,Fa3/48,Router
  1  0100.5e00.0127  static   --  Gi1/1,Gi2/1,Fa3/48,Router
  1  0100.5e00.0128  static   --  Gi1/1,Gi2/1,Fa3/48,Router
  1  0100.5e00.0001  static   --  Gi1/1,Gi2/1,Fa3/48,Router,Switch
Router#
```

This example shows how to display a total count of MAC address entries for a VLAN:

```
Router# show mac-address-table multicast 1 count

Multicast MAC Entries for vlan 1:    4
Router#
```

Displaying MLD Snooping Information for a VLAN Interface

Note When you apply the **ipv6 mld snooping** command and associated commands on any VLAN interface, the commands continue to function even if the VLAN interface is in shutdown state.

To display MLD snooping information for a VLAN interface, perform this task:

Command	Purpose
Router# **show ipv6 mld snooping** {{**explicit-tracking** *vlan_ID*}| {**mrouter** [**vlan** *vlan_ID*]} | {**report-suppression vlan** *vlan_ID*} | {**statistics vlan** *vlan_ID*}	Displays MLD snooping information on a VLAN interface.

This example shows how to display explicit tracking information on VLAN 25:

```
Router# show ipv6 mld snooping explicit-tracking vlan 25
Source/Group                       Interface    Reporter        Filter_mode
------------------------------------------------------------------------
10.1.1.1/226.2.2.2                 Vl25:1/2     16.27.2.3       INCLUDE
10.2.2.2/226.2.2.2                 Vl25:1/2     16.27.2.3       INCLUDE
```

This example shows how to display the multicast router interfaces in VLAN 1:

```
Router# show ipv6 mld snooping mrouter vlan 1
vlan            ports
-----+----------------------------------------
  1          Gi1/1,Gi2/1,Fa3/48,Router
```

This example shows IGMP snooping statistics information for VLAN 25:

```
Router# show ipv6 mld snooping statistics interface vlan 25

Snooping staticstics for Vlan25
#channels:2
#hosts   :1

Source/Group            Interface       Reporter      Uptime        Last-Join   Last-Leave
10.1.1.1/226.2.2.2      Gi1/2:Vl25      16.27.2.3     00:01:47      00:00:50      -
10.2.2.2/226.2.2.2      Gi1/2:Vl25      16.27.2.3     00:01:47      00:00:50      -
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 7

IP Multicast

Configuring IPv4 Multicast Layer 3 Switching

This chapter describes how to configure IPv4 multicast Layer 3 switching in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding IPv4 Multicast Layer 3 Switching

These sections describe how IPv4 multicast Layer 3 switching works:

IPv4 Multicast Layer 3 Switching Overview

The Policy Feature Card (PFC) provides Layer 3 switching for IP multicast flows using the hardware replication table and hardware Cisco Express Forwarding (CEF), which uses the forwarding information base (FIB) and the adjacency table on the PFC. In systems with Distributed Forwarding Cards (DFCs), IP multicast flows are Layer 3 switched locally using Multicast Distributed Hardware Switching (MDHS). MDHS uses local hardware CEF and replication tables on each DFC to perform Layer 3 switching and rate limiting of reverse path forwarding (RPF) failures locally on each DFC-equipped switching module.

The PFC and the DFCs support hardware switching of (*,G) state flows. The PFC and the DFCs support rate limiting of non-RPF traffic.

Multicast Layer 3 switching forwards IP multicast data packet flows between IP subnets using advanced application-specific integrated circuit (ASIC) switching hardware, which offloads processor-intensive multicast forwarding and replication from network routers.

Layer 3 flows that cannot be hardware switched are still forwarded in the software by routers. Protocol Independent Multicast (PIM) is used for route determination.

The PFC and the DFCs all use the Layer 2 multicast forwarding table to determine on which ports Layer 2 multicast traffic should be forwarded (if any). The multicast forwarding table entries are populated in conjunction with Internet Group Management Protocol (IGMP) snooping (see Chapter 38, "Configuring IGMP Snooping for IPv4 Multicast Traffic").

Multicast Layer 3 Switching Cache

This section describes how the PFC and the DFCs maintain Layer 3 switching information in hardware tables.

The PFC and DFC populate the (S,G) or (*,G) flows in the hardware FIB table with the appropriate masks; for example, (S/32, G/32) and (*/0, G/32). The RPF interface and the adjacency pointer information is also stored in each entry. The adjacency table contains the rewrite and a pointer to the replication entries. If a flow matches a FIB entry, the RPF check compares the incoming interface/VLAN with the entry. A mismatch is an RPF failure, which can be rate limited if this feature is enabled. In the event of a forwarding information database (FIB) fatal error, the default error action is for the system to reset and the FIB to reload.

The route processor (RP) updates its multicast routing table and forwards the new information to the PFC whenever it receives traffic for a new flow. In addition, if an entry in the multicast routing table on the RP ages out, the RP deletes the entry and forwards the updated information to the PFC. In systems with DFCs, flows are populated symmetrically on all DFCs and on the PFC.

The Layer 3 switching cache contains flow information for all active Layer 3-switched flows. After the switching cache is populated, multicast packets identified as belonging to an existing flow can be Layer 3 switched based on the cache entry for that flow. For each cache entry, the PFC maintains a list of outgoing interfaces for the IP multicast group. From this list, the PFC determines onto which VLANs traffic from a given multicast flow should be replicated.

These commands affect the Layer 3 switching cache entries:

- When you clear the multicast routing table using the **clear ip mroute** command, all multicast Layer 3 switching cache entries are cleared.
- When you disable IP multicast routing on the RP using the **no ip multicast-routing** command, all multicast Layer 3 switching cache entries on the PFC are purged.
- When you disable multicast Layer 3 switching on an individual interface basis using the **no mls ipmulticast** command, flows that use this interface as the RPF interface are routed only by the RP in software.

Layer 3-Switched Multicast Packet Rewrite

When a multicast packet is Layer 3 switched from a multicast source to a destination multicast group, the PFC and the DFCs perform a packet rewrite that is based on information learned from the RP and stored in the adjacency table.

For example, Server A sends a multicast packet addressed to IP multicast group G1. If there are members of group G1 on VLANs other than the source VLAN, the PFC must perform a packet rewrite when it replicates the traffic to the other VLANs (the switch also bridges the packet in the source VLAN).

When the PFC receives the multicast packet, it is (conceptually) formatted as follows:

Layer 2 Frame Header		**Layer 3 IP Header**				**Data**	**FCS**
Destination	Source	Destination	Source	TTL	Checksum		
Group G1 MAC[1]	*Source A MAC*	*Group G1 IP*	*Source A IP*	*n*	*calculation1*		

1. In this example, Destination B is a member of Group G1.

The PFC rewrites the packet as follows:

- Changes the source MAC address in the Layer 2 frame header from the MAC address of the host to the MAC address of the RP (This is the burned-in MAC address of the system. This MAC address will be the same for all outgoing interfaces and cannot be modified. This MAC address can be displayed using the **show mls multicast statistics** command.)
- Decrements the IP header Time to Live (TTL) by one and recalculates the IP header checksum

The result is a rewritten IP multicast packet that appears to have been routed. The PFC replicates the rewritten packet onto the appropriate destination VLANs, where it is forwarded to members of IP multicast group G1.

After the PFC performs the packet rewrite, the packet is (conceptually) formatted as follows:

Frame Header		**IP Header**				**Data**	**FCS**
Destination	Source	Destination	Source	TTL	Checksum		
Group G1 MAC	*RP MAC*	*Group G1 IP*	*Source A IP*	*n–1*	*calculation2*		

Partially and Completely Switched Flows

When at least one outgoing Layer 3 interface for a given flow is multilayer switched and at least one outgoing interface is not multilayer switched, that flow is considered partially switched. When a partially switched flow is created, all multicast traffic belonging to that flow still reaches the RP and is forwarded by software on those outgoing interfaces that are not multilayer switched.

These sections describe partially and completely switched flow:

- Partially Switched Flows, page 37-4
- Completely Switched Flows, page 37-5

Partially Switched Flows

A flow might be partially switched instead of completely switched in these situations:

- If the switch is configured as a member of the IP multicast group on the RPF interface of the multicast source (using the **ip igmp join-group** command).
- During the registering state, if the switch is the first-hop router to the source in PIM sparse mode (in this case, the switch must send PIM-register messages to the rendezvous point [RP]).
- If the multicast TTL threshold is configured on an outgoing interface for the flow (using the **ip multicast ttl-threshold** command).
- If the multicast helper is configured on the RPF interface for the flow, and multicast to broadcast translation is required.
- If the outgoing interface is a generic routing encapsulation (GRE) tunnel interface.
- If the outgoing interface is a Distance Vector Multicast Routing Protocol (DVMRP) tunnel interface.
- If Network Address Translation (NAT) is configured on an interface and source address translation is required for the outgoing interface.
- Flows are partially switched if any of the outgoing interfaces for a given flow are not Layer 3 switched.

(S,G) flows are partially switched instead of completely switched in these situations:

- (S,G) flows are partially switched if the (S,G) entry has the RPT-bit (R bit) set.
- (S,G) flows are partially switched if the (S,G) entry does not have the SPT bit (T flag) set and the Prune bit (P flag) set.

(*,G) flows are partially switched instead of completely switched in these situations:

- (*,G) flows are partially switched on the last-hop leaf router if the shared-tree to shortest-path-tree (SPT) threshold is not equal to infinity. This allows the flow to transition from the SPT.
- (*,G) flows are partially switched if at least one (S,G) entry has the same RPF as a (*,g) entry but any of these is true:
  - The RPT flag (R bit) is not set.
  - The SPT flag (T bit) is not set.
  - The Prune-flag (P bit) is not set.
- (*,G) flows are partially switched if a DVMRP neighbor is detected on the input interface of a (*,G) entry.
- (*,G) flows are partially switched if the interface and mask entry is not installed for the RPF-interface of a (*,G) entry and the RPF interface is not a point-to-point interface.

- In PFC2 systems, (*,G) flows will be partially switched on the last-hop leaf router if the shared-tree to shortest-path-tree (SPT) threshold is not equal to infinity. This allows the flow to transition from SPT.

Note With a PFC2, flows matching an output ACL on an outgoing interface are routed in software.

Completely Switched Flows

When all the outgoing interfaces for a given flow are Layer 3 switched, and none of the above situations apply to the flow, that flow is considered completely switched. When a completely switched flow is created, the PFC prevents multicast traffic bridged on the source VLAN for that flow from reaching the RP interface in that VLAN, freeing the RP of the forwarding and replication load for that flow.

One consequence of a completely switched flow is that multicast statistics on a per-packet basis for that flow cannot be recorded. Therefore, the PFC periodically sends multicast packet and byte count statistics for all completely switched flows to the RP. The RP updates the corresponding multicast routing table entry and resets the expiration timer for that multicast route.

Note A (*,G) state is created on the PIM-RP or for PIM-dense mode but is not used for forwarding the flows, and Layer 3 switching entries are not created for these flows.

Non-RPF Traffic Processing

These sections describe non-RPF traffic processing:

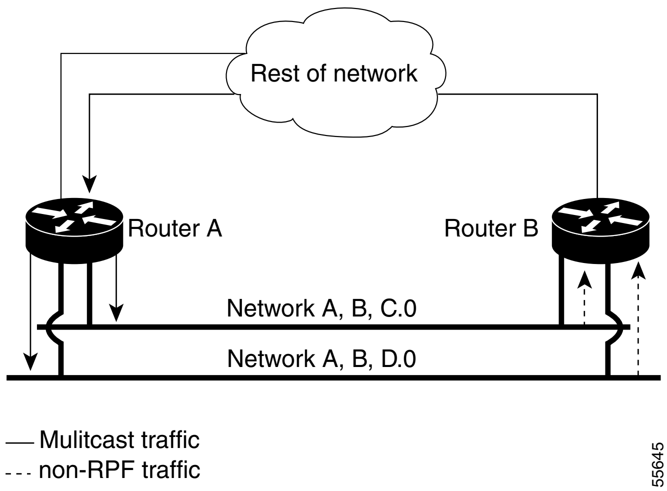
- Non-RPF Traffic Overview, page 37-5
- Filtering of RPF Failures for Stub Networks, page 37-8
- Rate Limiting of RPF Failure Traffic, page 37-8

Non-RPF Traffic Overview

In a redundant configuration where multiple routers connect to the same LAN segment, only one router forwards the multicast traffic from the source to the receivers on the outgoing interfaces (see Figure 37-1). In this kind of topology, only the PIM designated router (PIM DR) forwards the data in the common VLAN, but the non-PIM DR receives the forwarded multicast traffic. The redundant router (non-PIM DR) must drop this traffic because it has arrived on the wrong interface and fails the RPF check. Traffic that fails the RPF check is called non-RPF traffic.

The PFC hardware processes non-RPF traffic by filtering (dropping) or rate limiting the non-RPF traffic.

Figure 37-1 Redundant Multicast Router Configuration in a Stub Network



The conflicting requirements for the non-RPF traffic in the PFC3 prevent the majority of traffic from reaching the RP. However, leaking some packets to the RP ensures that correct protocol operations are met by using the hardware NetFlow table on the PFC3. By default, the NetFlow non-RPF traffic handling is enabled on the Supervisor Engine 720, and it cannot be disabled.

When the first non-RPF packet for an existing multicast FIB table entry is received, a matching (S,G) FIB TCAM entry for the packet is found; however, the RPF check is mismatched so a non-RPF NetFlow entry for the multicast entry is created in the NetFlow table. The packet is then bridged to the non-RPF VLAN and the RP for further processing.

The NetFlow search engine removes all of the non-RPF NetFlow entries from the hardware every 20 seconds. The next non-RPF packet received for a FIB TCAM entry triggers the creation of a non-RPF NetFlow entry while bridging the packet to the RP CPU. This operation results in only a single packet for each non-RPF NetFlow entry which will be bridged to the RP CPU approximately every 20 seconds, regardless of the rate of the various multicast traffic flows passing through the system.

Note The non-RPF NetFlow entries are created only for PIM-SM, PIM-DM, and SSM multicast FIB entries. Because the Bidir PIM does not use the PIM assert mechanism, non-RPF NetFlow entries are never created for Bidir PIM FIB entries.

In addition to the 20-second periodic timer, any non-RPF NetFlow entries that remain unused for more than 2 seconds are automatically purged to conserve NetFlow table resources. To view details of the multicast non-RPF entries in the NetFlow table, use the **show mls netflow ip multicast rpf-fail** command from the SP console.

This example shows how to display RPF fail information:

```
Router (config)# mls netflow ip multicast rpf-fail
Source       Destination     RPF      #packets   #bytes     Type
+---------+--------------+--------+-----------+--------+--------------------------
10.14.1.60   225.0.0.158   Vl19        2          92        NRPF
10.14.1.68   225.0.1.110   Vl19        2          92        NRPF
10.14.1.60   225.0.1.102   Vl19        2          92        NRPF
10.14.1.137  225.0.0.235   Vl19      121        5566        NRPF
10.14.1.135  225.0.0.233   Vl19      122        5612        NRPF
10.14.1.127  225.0.0.225   Vl19      122        5612        NRPF
10.14.1.124  225.0.0.222   Vl19      122        5612        NRPF
10.14.1.81   225.0.1.123   Vl19        2          92        NRPF
10.14.1.67   225.0.1.109   Vl19        2          92        NRPF
```

```
10.14.1.41    225.0.1.83    Vl19        2          92       NRPF
```

If the NetFlow table is full when the system attempts to create a new non-RPF NetFlow entry, the packet is bridged in the VLAN and is also forwarded to a reserved adjacency that punts the packet to the RP CPU. By default, the traffic sent to this adjacency entry is not rate limited. To protect the RP CPU from excess non-RPF packets, rate limit the traffic sent to this adjacency by using the **mls rate limit multicast non-rpf rate burst** command. You can check whether the NetFlow table is full by using the **show mls netflow table contention summary** command.

Filtering of RPF Failures for Stub Networks

The PFC and the DFCs support ACL-based filtering of RPF failures for sparse mode stub networks. When you enable the ACL-based method of filtering RPF failures by entering the **mls ip multicast stub** command on the redundant router, the following ACLs automatically download to the PFC and are applied to the interface you specify:

```
access-list 100 permit ip A.B.C.0 0.0.0.255 any
access-list 100 permit ip A.B.D.0 0.0.0.255 any
access-list 100 permit ip any 224.0.0.0 0.0.0.255
access-list 100 permit ip any 224.0.1.0 0.0.0.255
access-list 100 deny ip any 224.0.0.0 15.255.255.255
```

The ACLs filter RPF failures and drop them in hardware so that they are not forwarded to the router.

Use the ACL-based method of filtering RPF failures only in sparse mode stub networks where there are no downstream routers. For dense mode groups, RPF failure packets have to be seen on the router for the PIM assert mechanism to function properly. Use CEF-based or NetFlow-based rate limiting to limit the rate of RPF failures in dense mode networks and sparse mode transit networks.

For information on configuring ACL-based filtering of RPF failures, see the "Configuring ACL-Based Filtering of RPF Failures" section on page 37-18.

Rate Limiting of RPF Failure Traffic

When you enable rate limiting of packets that fail the RPF check (non-RPF packets), most non-RPF packets are dropped in hardware. According to the multicast protocol specification, the router needs to receive the non-RPF packets for the PIM assert mechanism to function properly, so all non-RPF packets cannot be dropped in hardware.

When a non-RPF packet is received, a NetFlow entry is created for each non-RPF flow.

When the first non-RPF packet arrives, the PFC bridges the packet to the RP and to any bridged ports and creates a NetFlow entry that contains source, group, and ingress interface information, after which the NetFlow entry handles all packets for that source and group, sending packets only to bridged ports and not to the RP.

To support the PIM assert mechanism, the PFC periodically forwards a percentage of the non-RPF flow packets to the RP. The first packets for directly connected sources in PIM sparse mode are also rate-limited and are processed by the CPU. By default, rate limiting of RPF failures is disabled .

The non-RPF hardware rate limiter offers an alternative method for handling the non-RPF multicast traffic. You can enable the traffic-handling method by using the **mls rate-limit multicast non-rpf** command. The configured rate represents the aggregate of all non-RPF traffic punted to the RP CPU. We recommend that you enable the rate limiter when the NetFlow table is full. By default, the rate limiter is disabled.

Multicast Boundary

The multicast boundary feature allows you to configure an administrative boundary for multicast group addresses. By restricting the flow of multicast data packets, you can reuse the same multicast group address in different administrative domains.

You configure the multicast boundary on an interface. A multicast data packet is blocked from flowing across the interface if the packet's multicast group address matches the access control list (ACL) associated with the multicast boundary feature.

Multicast boundary ACLs can be processed in hardware by the Policy Feature Card (PFC), a Distributed Forwarding Card (DFC), or in software by the RP. The multicast boundary ACLs are programmed to match the destination address of the packet. These ACLs are applied to traffic on the interface in both directions (input and output).

To support multicast boundary ACLs in hardware, the switch creates new ACL TCAM entries or modifies existing ACL TCAM entries (if other ACL-based features are active on the interface). To verify TCAM resource utilization, enter the **show tcam counts ip** command.

If you configure the **filter-autorp** keyword, the administrative boundary also examines auto-RP discovery and announcement messages and removes any auto-RP group range announcements from the auto-RP packets that are denied by the boundary ACL.

Understanding IPv4 Bidirectional PIM

The PFC3 supports hardware forwarding of IPv4 bidirectional PIM groups. To support IPv4 bidirectional PIM groups, the PFC3 implements a new mode called designated forwarder (DF) mode. The designated forwarder is the router elected to forward packets to and from a segment for a IPv4 bidirectional PIM group. In DF mode, the switch accepts packets from the RPF and from the DF interfaces.

When the switch is forwarding IPv4 bidirectional PIM groups, the RPF interface is always included in the outgoing interface list of (*,G) entry, and the DF interfaces are included depending on IGMP/PIM joins.

If the route to the RP becomes unavailable, the group is changed to dense mode. Should the RPF link to the RP become unavailable, the IPv4 bidirectional PIM flow is removed from the hardware FIB.

For information on configuring IPv4 bidirectional PIM, see the “Configuring IPv4 Bidirectional PIM” section on page 37-24.

Default IPv4 Multicast Layer 3 Switching Configuration

Table 37-1 shows the default IP multicast Layer 3 switching configuration.

Table 37-1 Default IP Multicast Layer 3 Switching Configuration

Feature	Default Value
ACL for stub networks	Disabled on all interfaces
Installing of directly connected subnet entries	Enabled globally
Multicast routing	Disabled globally
PIM routing	Disabled on all interfaces
IP multicast Layer 3 switching	Enabled when multicast routing is enabled and PIM is enabled on the interface
Shortcut consistency checking	Enabled

Internet Group Management Protocol (IGMP) snooping is enabled by default on all VLAN interfaces. If you disable IGMP snooping on an interface, multicast Layer 3 flows are still switched by the hardware. Bridging of the flow on an interface with IGMP snooping disabled causes flooding to all forwarding interfaces of the VLAN. For details on configuring IGMP snooping, see Chapter 38, "Configuring IGMP Snooping for IPv4 Multicast Traffic."

IPv4 Multicast Layer 3 Switching Configuration Guidelines and Restrictions

These sections describe IP Multicast Layer 3 switching configuration restrictions:

- Restrictions, page 37-10
- Unsupported Features, page 37-10

Restrictions

IP multicast Layer 3 switching is not provided for an IP multicast flow in the following situations:

- For IP multicast groups that fall into the range 224.0.0.* (where * is in the range 0 to 255), which is used by routing protocols. Layer 3 switching is supported for groups 224.0.2.* to 239.*.*.*.

Note Groups in the 224.0.0.* range are reserved for routing control packets and must be flooded to all forwarding ports of the VLAN. These addresses map to the multicast MAC address range 01-00-5E-00-00-*xx*, where *xx* is in the range 0–0xFF.

- For PIM auto-RP multicast groups (IP multicast group addresses 224.0.1.39 and 224.0.1.40).
- For packets with IP options. However, packets in the flow that do not specify IP options are hardware switched.
- For source traffic received on tunnel interfaces (such as MBONE traffic).
- If a (S,G) entry for sparse mode does not have the SPT-bit, RPT-bit, or Pruned flag set.
- A (*,G) entry is not hardware switched if at least one (S,G) entry has an RPF different from the (*,G) entry's RPF and the (S,G) is not hardware switched.
- If the ingress interface of a (S,G) or (*,G) entry is null, except if the (*,G) entry is a IPv4 bidirectional PIM entry and the switch is the RP for the group.
- For IPv4 bidirectional PIM entries when a DF interface or RPF interface is a tunnel.
- GRE tunnel encapsulation and de-encapsulation for multicast packets is handled in software.
- Supervisor Engine 32 does not support egress multicast replication and cannot detect the multicast replication mode.

Unsupported Features

If you enable IP multicast Layer 3 switching, IP accounting for Layer 3 interfaces does not report accurate values. The **show ip accounting** command is not supported.

Configuring IPv4 Multicast Layer 3 Switching

These sections describe how to configure IP multicast Layer 3 switching:

Note When you are in configuration mode you can enter EXEC mode commands by entering the **do** keyword before the EXEC mode command.

Source-Specific Multicast with IGMPv3, IGMP v3lite, and URD

For complete information and procedures about source-specific multicast with IGMPv3, IGMP v3lite, and URL Rendezvous Directory (URD), see this URL:

http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfssm.html

Enabling IPv4 Multicast Routing Globally

You must enable IP multicast routing globally before you can enable IP multicast Layer 3 switching on Layer 3 interfaces.

For complete information and procedures, see these publications:

- *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.2, at this URL:
  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/fipr_c.html
- *Cisco IOS IP and IP Routing Command Reference*, Release 12.1, at this URL:
  http://www.cisco.com/en/US/docs/ios/12_2/ipaddr/command/reference/fipras_r.html

To enable IP multicast routing globally, perform this task:

Command	Purpose
`Router(config)# ip multicast-routing`	Enables IP multicast routing globally.

This example shows how to enable multicast routing globally:

```
Router(config)# ip multicast-routing
Router(config)#
```

Enabling IPv4 PIM on Layer 3 Interfaces

You must enable PIM on the Layer 3 interfaces before IP multicast Layer 3 switching functions on those interfaces.

To enable IP PIM on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{vlan vlan_ID}	{type[1] slot/port}}`	Selects an interface to configure.
Step 2	`Router(config-if)# ip pim {dense-mode	sparse-mode	sparse-dense-mode}`	Enables IP PIM on a Layer 3 interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable PIM on an interface using the default mode (**sparse-dense-mode**):

```
Router(config-if)# ip pim
```

This example shows how to enable PIM sparse mode on an interface:

```
Router(config-if)# ip pim sparse-mode
```

Enabling IP Multicast Layer 3 Switching Globally

To enable hardware switching of multicast routes globally on your system, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls ip multicast`	Globally enables hardware switching of multicast routes.
Step 2	`Router# show mls ip multicast`	Displays MLS IP multicast configuration.

This example shows how to globally enable hardware switching of multicast routes:

```
Router(config)# mls ip multicast
Router(config)#
```

Enabling IP Multicast Layer 3 Switching on Layer 3 Interfaces

IP multicast Layer 3 switching is enabled by default on the Layer 3 interface when you enable PIM on the interface. Perform this task only if you disabled IP multicast Layer 3 switching on the interface and you want to reenable it.

PIM can be enabled on any Layer 3 interface, including VLAN interfaces.

Note You must enable PIM on all participating Layer 3 interfaces before IP multicast Layer 3 switching will function. For information on configuring PIM on Layer 3 interfaces, see the "Enabling IPv4 PIM on Layer 3 Interfaces" section on page 37-12.

To enable IP multicast Layer 3 switching on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {{**vlan** *vlan_ID*} | {*type*[1] *slot/port*}}	Selects an interface to configure.
Step 2	Router(config-if)# **mls ip multicast**	Enables IP multicast Layer 3 switching on a Layer 3 interface.
Step 3	Router(config-if)# **exit**	Returns you to global configuration mode.
Step 4	Router # **[no] mls ip multicast syslog**[2]	(Optional) Enables display of multicast related syslog messages on console.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

2. This command is only available in IOS Software Release 12.2SXI and later, and is disabled by default.

This example shows how to enable IP multicast Layer 3 switching on a Layer 3 interface:

```
Router(config-if)# mls ip multicast
Router(config-if)#
```

Configuring the Replication Mode

Note Supervisor Engine 32 and the Cisco ME 6500 Series Ethernet switches support only ingress replication mode.

The Supervisor Engine 720 and Supervisor Engine 720-10GE support the **egress** keyword. Support for the **egress** keyword is called "Multicast Enhancement - Replication Mode Detection" in the release notes and Feature Navigator.

By default, the switch automatically detects the replication mode based on the switching modules installed in the system. If all switching modules are capable of egress replication, the switch uses egress-replication mode. If the switch detects switching modules that are not capable of egress replication, the replication mode automatically changes to ingress replication. You can override this action by entering the **mls ip multicast replication-mode egress** command so that the switch continues to work in egress-replication mode even if there are fabric-enabled modules installed that do not support egress replication. You can also configure the switch to operate only in ingress-replication mode.

If the switch is functioning in automatic detection mode, and you install a switching module that cannot perform egress replication, the following occurs:

- The switch reverts to ingress mode
- A system log is generated

If the switch is functioning in forced egress mode, a system log is created that will display the presence of modules that are not capable of egress replication mode.

Note
- If you configure forced egress mode in a switch that has fabric-enabled modules that are not capable of egress replication, you must make sure that these modules are not sourcing or receiving multicast traffic.
- Egress mode is not compatible with QoS or SPAN. When QoS is configured, egress replication can result in the incorrect COS or DSCP marking. When SPAN is configured, egress replication can result in multicast packets not being sent to the SPAN destination port. If you are using QoS or SPAN and your switching modules are capable of egress replication, enter the **mls ip multicast replication-mode ingress** command to force ingress replication.
- During a change from egress- to ingress-replication mode, traffic interruptions may occur because the shortcuts will be purged and reinstalled. To avoid interruptions in traffic forwarding, enter the **mls ip multicast replication-mode ingress** command in global configuration mode. This command forces the system to operate in ingress-replication mode.
- The **no** form of the **mls ip multicast replication-mode ingress** command restores the system to automatic detection mode.

To enable IP multicast Layer 3 switching, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls ip multicast replication-mode** [**egress** | **ingress**]	Specifies the replication mode.
Step 2	Router# **show mls ip multicast capability**	Displays the configured replication mode.
Step 3	Router# **show mls ip multicast summary**	Displays the replication mode and if automatic detection is enabled or disabled.

This example shows how to enable the replication mode:

```
Router (config)# mls ip multicast replication-mode egress
Router# show mlp ip multicast capability
Current mode of replication is Ingress
Configured replication mode is Egress

Slot           Multicast replication capability
    2                       Egress
    3                       Egress
    4                       Ingress
    5                       Egress
    6                       Egress
```

```
Router# show mls ip multicast summary
4 MMLS entries using 656 bytes of memory
Number of partial hardware-switched flows:2
Number of complete hardware-switched flows:2

Directly connected subnet entry install is enabled
Current mode of replication is Ingress
Auto-detection of replication mode is enabled
Consistency checker is enabled
Router (config)#
```

Enabling Local Egress Replication

Note Supervisor Engine 32 and the Cisco ME 6500 Series Ethernet switches support only ingress replication mode.

With a Supervisor Engine 720 or Supervisor Engine 720-10GE, you can unconditionally enable local egress replication. This feature is called “Multicast enhancement - egress replication performance improvement” in the release notes and Feature Navigator.

DFC-equipped modules with dual switch-fabric connections host two packet replication engines, one per fabric connection. Each replication engine is responsible for forwarding packets to and from the interfaces associated with the switch-fabric connections. The interfaces that are associated with a switch-fabric connection are considered to be “local” from the perspective of the packet replication engine. When local egress replication mode is not enabled, both replication engines have the complete outgoing interface list for all modules, and the replication engines process and then drop traffic for nonlocal interfaces.

Local egress replication mode limits the outgoing interface list to only the local interfaces that each replication engine supports, which prevents unnecessary processing of multicast traffic.

Local egress replication is supported with the following software configuration and hardware:

- IPv4 egress replication mode.
- Dual fabric-connection DFC-equipped modules.
- All releases can provide local egress replication on Layer 3-routed interfaces and subinterfaces that are not members of an EtherChannel.
- Releases earlier than Release 12.2(33)SXI cannot provide local egress replication on members of Layer 3 EtherChannels or on VLAN interfaces.
- Release 12.2(33)SXI and later releases add local egress replication support for members of Layer 3 EtherChannels and VLAN interfaces.

The local egress replication feature is not supported for the following internal VLANs:

- Egress internal VLAN
- Partial-shortcut internal VLAN
- Internal VLAN for Multicast VPN Multicast Distribution Tree (MDT) tunnel
- Point-to-point tunnel internal VLAN
- QoS internal VLAN

Note The local egress replication feature is not supported with IPv6 multicast or in a system that has a mix of IPv4 and IPv6 multicast enabled.

To enable local egress replication, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls ip multicast egress local**	Enables local egress replication. **Note** This command requires a system reset for the configuration to take effect.
Step 2	Router # **reload**	Reloads the system.
Step 3	Router# **show mls ip multicast capability** Router# **show mls cef ip multicast detail**	Displays the configured replication mode.

This example shows how to enable local egress replication:

```
Router (config)# mls ip multicast egress local
Router (config)# exit
Router # reload
Router # show mls ip multicast capability
Current mode of replication is Ingress
Configured replication mode is Egress
Egress Local is Enabled
Slot Multicast replication capability Egress Local
2 Egress No
3 Egress Yes
4 Ingress No
5 Egress No
6 Egress No
```

Configuring the Layer 3 Switching Global Threshold

You can configure a global multicast rate threshold (specified in packets per second) below which all multicast traffic is routed by the RP. This configuration prevents creation of switching cache entries for low-rate Layer 3 flows.

Note This command does not affect flows that are already being routed. To apply the threshold to existing routes, clear the route and let it reestablish.

To configure the Layer 3 switching threshold, perform this task:

Command	Purpose
Router(config)# **mls ip multicast threshold** *ppsec*	Configures the IP MMLS threshold.

This example shows how to configure the Layer 3 switching threshold to 10 packets per second:

```
Router(config)# mls ip multicast threshold 10
Router(config)#
```

Enabling Installation of Directly Connected Subnets

In PIM sparse mode, a first-hop router that is the designated router for the interface may need to encapsulate the source traffic in a PIM register message and unicast it to the rendezvous point. To prevent new sources for the group from being learned in the routing table, the (*,G) flows should remain as completely hardware-switched flows. When (subnet/mask, 224/4) entries are installed in the hardware, the FIB allows both (*,G) flows to remain completely hardware-switched flows, and new, directly connected sources to be learned correctly. The installation of directly connected subnets is enabled globally by default. One (subnet/mask, 224/4) is installed per PIM-enabled interface.

To view FIB entries, enter the **show mls ip multicast connected** command.

To enable installation of directly connected subnets, perform this task:

Command	Purpose
`Router(config)# mls ip multicast connected`	Enables installation of directly connected subnets.

This example shows how to enable installation of directly connected subnets:

```
Router(config)# mls ip multicast connected
Router(config)#
```

Specifying the Flow Statistics Message Interval

By default, the switch processor (SP) forwards flow statistics messages to the route processor (RP) every 25 seconds. The messages are forwarded in batches, and each batch of messages contains statistics for 25 percent of all flows. If you leave the interval at the default of 25 seconds, it will take 100 seconds to forward statistics for all flows to the RP.

To specify how often flow statistics messages forwarded from the SP to the RP, perform this task:

Command	Purpose
`Router(config)# mls ip multicast flow-stat-timer num`	Specifies how the SP forwards flow statistics messages to the RP.

This example shows how to configure the SP to forward flow statistics messages to the RP every 10 seconds:

```
Router(config)# mls ip multicast flow-stat-timer 10
Router(config)#
```

Enabling Shortcut-Consistency Checking

When you enable the shortcut-consistency checking feature, the multicast route table and the multicast-hardware entries are checked for consistency, and any inconsistencies are corrected. You can view inconsistencies by entering the **show mls ip multicast consistency-check** command.

If consistency checking is enabled, the multicast route table will be scanned every two seconds and a full scan is completed within 4 minutes.

To enable shortcut-consistency checking, perform this task:

Command	Purpose
`Router(config)# mls ip multicast consistency-check`	Enables shortcut-consistency checking.

This example shows how to enable the hardware shortcut-consistency checker:

```
Router (config)# mls ip multicast consistency-check
Router (config)#
```

Configuring ACL-Based Filtering of RPF Failures

When you configure ACL-based filtering of RPF failures, ACLs that filter RPF failures in hardware are downloaded to the hardware-based ACL engine and applied on the interface you specify.

To enable ACL-based filtering of RPF failures on an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *number*}}	Selects an interface to configure.
Step 2	Router(config-if)# **mls ip multicast stub**	Enables ACL-based filtering of RPF failures on an interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Displaying RPF Failure Rate-Limiting Information

To display RPF failure rate-limiting information, perform this task:

Command	Purpose
Router# **show mls ip multicast summary**	Displays RPF failure rate-limiting information.

This example shows how to display RPF failure rate-limiting information:

```
Router# show mls ip multicast summary
10004 MMLS entries using 1280464 bytes of memory
Number of partial hardware-switched flows:4
Number of complete hardware-switched flows:10000
Router#
```

Configuring Multicast Boundary

To configure a multicast boundary, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`vlan`** *`vlan_ID`*`} | {`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`number`*`}}`	Selects an interface to configure.
Step 2	`Router(config-if)#` **`ip multicast boundary`** *`access_list`* `[`**`filter-autorp`**`]`	Enables an administratively scoped boundary on an interface. • For *access_list*, specify the access list that you have configured to filter the traffic at this boundary. • (Optional) Specify **filter-autorp** to filter auto-RP messages at this boundary.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note In releases earlier than 12.2(33)SXI, the switch creates an empty ACL (with implicit deny any any) even though the ACL is not preconfigured. However, from 12.2(33)SXI or later releases, if the ACL is not preconfigured, the **ip multicast boundary** command will not create an empty ACL (with implicit deny any any).

Note If you configure the **filter-autorp** keyword, the administrative boundary examines auto-RP discovery and announcement messages and removes any auto-RP group range announcements from the auto-RP packets that are denied by the boundary ACL. An auto-RP group range announcement is permitted and passed by the boundary only if all addresses in the auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the auto-RP message before the auto-RP message is forwarded.

The following example sets up a multicast boundary for all administratively scoped addresses:

```
Router (config)# access-list 1 deny 239.0.0.0 0.255.255.255
Router (config)# access-list 1 permit 224.0.0.0 15.255.255.255
Router (config)# interface gigabitethernet 5/2
Router (config-if)# ip multicast boundary 1
```

Displaying IPv4 Multicast Layer 3 Hardware Switching Summary

Note The **show interface statistics** command does not display hardware-switched packets, only packets switched by software.

The **show ip pim interface count** command displays the IP multicast Layer 3 switching enable state on IP PIM interfaces and the number of packets received and sent on the interface.

To display IP multicast Layer 3 switching information for an IP PIM Layer 3 interface, perform one of these tasks:

Command	Purpose
Router# **show ip pim interface** [{{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *number*}}] **count**	Displays IP multicast Layer 3 switching enable state information for all RP IP PIM Layer 3 interfaces.
Router# **show ip interface**	Displays the IP multicast Layer 3 switching enable state on the Layer 3 interfaces.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

These examples show how to display the IP PIM configuration of the interfaces:

```
Router# show ip pim interface count

State:* - Fast Switched, D - Distributed Fast Switched
      H - Hardware Switching Enabled
Address          Interface          FS  Mpackets In/Out
10.15.1.20       GigabitEthernet4/8 * H 952/4237130770
10.20.1.7        GigabitEthernet4/9 * H 1385673757/34
10.25.1.7        GigabitEthernet4/10* H 0/34
10.11.1.30       FastEthernet6/26   * H 0/0
10.37.1.1        FastEthernet6/37   * H 0/0
1.22.33.44       FastEthernet6/47   * H 514/68
```

The “*” flag indicates that this interface can be fast switched and the “H” flag indicates that this interface is hardware switched. The “In” flag indicates the number of multicast packet bytes that have been received on the interface. The “Out” flag indicates the number of multicast packet bytes that have been forwarded from this interface.

```
Router# show ip mroute count
IP Multicast Statistics
56 routes using 28552 bytes of memory
13 groups, 3.30 average sources per group
Forwarding Counts:Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts:Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group:224.2.136.89, Source count:1, Group pkt count:29051
  Source:132.206.72.28/32, Forwarding:29051/-278/1186/0, Other:85724/8/56665
Router#
```

Note The -tive counter means that the outgoing interface list of the corresponding entry is NULL, and this indicates that this flow is still active.

This example shows how to display the IP multicast Layer 3 switching configuration of interface VLAN 10:

```
Router# show ip interface vlan 10
Vlan10 is up, line protocol is up
  Internet address is 10.0.0.6/8
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.1 224.0.0.2 224.0.0.13 224.0.0.10
  Outgoing access list is not set
  Inbound  access list is not set
```

```
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are never sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP Fast switching turbo vector
  IP Normal CEF switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
IP multicast multilayer switching is enabled
IP mls switching is enabled
Router#
```

This example shows how to display the IP multicast Layer 3 switching configuration of Gigabit Ethernet interface 1/2:

```
Router# show interfaces gigabitEthernet 1/2
GigabitEthernet1/2 is up, line protocol is up (connected)
  Hardware is C6k 1000Mb 802.3, address is 0001.c9db.2441 (bia 0001.c9db.2441)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
  Last clearing of "show interface" counters 00:05:13
  ….
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 10000 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     284 packets input, 113104 bytes, 0 no buffer
     Received 284 broadcasts (284 multicast)
     0 runts, 41 giants, 0 throttles
     41 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     198 packets output, 14732 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Router#
```

Displaying the IPv4 Multicast Routing Table

The **show ip mroute** command displays the IP multicast routing table.

To display the IP multicast routing table, perform this task:

Command	Purpose
`Router#` **`show ip mroute partical-sc`** [*hostname* | *group_number*]	Displays the IP multicast routing table and the hardware-switched interfaces.

This example shows how to display the IP multicast routing table:

```
Router# show ip mroute 230.13.13.1
IP Multicast Routing Table
Flags:D - Dense, S - Sparse, s - SSM Group, C - Connected, L - Local,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP, U - URD, I - Received Source Specific Host
           Report
Outgoing interface flags:H - Hardware switched
Timers:Uptime/Expires
Interface state:Interface, Next-Hop or VCD, State/Mode

(*, 230.13.13.1), 00:16:41/00:00:00, RP 10.15.1.20, flags:SJC
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20
  Outgoing interface list:
 GigabitEthernet4/9, Forward/Sparse-Dense, 00:16:41/00:00:00, H

(*, 230.13.13.2), 00:16:41/00:00:00, RP 10.15.1.20, flags:SJC
 Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/9, Forward/Sparse-Dense, 00:16:41/00:00:00, H

(10.20.1.15, 230.13.13.1), 00:14:31/00:01:40, flags:CJT
 Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/9, Forward/Sparse-Dense, 00:14:31/00:00:00, H
(132.206.72.28, 224.2.136.89), 00:14:31/00:01:40, flags:CJT
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
 Outgoing interface list:Null
Router#
```

Note The RPF-MFD flag indicates that the flow is completely switched by the hardware. The H flag indicates the flow is switched by the hardware on the outgoing interface.

Displaying IPv4 Multicast Layer 3 Switching Statistics

The **show mls ip multicast** command displays detailed information about IP multicast Layer 3 switching.

To display detailed IP multicast Layer 3 switching information, perform one of these tasks:

Command	Purpose
Router# **show mls ip multicast group** *ip_address* [**interface** *type slot/port* | **statistics**]	Displays IP multicast Layer 3 switching group information.
Router# **show mls ip multicast interface** {{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *number*}} [**statistics** | **summary**]	Displays IP multicast Layer 3 switching details for all interfaces.
Router# **show mls ip multicast source** *ip_address* [**interface** {{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *number*}} | **statistics**]	Displays IP multicast Layer 3 switching source information.
Router# **show mls ip multicast summary**	Displays a summary of IP multicast Layer 3 switching information.
Router# **show mls ip multicast statistics**	Displays IP multicast Layer 3 switching statistics.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to display information on a specific IP multicast Layer 3 switching entry:

```
Router# show mls ip multicast group 10.1.0.11
Multicast hardware switched flows:
Total shortcut installed: 0
```

This example shows how to display IP multicast group information:

```
Router# show mls ip multicast group 230.13.13.1 source 10.20.1.15
Multicast hardware switched flows:
(10.20.1.15, 230.13.13.1) Incoming interface:Gi4/8, Packets switched:0
Hardware switched outgoing interfaces:Gi4/9
RPF-MFD installed

Total hardware switched flows :1
Router#
```

This example shows how to display IP multicast Layer 3 switching information for VLAN 10:

```
Router# show mls ip multicast interface vlan 10
Multicast hardware switched flows:
(10.1.0.15, 224.2.2.15) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
MFD installed: Vlan10

(10.1.0.19, 224.2.2.19) Incoming interface: Vlan10, Packets switched: 1970
Hardware switched outgoing interfaces:
MFD installed: Vlan10

(10.1.0.11, 224.2.2.11) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
MFD installed: Vlan10

(10.1.0.10, 224.2.2.10) Incoming interface: Vlan10, Packets switched: 2744
Hardware switched outgoing interfaces:
MFD installed: Vlan10
```

```
(10.1.0.17, 224.2.2.17) Incoming interface: Vlan10, Packets switched: 3340
Hardware switched outgoing interfaces:
MFD installed: Vlan10

(10.1.0.13, 224.2.2.13) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
```

This example shows how to display the IP multicast Layer 3 switching statistics:

```
Router# show mls ip multicast statistics
MLS Multicast Operation Status:
MLS Multicast configuration and state:
    Router Mac: 00e0.b0ff.7b00, Router IP: 33.0.33.24
    MLS multicast operating  state: ACTIVE
    Shortcut Request Queue size 4
    Maximum number of allowed outstanding messages: 1
    Maximum size reached from feQ: 3096
    Feature Notification sent: 1
    Feature Notification Ack received: 1
    Unsolicited Feature Notification received: 0
    MSM sent: 205170
    MSM ACK received: 205170
    Delete notifications received: 0
    Flow Statistics messages received: 35211
MLS Multicast statistics:
    Flow install Ack: 996508
    Flow install Nack: 1
    Flow update Ack: 1415959
    Flow update Nack: 0
    Flow delete Ack: 774953
    Complete flow install Ack: 958469
Router#
```

Configuring IPv4 Bidirectional PIM

These sections describe how to configure IPv4 bidirectional protocol independent multicast (PIM):

- Enabling IPv4 Bidirectional PIM Globally, page 37-24
- Configuring the Rendezvous Point for IPv4 Bidirectional PIM Groups, page 37-25
- Setting the IPv4 Bidirectional PIM Scan Interval, page 37-25
- Displaying IPv4 Bidirectional PIM Information, page 37-26

Enabling IPv4 Bidirectional PIM Globally

To enable IPv4 bidirectional PIM, perform this task:

Command	Purpose
Router(config)# **ip pim bidir-enable**	Enables IPv4 bidirectional PIM globally on the switch.

This example shows how to enable IPv4 bidirectional PIM on the switch:

```
Router(config)# ip pim bidir-enable
Router(config)#
```

Configuring the Rendezvous Point for IPv4 Bidirectional PIM Groups

To statically configure the rendezvous point for an IPv4 bidirectional PIM group, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip pim rp-adress`** *`ip_address access_list`* `[`**`override`**`]`	Statically configures the IP address of the rendezvous point for the group. When you specify the **override** option, the static rendezvous point is used.
Step 2	`Router(config)#` **`access-list`** *`access-list`* **`permit`** `|` **`deny`** *`ip_address`*	Configures an access list.
Step 3	`Router(config)#` **`ip pim send-rp-announce`** *`type number`* **`scope`** *`ttl_value`* `[`**`group-list`** `access-list] [`**`interval`** `seconds] [`**`bidir`**`]`	Configures the system to use auto-RP to configure groups for which the router will act as a rendezvous point (RP).
Step 4	`Router(config)#` **`ip access-list standard`** *`access-list-name`* **`permit`** `|` **`deny`** *`ip_address`*	Configures a standard IP access list.
Step 5	`Router(config)#` **`mls ip multicast`**	Enables MLS IP multicast.

This example shows how to configure a static rendezvous point for an IPv4 bidirectional PIM group:

```
Router(config)# ip pim rp-address 10.0.0.1 10 bidir override
Router(config)# access-list 10 permit 224.1.0.0 0.0.255.255
Router(config)# ip pim send-rp-announce Loopback0 scope 16 group-list c21-rp-list-0 bidir
Router(config)# ip access-list standard c21-rp-list-0 permit 230.31.31.1 0.0.255.255
```

Setting the IPv4 Bidirectional PIM Scan Interval

You can specify the interval between the IPv4 bidirectional PIM RP Reverse Path Forwarding (RPF) scans.

To set the IPv4 bidirectional PIM RP RPF scan interval, perform this task:

Command	Purpose
`Router(config)#` **`mls ip multicast bidir gm-scan-interval`** *`interval`*	Specifies the IPv4 bidirectional PIM RP RPF scan interval; valid values are from 1 to 1000 seconds. The default is 10 seconds.

This example shows how to set the IPv4 bidirectional PIM RP RPF scan interval:

```
Router(config)# mls ip multicast bidir gm-scan-interval 30
Router(config)#
```

Displaying IPv4 Bidirectional PIM Information

To display IPv4 bidirectional PIM information, perform one of these tasks:

Command	Purpose
Router# **show ip pim rp mapping** [**in-use**]	Displays mappings between PIM groups and rendezvous points and shows learned rendezvous points in use.
Router# **show mls ip multicast rp-mapping** [*rp_address*]	Displays PIM group to active rendezvous points mappings.
Router# **show mls ip multicast rp-mapping gm-cache**	Displays information based on the group/mask ranges in the RP mapping cache.
Router# **show mls ip multicast rp-mapping df-cache**	Displays information based on the DF list in RP mapping cache.
Router# **show mls ip multicast bidir**	Displays IPv4 bidirectional PIM information.
Router# **show ip mroute**	Displays information about the multicast routing table.

This example shows how to display information about the PIM group and rendezvous point mappings:

```
Router# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent
Group(s) 230.31.0.0/16
  RP 60.0.0.60 (?), v2v1, bidir
    Info source:60.0.0.60 (?), elected via Auto-RP
         Uptime:00:03:47, expires:00:02:11
  RP 50.0.0.50 (?), v2v1, bidir
    Info source:50.0.0.50 (?), via Auto-RP
         Uptime:00:03:04, expires:00:02:55
  RP 40.0.0.40 (?), v2v1, bidir
    Info source:40.0.0.40 (?), via Auto-RP
         Uptime:00:04:19, expires:00:02:38
```

This example shows how to display information in the IP multicast routing table that is related to IPv4 bidirectional PIM:

```
Router# show ip mroute bidirectional
    (*, 225.1.3.0), 00:00:02/00:02:57, RP 3.3.3.3, flags:BC
      Bidir-Upstream:GigabitEthernet2/1, RPF nbr 10.53.1.7, RPF-MFD
      Outgoing interface list:
        GigabitEthernet2/1, Bidir-Upstream/Sparse-Dense, 00:00:02/00:00:00,H
        Vlan30, Forward/Sparse-Dense, 00:00:02/00:02:57, H

    (*, 225.1.2.0), 00:00:04/00:02:55, RP 3.3.3.3, flags:BC
      Bidir-Upstream:GigabitEthernet2/1, RPF nbr 10.53.1.7, RPF-MFD
      Outgoing interface list:
        GigabitEthernet2/1, Bidir-Upstream/Sparse-Dense, 00:00:04/00:00:00,H
        Vlan30, Forward/Sparse-Dense, 00:00:04/00:02:55, H

    (*, 225.1.4.1), 00:00:00/00:02:59, RP 3.3.3.3, flags:BC
      Bidir-Upstream:GigabitEthernet2/1, RPF nbr 10.53.1.7, RPF-MFD
      Outgoing interface list:
        GigabitEthernet2/1, Bidir-Upstream/Sparse-Dense, 00:00:00/00:00:00,H
        Vlan30, Forward/Sparse-Dense, 00:00:00/00:02:59, H
```

This example show how to display information related to a specific multicast route. In the output below, the arrow in the margin points to information about a partical short cut:

```
Router# show ip mroute 239.1.1.2 4.4.4.4
IP Multicast Routing Table
Flags:D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags:H - Hardware switched
 Timers:Uptime/Expires
 Interface state:Interface, Next-Hop or VCD, State/Mode

(4.4.4.4, 239.1.1.2), 1d02h/00:03:20, flags:FTZ
 Incoming interface:Loopback0, RPF nbr 0.0.0.0, Partial-SC
  Outgoing interface list:
    Vlan10, Forward/Sparse-Dense, 1d02h/00:02:39 (ttl-threshold 5)
```

This example shows how to display the entries for a specific multicast group address:

```
Router# show mls ip multicast group 230.31.31.1
Multicast hardware switched flows:
(*, 230.31.31.1) Incoming interface:Vlan611, Packets switched:1778
Hardware switched outgoing interfaces:Vlan131 Vlan151 Vlan415 Gi4/16 Vlan611
RPF-MFD installed
```

This example shows how to display PIM group to active rendezvous points mappings:

```
Router# show mls ip multicast rp-mapping
State:H - Hardware Switched, I - Install Pending, D - Delete Pending, Z - Zombie

RP Address      State       RPF            DF-count    GM-count
60.0.0.60       H           Vl611                 4           1
```

This example shows how to display information based on the group/mask ranges in the RP mapping cache:

```
Router# show mls ip multicast rp-mapping gm-cache
State:H - Hardware Switched, I - Install Pending, D - Delete Pending,
       Z - Zombie

RP Address     State       Group          Mask            State         Packet/Byte-count
60.0.0.60      H           230.31.0.0     255.255.0.0     H             100/6400
```

This example shows how to display information about specific MLS IP multicasting groups:

```
Router# show mls ip multicast rp-mapping df-cache
State:H - Hardware Switched, I - Install Pending, D - Delete Pending, Z - Zombie

RP Address      State        DF             State
60.0.0.60       H           Vl131            H
60.0.0.60       H           Vl151            H
60.0.0.60       H           Vl415            H
60.0.0.60       H           Gi4/16           H
```

Using IPv4 Debug Commands

Table 37-2 describes IPv4 multicast Layer 3 switching debug commands that you can use to troubleshoot IP multicast Layer 3 switching problems.

Table 37-2 IP Multicast Layer 3 Switching Debug Commands

Command	Description
[**no**] **debug mls ip multicast events**	Displays IP multicast Layer 3 switching events.
[**no**] **debug mls ip multicast errors**	Turns on debug messages for multicast MLS-related errors.
[**no**] **debug mls ip multicast group** *group_id group_mask*	Turns on debugging for a subset of flows.
[**no**] **debug mls ip multicast messages**	Displays IP multicast Layer 3 switching messages from and to hardware switching engine.
[**no**] **debug mls ip multicast all**	Turns on all IP multicast Layer 3 switching messages.
[**no**] **debug mdss errors**	Turns on MDSS[1] error messages.
[**no**] **debug mdss events**	Displays MDSS-related events for debugging.
[**no**] **debug mdss events mroute-bidir**	Displays IPv4 bidirectional PIM MDSS events for debugging.
[**no**] **debug mdss all**	Displays all MDSS messages.
[**no**] **debug ip pim df** *ip_address*	Displays the DF election for a given rendezvous point for debug purposes.

1. MDSS = Multicast Distributed Switching Services

Clearing IPv4 Multicast Layer 3 Switching Statistics

To clear IP multicast Layer 3 switching statistics, perform this task:

Command	Purpose
Router# **clear mls ip multicast statistics**	Clears IP multicast Layer 3 switching statistics.

This example shows how to clear IP multicast Layer 3 switching statistics:

```
Router# clear mls ip multicast statistics
```

The **show mls multicast statistics** command displays a variety of information about the multicast flows being handled by the PFC. You can display entries based on any combination of the participating RP, the VLAN, the multicast group address, or the multicast traffic source. For an example of the **show mls ip multicast statistics** command, see the "Displaying IPv4 Multicast Layer 3 Switching Statistics" section on page 37-23.

Redundancy for Multicast Traffic

Redundancy for multicast traffic requires the following conditions:

- Unicast routing protocol such as OSPF or EIGRP

  PIM uses RPF checks on the unicast routing table to determine the proper paths for multicast data to traverse. If a unicast routing path changes, PIM relies upon the unicast routing protocol (OSPF) to properly converge, so that the RPF checks used by PIM continue to work and show valid unicast paths to and from the source IP address of the server sourcing the multicast stream.

- PIM configured on all related Layer 3 interfaces

  The unicast routing table is used to do path selection for PIM. PIM uses RPF checks to ultimately determine the shortest path tree (SPT) between the client (receiver VLAN) and the source (multicast VLAN). Therefore, the objective of PIM is to find the shortest unicast path between the receiver subnet and the source subnet. You do not need to configure anything else for multicast when the unicast routing protocol is working as expected and PIM is configured on all the Layer 3 links associated with the unicast routing protocol.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring IGMP Snooping for IPv4 Multicast Traffic

This chapter describes how to configure Internet Group Management Protocol (IGMP) snooping in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- To constrain IPv6 Multicast traffic, see Chapter 36, "Configuring MLD Snooping for IPv6 Multicast Traffic."

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding IGMP Snooping

These sections describe IGMP snooping:

IGMP Snooping Overview

IGMP snooping allows switches to examine IGMP packets and make forwarding decisions based on their content. You can configure the switch to use IGMP snooping in subnets that receive IGMP queries from either IGMP or the IGMP snooping querier. IGMP snooping constrains IPv4 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv4 multicast traffic only to those ports that want to receive it. With Release 12.2(33)SXJ2 and later releases, IGMP snooping also constrains multicast traffic to VPLS interfaces.

IGMP, which runs at Layer 3 on a multicast router, generates Layer 3 IGMP queries in subnets where the multicast traffic needs to be routed. For information about IGMP, see Chapter 37, “Configuring IPv4 Multicast Layer 3 Switching.”

You can configure the IGMP snooping querier on the switch to support IGMP snooping in subnets that do not have any multicast router interfaces. For more information about the IGMP snooping querier, see the “Configuring the IGMP Snooping Querier” section on page 38-10.

IGMP (on a multicast router) or, locally, the IGMP snooping querier, sends out periodic general IGMP queries that the switch forwards through all ports in the VLAN and to which hosts respond. IGMP snooping monitors the Layer 3 IGMP traffic.

Note If a multicast group has only sources and no receivers in a VLAN, IGMP snooping constrains the multicast traffic to only the multicast router ports.

Joining a Multicast Group

Hosts join multicast groups either by sending an unsolicited IGMP join message or by sending an IGMP join message in response to a general query from a multicast router (the switch forwards general queries from multicast routers to all ports in a VLAN).

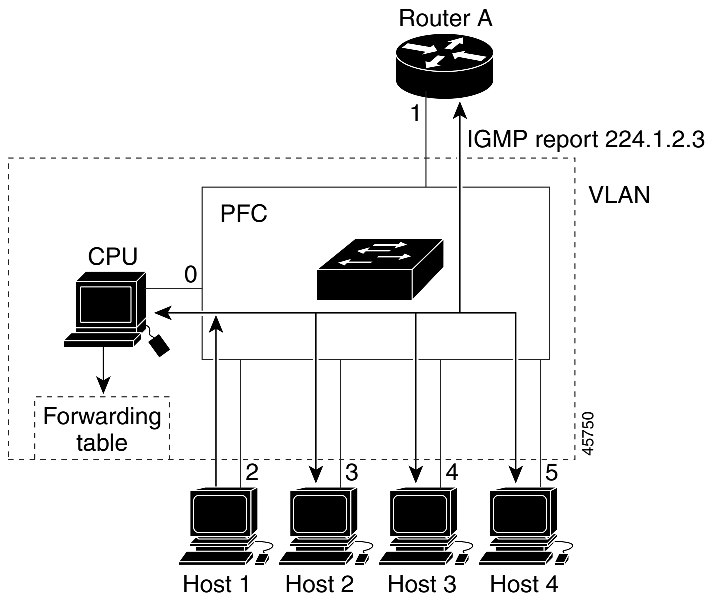
In response to an IGMP join request, the switch creates an entry in its Layer 2 forwarding table for the VLAN on which the join request was received. When other hosts that are interested in this multicast traffic send IGMP join requests, the switch adds them to the existing Layer 2 forwarding table entry. The switch creates only one entry per VLAN in the Layer 2 forwarding table for each multicast group for which it receives an IGMP join request.

IGMP snooping suppresses all but one of the host join messages per multicast group and forwards this one join message to the multicast router.

The switch forwards multicast traffic for the multicast group specified in the join message to the interfaces where join messages were received (see Figure 38-1).

Layer 2 multicast groups learned through IGMP snooping are dynamic. However, you can statically configure Layer 2 multicast groups using the **mac-address-table static** command. When you specify group membership for a multicast group address statically, the static setting supersedes any IGMP snooping learning. Multicast group membership lists can consist of both static and IGMP snooping-learned settings.

Figure 38-1 Initial IGMP Join Message



Multicast router A sends a general query to the switch, which forwards the query to ports 2 through 5 (all members of the same VLAN). Host 1 wants to join multicast group 224.1.2.3 and multicasts an IGMP membership report (IGMP join message) to the group with the equivalent MAC destination address of 0x0100.5E01.0203. When the CPU receives the IGMP report multicast by Host 1, the CPU uses the information in the IGMP report to set up a forwarding-table entry, as shown in Table 38-1, that includes the port numbers of Host 1, the multicast router, and the switch internal CPU.

Table 38-1 IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1, 2

The switch hardware can distinguish IGMP information packets from other packets for the multicast group. The first entry in the table tells the switching engine to send only IGMP packets to the CPU. This prevents the CPU from becoming overloaded with multicast frames. The second entry tells the switching engine to send frames addressed to the 0x0100.5E01.0203 multicast MAC address that are not IGMP packets (!IGMP) to the multicast router and to the host that has joined the group.

If another host (for example, Host 4) sends an unsolicited IGMP join message for the same group (Figure 38-2), the CPU receives that message and adds the port number of Host 4 to the forwarding table as shown in Table 38-2. Because the forwarding table directs IGMP messages only to the CPU, the message is not flooded to other ports. Any known multicast traffic is forwarded to the group and not to the CPU.

Figure 38-2 Second Host Joining a Multicast Group

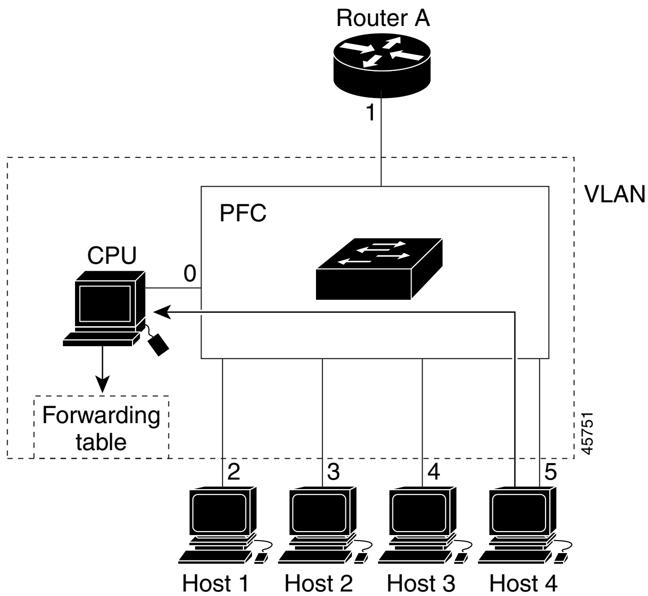


Table 38-2 Updated IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1, 2, 5

Leaving a Multicast Group

These sections describe leaving a multicast group:

- Normal Leave Processing, page 38-4
- Fast-Leave Processing, page 38-5

Normal Leave Processing

Interested hosts must continue to respond to the periodic general IGMP queries. As long as at least one host in the VLAN responds to the periodic general IGMP queries, the multicast router continues forwarding the multicast traffic to the VLAN. When hosts want to leave a multicast group, they can either ignore the periodic general IGMP queries (called a “silent leave”), or they can send a group-specific IGMPv2 leave message.

When IGMP snooping receives a group-specific IGMPv2 leave message from a host, it sends out a MAC-based general query to determine if any other devices connected to that interface are interested in traffic for the specific multicast group. If IGMP snooping does not receive an IGMP Join message in response to the general query, it assumes that no other devices connected to the interface are interested in receiving traffic for this multicast group, and it removes the interface from its Layer 2 forwarding table entry for that multicast group. If the leave message was from the only remaining interface with hosts interested in the group and IGMP snooping does not receive an IGMP Join in response to the general query, it removes the group entry and relays the IGMP leave to the multicast router. If the multicast router receives no reports from a VLAN, the multicast router removes the group for the VLAN from its IGMP cache.

The interval for which the switch waits before updating the table entry is called the "last member query interval," (See the "Configuring the IGMP Snooping Querier" section on page 38-10.)

Fast-Leave Processing

IGMP snooping fast-leave processing allows IGMP snooping to remove a Layer 2 LAN interface from the forwarding-table entry without first sending out IGMP group-specific queries to the interface. Upon receiving a group-specific IGMPv2 leave message, IGMP snooping immediately removes the interface from the Layer 2 forwarding table entry for that multicast group, unless a multicast router was learned on the port. Fast-leave processing improves bandwidth management for all hosts on a switched network.

Note Use fast-leave processing only on VLANs where only one host is connected to each Layer 2 LAN port. If fast-leave is enabled in VLANs where more than one host is connected to a Layer 2 LAN port, some hosts might be dropped inadvertently. Fast-leave processing is supported only with IGMP version 2 hosts.

Understanding the IGMP Snooping Querier

Use the IGMP snooping querier to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not need to be routed.

In a network where IP multicast routing is configured, the IP multicast router acts as the IGMP querier. If the IP-multicast traffic in a VLAN only needs to be Layer 2 switched, an IP-multicast router is not required, but without an IP-multicast router on the VLAN, you must configure another switch as the IGMP querier so that it can send queries.

When enabled, the IGMP snooping querier sends out periodic IGMPv3 queries that trigger IGMP report messages from the switch that wants to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

Configure one switch as the IGMP snooping querier in each VLAN that is supported on switches that use IGMP to report interest in IP multicast traffic.

Note Enable the IGMP snooping querier on only one switch in the VLAN.

You can configure a switch to generate IGMP queries on a VLAN regardless of whether or not IP multicast routing is enabled (see the "Configuring the IGMP Snooping Querier" section on page 38-10).

Understanding IGMP Version 3 Support

These sections describe IGMP version 3 support:

IGMP Version 3 Support Overview

IGMP snooping supports IGMP version 3 (IGMPv3). IGMPv3 uses source-based filtering, which enables hosts and routers to specify which source addresses should be allowed or blocked for a specific multicast group. When you enable IGMPv3 snooping, the switch maintains IGMPv3 states based on messages it receives for a particular group in a particular VLAN and either allows or blocks traffic based on the following information in these messages:

- Source lists
- Allow (include) or block (exclude) filtering options

Because the Layer 2 table is (MAC-group, VLAN) based, with IGMPv3 hosts it is preferable to have only a single multicast source per MAC-group.

Note Source-based filtering for IGMPv3 reports is not supported in hardware. The states are maintained only in software and used for explicit host tracking and statistics collection. The source-only entries are deleted every 5 minutes and relearned to ensure that they are still valid.

IGMPv3 Fast-Leave Processing

IGMPv3 fast-leave processing is active if explicit-host tracking is enabled. The **ip igmp snooping fast-leave** command that enables IGMP version 2 fast-leave processing does not affect IGMPv3 fast-leave processing.

Fast-leave processing with IGMPv3 is implemented by maintaining source-group based membership information in software while also allocating LTL indexes on a MAC GDA basis.

When fast-leave processing is active, hosts send BLOCK_OLD_SOURCES{src-list} messages for a specific group when they no longer want to receive traffic from that source. When the switch receives such a message from a host, it parses the list of sources for that host for the given group. If this source list is exactly the same as the source list received in the leave message, the switch removes the host from the LTL index and stops forwarding this multicast group traffic to this host.

If the source lists do not match, the switch does not remove the host from the LTL index until the host is no longer interested in receiving traffic from any source.

Proxy Reporting

IGMP supports proxy reporting for IGMPv1 and IGMPv2 messages to handle group-specific queries. These queries are not sent downstream, but the switch does respond to them directly. When the switch receives a group-specific query, the switch terminates the query and sends an IGMP proxy report if there is a receiver for the group. There is no proxy reporting for IGMPv3 messages. For IGMPv3, a group-specific query or a group source-specific query is flooded to all VLAN member ports. The database for the IGMPv3 membership report is built based on the reports received.

Host reports responding to a specific query can be suppressed by the report suppression feature. Report suppression is supported for IGMPv1, IGMPv2, and IGMPv3 messages. With report suppression enabled (by default), when the switch receives a general query, the switch starts a suppression cycle for reports from all hosts to each group or channel (S,G). Only the first report to the discovered multicast routers is forwarded; the rest of the reports are suppressed. For IGMPv1 and IGMPv2, the time of suppression is the report response time indicated in the general query message. For IGMPv3, suppression occurs for the entire general query interval.

Note
- Source-based filtering for IGMP version 3 reports is not supported in hardware. The states are maintained only in software and used for explicit host tracking and statistics collection. The source-only entries are deleted every 5 minutes and relearned to ensure that they are still valid.
- Turning off explicit host tracking disables fast-leave processing and proxy reporting.

Explicit Host Tracking

IGMPv3 supports explicit tracking of membership information on any port. The explicit-tracking database is used for fast-leave processing for IGMPv3 hosts, proxy reporting, and statistics collection. When explicit tracking is enabled on a VLAN, the IGMP snooping software processes the IGMPv3 report it receives from a host and builds an explicit-tracking database that contains the following information:

- The port connected to the host
- The channels reported by the host
- The filter mode for each group reported by the host
- The list of sources for each group reported by the hosts
- The router filter mode of each group
- For each group, the list of hosts requesting the source

Note
- Turning off explicit host tracking disables fast-leave processing and proxy reporting.
- When explicit tracking is enabled and the switch is working in proxy-reporting mode, the router may not be able to track all the hosts behind a VLAN interface.

Default IGMP Snooping Configuration

Table 38-3 shows the default IGMP snooping configuration.

Table 38-3 IGMP Snooping Default Configuration

Feature	Default Values
IGMP snooping querier	Disabled
IGMP snooping	Enabled
Multicast routers	None configured
IGMPv3 proxy reporting	Enabled
IGMP snooping router learning method	Learned automatically through PIM or IGMP packets
Fast-Leave Processing	Disabled

Table 38-3 IGMP Snooping Default Configuration (continued)

Feature	Default Values
CGMP Automatic Detection	Enabled
IGMPv3 Explicit Host Tracking	Enabled

IGMP Snooping Configuration Guidelines and Restrictions

When configuring IGMP snooping, follow these guidelines and restrictions:

- To support Cisco Group Management Protocol (CGMP) client devices, configure the route processor (RP) as a CGMP server. See the *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.2, "IP Multicast," "Configuring IP Multicast Routing," at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfmulti.html
- For more information on IP multicast and IGMP, see RFC 1112 and RFC 2236.
- IGMP snooping supports private VLANs. Private VLANs do not impose any restrictions on IGMP snooping.
- IGMP snooping constrains traffic in MAC multicast groups 0100.5e00.0001 to 0100.5eff.ffff.
- IGMP snooping does not constrain Layer 2 multicasts generated by routing protocols.

IGMP Snooping Querier Configuration Guidelines and Restrictions

When configuring the IGMP snooping querier, follow these guidelines and restrictions:

- Release 12.2(33)SXJ1 and later releases support redundant IGMP snooping queriers. To configure redundant IGMP snooping queriers, ensure that the tasks in the "Enabling IGMP Snooping" section on page 38-9 and "Configuring the IGMP Snooping Querier" section on page 38-10 are completed on more than one switch in the VLAN.

  When multiple IGMP snooping queriers are enabled in a VLAN, the querier with the lowest IP address in the VLAN is elected as the active IGMP snooping querier.

  An IGMP snooping querier election occurs if the active IGMP snooping querier goes down or if there is an IP address change on any of the queriers.

  Note To avoid unnecessary active querier time outs, configure the **ip igmp snooping querier query-interval** command with the same value on all queriers in a VLAN.

  Releases earlier than Release 12.2(33)SXJ1 do not support redundant IGMP snooping queriers. Enable the IGMP snooping querier on only one switch in the VLAN. (CSCsk48795)
- Configure the VLAN in global configuration mode (see Chapter 23, "Configuring VLANs").
- Configure an IP address on the VLAN interface (see Chapter 30, "Configuring Layer 3 Interfaces"). When enabled, the IGMP snooping querier uses the IP address as the query source address.

- If there is no IP address configured on the VLAN interface, the IGMP snooping querier does not start. The IGMP snooping querier disables itself if the IP address is cleared. When enabled, the IGMP snooping querier restarts if you configure an IP address.
- The IGMP snooping querier sends IGMPv3 querier messages. Although the IGMP version of the querier messages is not configurable, the querier is compatible with IGMPv2 hosts.
- When enabled, the IGMP snooping querier starts immediately. If IGMP traffic from a multicast router, or from another IGMP snooping querier in the VLAN, is detected after the IGMP snooping querier has started, the querier will disable itself.
- QoS does not support IGMP packets when IGMP snooping is enabled.

Configuring IGMP Snooping

Note To use IGMP snooping, configure a Layer 3 interface in the subnet for multicast routing (see Chapter 37, "Configuring IPv4 Multicast Layer 3 Switching") or enable the IGMP snooping querier in the subnet (see the "Configuring the IGMP Snooping Querier" section on page 38-10).

These sections describe how to configure IGMP snooping:

Note Except for the **ip igmp snooping** command, all IGMP snooping commands are supported only on VLAN interfaces.

Enabling IGMP Snooping

To enable IGMP snooping globally, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip igmp snooping`**	Enables IGMP snooping.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.
Step 3	`Router#` **`show ip igmp interface vlan`** *`vlan_ID`* `|` **`include globally`**	Verifies the configuration.

This example shows how to enable IGMP snooping globally and verify the configuration:

```
Router(config)# ip igmp snooping
Router(config)# end
Router# show ip igmp interface vlan 200 | include globally
```

```
  IGMP snooping is globally enabled
Router#
```

To enable IGMP snooping in a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ip igmp snooping`	Enables IGMP snooping.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show ip igmp interface vlan vlan_ID | include snooping`	Verifies the configuration.

This example shows how to enable IGMP snooping on VLAN 25 and verify the configuration:

```
Router# interface vlan 25
Router(config-if)# ip igmp snooping
Router(config-if)# end
Router# show ip igmp interface v125 | include snooping
IGMP snooping is globally enabled
  IGMP snooping CGMP-AutoDetect is globally enabled
  IGMP snooping is enabled on this interface
  IGMP snooping fast-leave (for v2) is disabled
  IGMP snooping querier is enabled. Querier is 11.1.22.252 (this system)
  IGMP snooping explicit-tracking is enabled
  IGMP snooping last member query response interval is 25000 ms
  IGMP snooping report-suppression is disabled
  IGMP snooping query interval is 60000 ms• New output
  IGMP snooping querier version is 3• New output
  IGMP snooping querier - TCN query count is 5• New output
  IGMP snooping querier - TCN query interval is 1000 ms
Router#
```

Configuring the IGMP Snooping Querier

Enabling the IGMP Snooping Querier

Use the IGMP snooping querier to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not need to be routed. To enable the IGMP snooping querier in a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects the VLAN interface.
Step 2	`Router(config-if)# ip address ip_address subnet_mask`	Configures the IP address and IP subnet.

	Command	Purpose
Step 3	`Router(config-if)# ip igmp snooping querier`	Enables the IGMP snooping querier.
Step 4	`Router(config-if)# end`	Exits configuration mode.
Step 5	`Router# show ip igmp interface vlan vlan_ID | include querier`	Verifies the configuration.

This example shows how to enable the IGMP snooping querier on VLAN 200 and verify the configuration:

```
Router# interface vlan 200
Router(config-if)# ip address 11.1.22.60 255.255.255.248
Router(config-if)# igmp snooping querier
Router(config-if)# end
Router# show ip igmp interface vlan 200 | include Querier
IGMP snooping querier is enabled. Querier is 11.1.22.60 (this system)
```

Configuring the IGMP Snooping General Query Interval

In Release 12.2(33)SXJ1 and later releases, you can configure the interval for which the switch waits after sending a general query to determine if hosts are still interested in any multicast groups.

Note In releases earlier than Release 12.2(33)SXJ1, the general query interval is 60 seconds and is not configurable.

To configure the IGMP snooping general query interval, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ip igmp snooping querier query-interval interval`	Configures the IGMP snooping general query interval. • Default value: 60000ms (60s). • The valid range is 1000 to 18000000 milliseconds. • With redundant IGMP snooping queriers, to avoid unnecessary active querier time outs, configure the same query-interval value on all queriers in a VLAN.

This example shows how to configure the IGMP snooping general query interval:

```
Router(config-if)# ip igmp snooping querier query-interval 60000
Router(config-if)# exit
Router# show ip igmp interface vlan 200 | include snooping query interval
  IGMP snooping query interval on this interface is 60000 ms
```

Configuring the IGMP Snooping TCN General Query Count

In Release 12.2(33)SXJ1 and later releases, you can configure the number of general queries that the IGMP snooping querier sends after receiving a topology change notification (TCN).

Note In releases earlier than Release 12.2(33)SXJ1, the TCN general query count is 1 and is not configurable.

To configure the TCN general query count, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface vlan`** *`vlan_ID`*	Selects a VLAN interface.
Step 2	`Router(config-if)#` **`ip igmp snooping querier tcn query count`** *`count`*	Configures the TCN general query count. • Default value: 2. • The valid range is 1 to 10 queries.

This example shows how to configure the TCN general query count:

```
Router(config-if)# ip igmp snooping querier tcn query count 2
Router(config-if)# exit
Router# show ip igmp interface vlan 200 | include TCN query count
  IGMP snooping querier - TCN query count is 2
```

Configuring the IGMP Snooping TCN General Query Interval

In Release 12.2(33)SXJ1 and later releases, you can configure the interval between general queries sent in response to TCNs.

Note In releases earlier than Release 12.2(33)SXJ1, the TCN general query interval is not applicable because the TCN general query count is 1 and is not configurable.

To configure the TCN general query interval, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface vlan`** *`vlan_ID`*	Selects a VLAN interface.
Step 2	`Router(config-if)#` **`ip igmp snooping querier tcn query interval`** *`interval`*	Configures the TCN general query interval. • Default value: 10000 ms (10s). • The valid range is 1000 to 255000 ms.

This example shows how to configure the TCN general query interval:

```
Router(config-if)# ip igmp snooping querier tcn query interval 10000
Router(config-if)# exit
Router# show ip igmp interface vlan 200 | include TCN query interval
  IGMP snooping querier - TCN query interval is 10000 ms
```

Configuring the IGMP Snooping Group-Specific Query Interval

You can configure the interval for which the switch waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group.

Note When both IGMP fast-leave processing and the IGMP query interval are configured, fast-leave processing takes precedence.

To configure the IGMP snooping group-specific query interval, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ip igmp snooping last-member-query-interval interval`	Configures the IGMP snooping group-specific query interval. • The default value is 1000 ms (1s). • In Release 12.2(33)SXJ1 and later releases, the valid range is 100 to 25500 milliseconds. • In releases earlier than Release 12.2(33)SXJ1, the valid range is 100 to 999 milliseconds.

This example shows how to configure the IGMP snooping query interval:

```
Router(config-if)# ip igmp snooping last-member-query-interval 200
Router(config-if)# exit
Router# show ip igmp interface vlan 200 | include last
  IGMP snooping last member query interval on this interface is 200 ms
```

Enabling IGMP Fast-Leave Processing

Fast-leave configuration applies to IGMP version 2 hosts only. To enable IGMP fast-leave processing in a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ip igmp snooping`	Enables IGMP snooping. This step is only necessary if IGMP snooping is not already enabled on this VLAN.
Step 3	`Router(config-if)# ip igmp snooping fast-leave`	Enables IGMP fast-leave processing in the VLAN.

This example shows how to enable IGMP fast-leave processing for IGMP version 2 hosts on the VLAN 200 interface, and how to verify the configuration:

```
Router# interface vlan 200
Router(config-if)# ip igmp snooping
Router(config-if)# ip igmp snooping fast-leave
Configuring fast leave on vlan 200
Router(config-if)# end
Router# show ip igmp interface vlan 200 | include fast-leave
IGMP snooping fast-leave is enabled on this interface
```

Configuring Source-Specific Multicast Mapping

Note Do not configure SSM mapping in a VLAN that supports IGMPv3 multicast receivers.

To configure source-specific multicast (SSM) mapping, see this publication:

http://www.cisco.com/en/US/docs/ios/ipmulti/configuration/guide/imc_basic_cfg.html#SSM_Overview

CGMP Automatic Detection

By default, the switch will detect Cisco group management protocol (CGMP) packets using the CGMP automatic detection feature. CGMP automatic detection operates as follows:

- When CGMP traffic is detected on a VLAN, IGMP report suppression is disabled on that VLAN for a period of five minutes.
- Any new CGMP traffic on the VLAN will begin a new five-minute period.
- When no new CGMP traffic has been detected on the VLAN for five minutes, the IGMP report suppression will revert to the configured status.

The CGMP automatic detection feature has no access to VTP information and causes the switch to send CGMP traffic to VLANs that VTP has pruned from trunks. To avoid this situation, you can disable the CGMP automatic detection feature by entering the **no ip igmp snooping cgmp auto-detect** global configuration command. Disabling CGMP automatic detection restricts CGMP traffic to Layer 2. When CGMP automatic detection is disabled, IGMP report suppression must be disabled manually for any VLAN that will use CGMP.

To disable CGMP automatic detection, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`no ip igmp snooping cgmp auto-detect`**	Disables the CGMP auto-detect mode globally.
Step 2	`Router(config)#` **`interface vlan`** *`vlan_ID`*	Selects a VLAN interface.
Step 3	`Router(config-if)#` **`no ip igmp snooping report-suppression`**	Disables IGMP snooping report suppression so that CGMP receives all the report messages on this VLAN.
Step 4	`Router(config-if)#` **`ip cgmp`**	Enables CGMP mode on this VLAN.

Configuring IGMPv3 Explicit Host Tracking

To enable explicit host tracking on a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ip igmp explicit-tracking`	Enables explicit host tracking.
Step 3	`Router# show ip igmp snooping explicit-tracking {vlan vlan-id}`	Displays information about the explicit host tracking status for IGMPv3 hosts.

This example shows how to enable explicit host tracking:

```
Router(config)# interface vlan 25
Router(config-if)# ip igmp explicit-tracking
Router(config-if)# end
```

This example shows how to display information about explicit host tracking:

```
Router# show ip igmp snooping explicit-tracking vlan 25
Source/Group                    Interface    Reporter        Filter_mode
------------------------------------------------------------------------
10.1.1.1/226.2.2.2              Vl25:1/2     16.27.2.3       INCLUDE
10.2.2.2/226.2.2.2              Vl25:1/2     16.27.2.3       INCLUDE
```

This example shows the information displayed for VPLS interfaces:

```
Router# show ip igmp snooping explicit-tracking vlan 25
Source/Group                    Interface      Reporter      Filter_mode
------------------------------------------------------------------------
10.1.1.1/224.1.1.1              Vl25:VPLS3/62   2.2.2.1       INCLUDE
10.2.2.2/224.2.2.1              Vl25:A-VPLS14/0 2.2.2.2       INCLUDE
```

Displaying IGMP Snooping Information

These sections describe displaying IGMP snooping information:

- Displaying Multicast Router Interfaces, page 38-15
- Displaying MAC Address Multicast Entries, page 38-16
- Displaying IGMP Snooping Information for a VLAN Interface, page 38-16
- Displaying IGMP Snooping Statistics, page 38-17

Displaying Multicast Router Interfaces

When you enable IGMP snooping, the switch automatically learns to which interface the multicast routers are connected.

To display multicast router interfaces, perform this task:

Command	Purpose
`Router# show ip igmp snooping mrouter vlan_ID`	Displays multicast router interfaces.

This example shows how to display the multicast router interfaces in VLAN 3:

```
Router# show ip igmp snooping mrouter vlan 3
vlan            ports
-----+----------------------------------------
   3  Router,VPLS3/62,A-VPLS14/0
Router#
```

Displaying MAC Address Multicast Entries

To display MAC address multicast entries for a VLAN, perform this task:

Command	Purpose
Router# **show mac-address-table multicast** *vlan_ID* [**count**]	Displays MAC address multicast entries for a VLAN.

This example shows how to display MAC address multicast entries for VLAN 3:

```
Router# show mac-address-table multicast vlan 3
vlan   mac address     type   learn qos             ports
-----+---------------+--------+-----+---+--------------------------------
   3  0100.5e01.0101    static  Yes          -   Router,VPLS 2.2.2.2,A-VPLS 7.7.7.7
   3  0100.5e02.0201    static  Yes          -   Router,VPLS 2.2.2.2,A-VPLS 7.7.7.7
Router#
```

This example shows how to display a total count of MAC address entries for a VLAN:

```
Router# show mac-address-table multicast 3 count

Multicast MAC Entries for vlan 3:    2
Router#
```

Displaying IGMP Snooping Information for a VLAN Interface

Note When you apply the **ip igmp snooping** command and associated commands on any VLAN interface, the commands continue to function even if the VLAN interface is in shutdown state.

To display IGMP snooping information for a VLAN interface, perform this task:

Command	Purpose
Router# **show ip igmp interface** *vlan_ID*	Displays IGMP snooping information on a VLAN interface.

This example shows how to display IGMP snooping information on the VLAN 200 interface:

```
Router# show ip igmp interface vlan 43
Vlan43 is up, line protocol is up
  Internet address is 43.0.0.1/24
  IGMP is enabled on interface
  Current IGMP host version is 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds
```

```
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query count is 2
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity:1 joins, 0 leaves
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 43.0.0.1 (this system)
  IGMP querying router is 43.0.0.1 (this system)
  Multicast groups joined by this system (number of users):
      224.0.1.40(1)
  IGMP snooping is globally enabled
  IGMP snooping is enabled on this interface
  IGMP snooping fast-leave is disabled and querier is disabled
  IGMP snooping explicit-tracking is enabled on this interface
  IGMP snooping last member query interval on this interface is 1000 ms
Router#
```

Displaying IGMP Snooping Statistics

The **show ip igmp snooping statistics interface** *vlan_ID* command displays the following information:

- The list of ports that are members of a group
- The filter mode
- The reporter-address behind the port
- The last-join and last-leave information collected since the last time a **clear ip igmp snooping statistics** command was entered

To display IGMP snooping statistics, perform this task:

Command	Purpose
Router# **show ip igmp snooping statistics interface** *vlan_ID*	Displays IGMP snooping information on a VLAN interface.

This example shows IGMP snooping statistics information for interface VLAN 25:

```
Router# show ip igmp snooping statistics interface vlan 25

Snooping statistics for Vlan25
#channels: 2
#hosts   : 2
Source/Group            Interface        Reporter       Uptime         Last-Join   Last-Leave
10.1.1.1/224.1.1.1      Vl25:VPLS3/62     2.2.2.1        00:01:47       00:00:51       -
10.2.2.2/224.2.2.1      Vl25:A-VPLS14/0   2.2.2.2        00:01:50       00:00:52       -
Router#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring MVR for IPv4 Multicast Traffic

This chapter describes how to configure Multicast VLAN Registration (MVR) for IPv4 multicast traffic in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- To constrain IPv6 Multicast traffic, see Chapter 36, "Configuring MLD Snooping for IPv6 Multicast Traffic."

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding MVR

Release 12.2(33)SXH and later releases support Multicast VLAN Registration (MVR). MVR is designed for applications that use wide-scale deployment of multicast traffic across an Ethernet ring-based service-provider network (for example, the broadcast of multiple television channels over a service-provider network). MVR allows a subscriber on a port to subscribe and unsubscribe to a multicast stream on the network-wide multicast VLAN. It allows the single multicast VLAN to be shared in the network while subscribers remain in separate VLANs. MVR provides the ability to continuously send multicast streams in the multicast VLAN, but to isolate the streams from the subscriber VLANs for bandwidth and security reasons.

MVR assumes that subscriber ports subscribe and unsubscribe (join and leave) these multicast streams by sending out IGMP join and leave messages. These messages can originate from an IGMP Version-2-compatible host with an Ethernet connection. Although MVR operates on the underlying mechanism of IGMP snooping, the two features operate independently of each other. One feature can be enabled or disabled without affecting the operation of the other feature. However, if IGMP snooping and MVR are both enabled, MVR reacts only to join and leave messages from multicast groups configured under MVR. Join and leave messages from all other multicast groups are managed by IGMP snooping.

MVR does the following:

- Identifies the MVR IP multicast streams and their associated IP multicast group in the Layer 2 forwarding table.
- Intercepts the IGMP messages.
- Modifies the Layer 2 forwarding table to include or remove the subscriber as a receiver of the multicast stream, even though the receivers might be in a different VLAN from the source.

This forwarding behavior selectively allows traffic to cross between different VLANs.

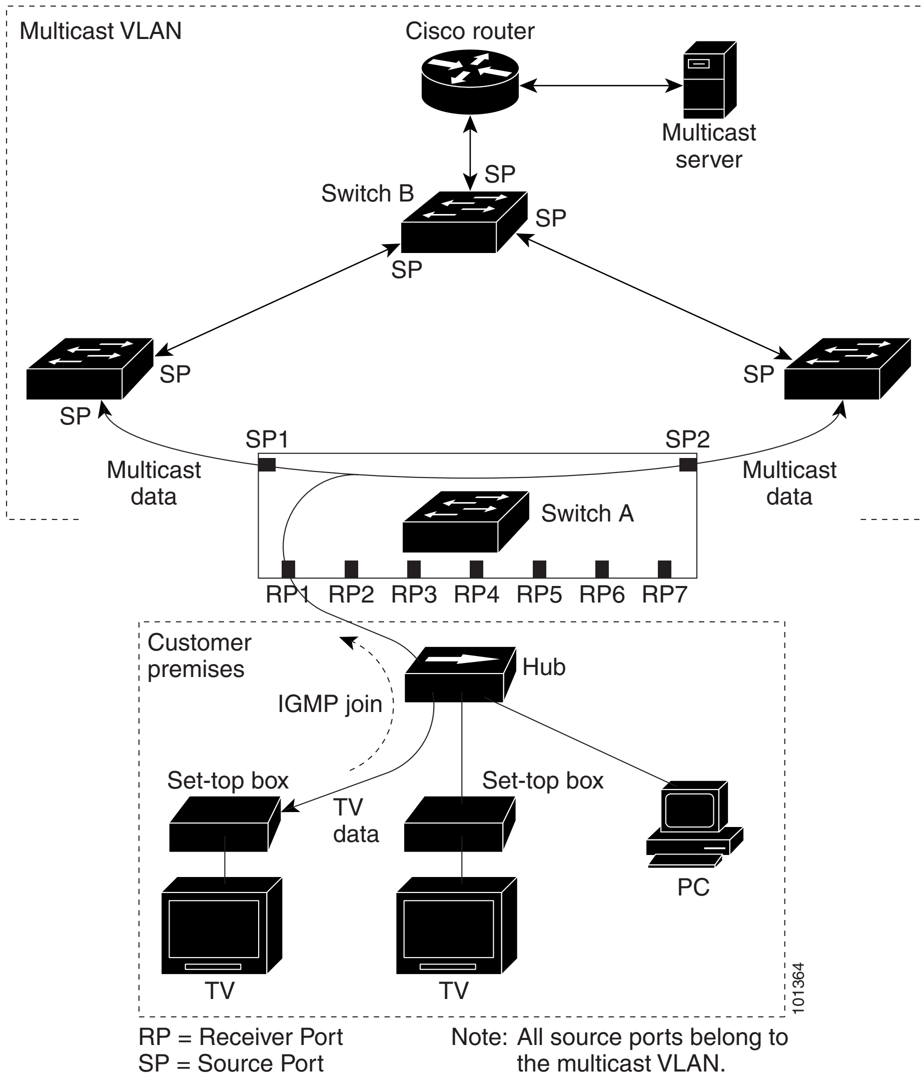
The switch will forward multicast data for MVR IP multicast streams only to MVR ports on which hosts have joined, either by IGMP reports or by MVR static configuration. The switch will forward IGMP reports received from MVR hosts only to the source (uplink) port. This eliminates using unnecessary bandwidth on MVR data port links.

Only Layer 2 ports participate in MVR. You must configure ports as MVR receiver ports. Only one MVR multicast VLAN per switch.

Using MVR in a Multicast Television Application

In a multicast television application, a PC or a television with a set-top box can receive the multicast stream. Multiple set-top boxes or PCs can be connected to one subscriber port, which is a switch port configured as an MVR receiver port. Figure 39-1 is an example configuration. DHCP assigns an IP address to the set-top box or the PC. When a subscriber selects a channel, the set-top box or PC sends an IGMP report to Switch A to join the appropriate multicast. If the IGMP report matches one of the configured IP multicast group addresses, the SP CPU modifies the hardware address table to include this receiver port and VLAN as a forwarding destination of the specified multicast stream when it is received from the multicast VLAN. Uplink ports that send and receive multicast data to and from the multicast VLAN are called MVR source ports.

Figure 39-1 Multicast VLAN Registration Example



When a subscriber changes channels or turns off the television, the set-top box sends an IGMP leave message for the multicast stream. The SP CPU sends a MAC-based general query through the receiver port VLAN. If there is another set-top box in the VLAN still subscribing to this group, that set-top box must respond within the maximum response time specified in the query. If the CPU does not receive a response, it eliminates the receiver port as a forwarding destination for this group.

Unless the Immediate Leave feature is enabled, when the switch receives an IGMP leave message from a subscriber on a receiver port, it sends out an IGMP query on that port and waits for IGMP group membership reports. If no reports are received in a configured time period, the receiver port is removed from multicast group membership. With the Immediate Leave feature enabled, an IGMP query is not sent from the receiver port on which the IGMP leave was received. As soon as the leave message is received, the receiver port is removed from multicast group membership, which speeds up leave latency. Enable the Immediate Leave feature only on receiver ports to which a single receiver device is connected.

MVR eliminates the need to duplicate television-channel multicast traffic for subscribers in each VLAN. Multicast traffic for all channels is only sent around the VLAN trunk once—only on the multicast VLAN. The IGMP leave and join messages are in the VLAN to which the subscriber port is assigned.

These messages dynamically register for streams of multicast traffic in the multicast VLAN on the Layer 3 device, Switch B. The access layer switch, Switch A, modifies the forwarding behavior to allow the traffic to be forwarded from the multicast VLAN to the subscriber port in a different VLAN, selectively allowing traffic to cross between two VLANs.

IGMP reports are sent to the same IP multicast group address as the multicast data. The Switch A CPU must capture all IGMP join and leave messages from receiver ports and forward them to the multicast VLAN of the source (uplink) port.

Configuring MVR

These sections contain this configuration information:

Default MVR Configuration

Table 39-1 shows the default MVR configuration.

Table 39-1 Default MVR Configuration

Feature	Default Setting
MVR	Disabled globally and per interface
Multicast addresses	None configured
Query response time	1 second
Multicast VLAN	VLAN 1
Interface (per port) default	Neither a receiver nor a source port
Immediate Leave	Disabled on all ports

MVR Configuration Guidelines and Limitations

When configuring MVR, follow these guidelines:

- Only one MVR VLAN can be present in a switch, and you should configure the same VLAN as the MVR VLAN for all the switches in the same network.
- Source ports must be in the MVR VLAN.
- Receiver ports on a switch can be in different VLANs, but must not be in the MVR VLAN.
- Receiver ports can only be access ports; they cannot be trunk ports.
- When using private VLANs, you cannot configure a secondary VLAN as the MVR VLAN.
- Do not connect a multicast router to a receiver port.
- The MVR VLAN must not be a reverse path forwarding (RPF) interface for any multicast route.
- MVR data received on an MVR receiver port is not forwarded to MVR source ports.
- The maximum number of multicast entries (MVR group addresses) that can be configured on a switch (that is, the maximum number of television channels that can be received) is 8000.
- MVR is available only on native systems.
- VTP pruning should be disabled if the MVR VLAN number is between 1 and 1000.
- MVR can coexist with IGMP snooping on a switch.
- MVR supports IGMPv3 messages.

Configuring MVR Global Parameters

You do not need to set the optional MVR parameters if you choose to use the default settings. Before changing the default parameters (except for the MVR VLAN), you must first enable MVR.

Note For complete syntax and usage information for the commands used in this section, see the Cisco IOS Master Command List.

To configure the MVR global parameters, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`mvr`**	Enables MVR on the switch.
Step 3	`Router(config)#` **`mvr max-groups`** *`max-groups`*	Specifies the maximum number of MVR groups. The range is 1 to 8000. The default is 1000.
Step 4	`Router(config)#` **`mvr group`** *`ip-address`* `[`*`count`*`]`	Configures an IP multicast address on the switch or uses the *count* parameter to configure a contiguous series of MVR group addresses (the range for *count* is 1 to 256; the default is 1). Any multicast data sent to this address is sent to all source ports on the switch and all receiver ports that have elected to receive data on that multicast address. Each multicast address would correspond to one television channel.

	Command	Purpose
Step 5	`Router(config)# mvr querytime value`	(Optional) Defines the maximum time to wait for IGMP report memberships on a receiver port before removing the port from multicast group membership. The value is in units of tenths of a second. The range is 1 to 100, and the default is 10 tenths or one second.
Step 6	`Router(config)# mvr vlan vlan-id`	(Optional) Specifies the VLAN in which multicast data is received; all source ports must belong to this VLAN. The VLAN range is 1 to 1001 and 1006 to 4094. The default is VLAN 1.
Step 7	`Router(config)# end`	Returns to privileged EXEC mode.
Step 8	`Router# show mvr`	Verifies the configuration.

To return the switch to its default settings, use the **no mvr** [**group** *ip-address* | **querytime** | **vlan**] global configuration command.

This example shows how to enable MVR, configure the group address, set the query time to 1 second (10 tenths), and specify the MVR multicast VLAN as VLAN 22:

```
Router(config)# mvr
Router(config)# mvr group 228.1.23.4
Router(config)# mvr querytime 10
Router(config)# mvr vlan 22
Router(config)# end
```

You can use the **show mvr groups** privileged EXEC command to verify the MVR multicast group addresses on the switch.

Configuring MVR Interfaces

To configure Layer 2 MVR interfaces, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# mvr`	Enables MVR on the switch.
Step 3	`Router(config)# interface interface-id`	Specifies the Layer 2 port to configure, and enters interface configuration mode.

	Command	Purpose
Step 4	`Router(config-if)# mvr type {source | receiver}`	Configures an MVR port as one of these types of ports: • **source**—Configures uplink ports that receive and send multicast data as source ports. Subscribers cannot be directly connected to source ports. All source ports on a switch belong to the single multicast VLAN. • **receiver**—Configures a port as a receiver port if it is a subscriber port and should only receive multicast data. It does not receive data unless it becomes a member of the multicast group, either statically or by using IGMP leave and join messages. Receiver ports cannot belong to the multicast VLAN. If you attempt to configure a non-MVR port with MVR characteristics, the operation fails. The default configuration is as a non-MVR port.
Step 5	`Router(config-if)# mvr immediate`	(Optional) Enables the Immediate Leave feature of MVR on the port. The Immediate Leave feature is disabled by default. **Note** This command applies to only receiver ports and should only be enabled on receiver ports to which a single receiver device is connected.
Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show mvr`	Verifies the configuration.

To return the interface to its default settings, use the **no mvr** [**type** | **immediate**] interface configuration commands.

This example shows how to configure a source port and a receiver port and to configure Immediate Leave on the receiver port:

```
Router(config)# mvr
Router(config)# interface gigabitethernet 3/48
Router(config-if)# switchport
Router(config-if)# switchport access vlan 22
Router(config-if)# mvr type source
Router(config-if)# exit
Router(config)# interface gigabitethernet 3/47
Router(config-if)# switchport
Router(config-if)# switchport access vlan 30
Router(config-if)# mvr type receiver
Router(config-if)# mvr immediate
Router(config-if)# exit
Router(config)#
```

Displaying MVR Information

You can display MVR information for the switch or for a specified interface. To display MVR configurations, perform one or more of these tasks:

Command	Purpose
Router# **show mvr**	Displays MVR status and these values for the switch: whether MVR is enabled or disabled, the multicast VLAN, the configured maximum and current number of multicast groups, and the query response time.
Router# **show mvr groups**	Displays the MVR group configuration.
Router# **show mvr interface** [*type module/port*]	Displays all MVR interfaces and their MVR configurations. When a specific interface is entered, displays this information: • Type—Receiver or Source • Status—One of these: – Active—At least one IGMP join has been received for an MVR group on the port. – Inactive—The port is not participating in any MVR groups. – Up/Down—The port is forwarding (Up) or nonforwarding (Down). • Immediate Leave—Enabled or Disabled
Router# **show mvr members** [[**vlan** *vlan-id*] | [*type module/port*]]	Displays details of all MVR members or MVR members on a specified VLAN or port.
Router# **show mvr members** [[**vlan** *vlan-id*] | [*type module/port*]] **count**	Displays number of MVR members in all active MVR groups. or on a specified VLAN or port.
Router# **show mvr** {**receiver-ports** | **source-ports**} [*type module/port*]	Displays all receiver or source ports that are members of any IP multicast group or those on the specified interface port.

This example displays MVR status and values for the switch:

```
Router# show mvr
MVR Running: TRUE
MVR multicast vlan: 22
MVR Max Multicast Groups: 1000
MVR Current multicast groups: 256
MVR Global query response time: 10 (tenths of sec)
```

This example displays the MVR group configuration:

```
Router# show mvr groups
MVR max Multicast Groups allowed: 8000
MVR current multicast groups: 8000
MVR groups:
        Group start     Group end       Type  Count/Mask
        --------------- --------------- ----- ---------------
        225.0.7.226     225.0.7.226     count 1
        225.0.7.227     225.0.7.227     count 1
        225.0.7.228     225.0.7.228     count 1
```

```
225.0.7.229     225.0.7.229     count 1
225.0.7.230     225.0.7.230     count 1
225.0.7.231     225.0.7.231     count 1
236.8.7.0       236.8.7.255     mask  255.255.255.0
237.8.7.0       237.8.7.255     mask  255.255.255.0
237.8.8.0       237.8.8.255     mask  255.255.255.0
```

This example displays all MVR interfaces and their MVR configurations:

```
Router# show mvr interface
Port     VLAN  Type       Status           Immediate Leave
----     ----  ----       ------           ---------------
Gi1/20      2  RECEIVER   ACTIVE/UP        DISABLED
Gi1/21      2  SOURCE     ACTIVE/UP        DISABLED
```

This example displays all MVR members on VLAN 2:

```
Router# show mvr members vlan 2
MVR Group IP     Status    Members
------------     ------    -------
224.000.001.001   ACTIVE    Gi1/20(u),Gi1/21(u)
224.000.001.002   ACTIVE    Fa3/2(d),Gi1/12(u)
```

This example displays the number of MVR members on all MVR VLANs:

```
Router# show mvr members count

Count of active MVR groups:
    Vlan 490: 400
    Vlan 600: 400
    Vlan 700: 0
    Vlan 950: 0
```

This example displays all receiver ports that are members of any IP multicast group:

```
Router# show mvr receiver-ports
Joins: v1,v2,v3 counter shows total IGMP joins
       v3 counter shows IGMP joins received with both MVR and non-MVR groups
Port   VLAN Status         Immediate          Joins
                          Leave        (v1,v2,v3)      (v3)
----   ---- -------------  ----------   ---------- -----------
Gi1/7   202 INACTIVE/UP    ENABLED         305336           0
Gi1/8   202 ACTIVE/UP     DISABLED           4005           0
Gi1/9   203 INACTIVE/DOWN  DISABLED         53007           0
Gi1/10  203 ACTIVE/UP     DISABLED           6204           0
Gi1/11  204 ACTIVE/UP     DISABLED              0         940
Gi1/12  205 INACTIVE/UP    ENABLED           8623           0
```

Clearing MVR Counters

You can clear MVR join counters for the switch, for source or receiver ports, or for a specified interface.

To clear MVR counters, perform this task:

Command	Purpose
Router# **clear mvr counters** [[**receiver-ports** | **source-ports**] [*type module/port*]]	Clears the join counters of all the MVR ports, or source or receiver ports, or of a specified MVR interface port.

This example clears the join counters for the receiver port on GigabitEthernet port 1/7:

```
Router# clear mvr receiver-ports GigabitEthernet 1/7
Router# show mvr receiver-ports GigabitEthernet 1/7
Joins: v1,v2,v3 counter shows total IGMP joins
       v3 counter shows IGMP joins received with both MVR and non-MVR groups
Port   VLAN Status        Immediate          Joins
                          Leave        (v1,v2,v3)     (v3)
----   ---- ------------- ----------   ---------- -----------
Gi1/7   202 INACTIVE/UP    ENABLED              0           0
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring IPv4 IGMP Filtering and Router Guard

This chapter describes how IGMP traffic filtering and Router Guard are used to control the access of a port to IGMP traffic. Release 12.2(33)SXH and later releases support IGMP traffic filtering and Router Guard.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

The following sections describe IGMP filtering and Router Guard features for multicast hosts (receivers):

Understanding IGMP Filtering

These sections describe IGMP filtering:

IGMP Filtering Overview

IGMP snooping is a protocol that learns and maintains multicast group membership at the Layer 2 level. IGMP snooping looks at IGMP traffic to decide which ports should be allowed to receive multicast traffic from certain sources and for certain groups. This information is used to forward multicast traffic to only interested ports. The main benefit of IGMP snooping is to reduce flooding of packets. For information about IGMP snooping, see "Understanding IGMP Filtering" section on page 40-1.

IGMP filtering allows users to configure filters on a switch virtual interface (SVI), a per-port, or a per-port per-VLAN basis to control the propagation of IGMP traffic through the network. By managing the IGMP traffic, IGMP filtering provides the capability to manage IGMP snooping, which in turn controls the forwarding of multicast traffic.

When an IGMP packet is received, IGMP filtering uses the filters configured by the user to determine whether the IGMP packet should be discarded or allowed to be processed by the existing IGMP snooping code. With a IGMP version 1 or version 2 packet, the entire packet is discarded. With a IGMPv3 packet, the packet is rewritten to remove message elements that were denied by the filters.

The IGMP filtering feature is SSO compliant.

IGMP traffic filters control the access of a port to multicast traffic. Access can be restricted based on the following:

- Which multicast groups or channels can be joined on a port. Channels are joined by IGMPv3 hosts that specify both the group and the source of the multicast traffic.
- Maximum number of groups or channels allowed on a specific port or interface (regardless of the number of hosts requesting service).
- IGMP protocol versions (for example, disallow all IGMPv1 messages).

When you enter an IGMP filtering command, a user policy is applied to a Layer 3 SVI interface, a Layer 2 port, or a particular VLAN on a Layer 2 trunk port. The Layer 2 port may be an access port or a trunk port. The IGMP filtering features will work only if IGMP snooping is enabled (either on the interface or globally).

IGMP filtering is typically used in access switches connected to end-user devices in Ethernet-to-home deployment scenarios.

Note IGMP, which runs at Layer 3 on a multicast router, generates Layer 3 IGMP queries in subnets where the multicast traffic needs to be routed. For information about IGMP, see Chapter 37, "Understanding IPv4 Multicast Layer 3 Switching."

IGMP Filters

There are three different types of IGMP filters: IGMP group and channel access control, several IGMP groups and channels limit, and an IGMP minimum version. These filters are configurable and operate differently on different types of ports:

- Per SVI
- Per port
- Per VLAN basis on a trunk port

In the case of trunk ports, filters may also be configured separately for each of the different VLANs passing through that trunk port.

The following sections describe each type of filter in more detail:

- IGMP Group and Channel Access Control, page 40-3
- Number of IGMP Groups and Channels Limit, page 40-3
- IGMP Minimum Version, page 40-4

IGMP Group and Channel Access Control

Filtering on the IGMP group or channel allows the user to control which IGMP groups or channels can be joined on a port or on a per VLAN basis on a trunk port.

To configure filtering on the IGMP group or channel use the following CLI command:

ip igmp snooping access-group *acl* [**vlan** *vlan_id*]

To allow or deny several groups or channels, you must configure multiple access control entries in the access control list. Depending on whether the ACL is configured as permit or deny, the corresponding group or channel is allowed or denied. The ACL specified may be either a simple or extended ACL.

Filtering by IGMP group or channel is configurable on a Layer 3 SVI as a default filter for all ports in access mode under that SVI and for the corresponding VLAN on all trunk ports carrying that VLAN. This filter is also configurable on a Layer 2 port. If the port is in access mode, this filter will override any default SVI filter. If the port is in trunk mode, this filter will act as a default for all VLANs on that trunk and will override the SVI filter for each corresponding VLAN.

The **vlan** keyword can apply the filter only to IGMP packets arriving on the specified Layer 2 VLAN if the port is a trunk port. This per-VLAN filter (configured using the **vlan** keyword) will override any interface level filter and any SVI filter for the same VLAN.

Number of IGMP Groups and Channels Limit

Limiting the number of IGMP groups or channels allows you to control how many IGMP groups or channels can be joined on a port or on a per-VLAN basis on a trunk port.

To limit the number of IGMP groups or channels, use the following interface command CLI:

ip igmp snooping limit *n* [**except** *acl*] [**vlan** *vlan_id*]

A maximum of *n* groups or channels are allowed on the port or interface. The **except** keyword allows you to specify groups or channels that are exempt from the configured limit. The ACL used with the **except** keyword may be either a simple or extended ACL.

If joins are received for (*,G1) and (S1,G1) on the same interface, these are counted as two separate joins. If the limit on an interface has been set to 2, and joins are received for (*,G1) and (S1,G1), all other joins (for groups or channels different from these two) will then be discarded.

This filter is configurable on a Layer 3 SVI as a default filter for all ports in access mode under that SVI and for the corresponding VLAN on all trunk ports carrying that VLAN. This filter is also configurable on a Layer 2 port. If the Layer 2 port is in access mode, this filter will override any default SVI filter. If the Layer 2 switch port is in trunk mode, this filter will act as a default for all VLANs on that trunk and will override the SVI filter for each corresponding VLAN. The **vlan** keyword allows the user to apply the filter only to IGMP packets arriving on the specified Layer 2 VLAN if the Layer 2 switch port is a trunk port. This per-VLAN filter, configured using the **vlan** keyword, will override any interface level filter and any SVI filter for the same VLAN.

IGMP Minimum Version

Filtering on the IGMP protocol allows you to configure the minimum version of IGMP hosts allowed on the SVI. For example, you may want to disallow all IGMPv1 hosts (such as, allow a minimum IGMP version of 2) or all IGMPv1 and IGMPv2 hosts (such as, allow a minimum IGMP version of 3). This filtering applies only to membership reports.

To configure filtering on the IGMP protocol, use the following CLI command:

ip igmp snooping minimum-version 2 | 3

This filter is only configurable on a Layer 3 SVI as a default filter for all ports in access mode under that SVI and for the corresponding VLAN on all trunk ports.

IGMP Filter Precedence

These sections describe the hierarchy of the different filters on various ports.

Access Mode

In access mode, filters can be configured on both the port and the SVI. When an IGMP packet is received on a port in access mode, the port filter is checked first. If the port filter exists, it is applied and the SVI filter is ignored. If no per-port filter exists, the SVI filter is used.

This hierarchy is applied separately for each type of filter. For example, a limit filter configured on the port overrides the default limit filter on the SVI, but has no affect on any of the other filters.

Trunk Mode

With ports in trunk mode, a filter can be configured for an SVI corresponding to one of the VLANs on the trunk port, another filter configured on the trunk port itself, and a third filter configured on one of the Layer 2 VLANs passing through the trunk. When an IGMP packet is received, the trunk-per-VLAN specific filter will be checked first. If this filter exists, it is applied. The main trunk port filter and SVI filter will be ignored. If no trunk-per-VLAN filter exists, the main trunk port filter will be used. If neither of these filters exist, the SVI filter for the VLAN will be used as a final default for ports in trunk mode.

Filter Hierarchy Example

This example shows the filter hierarchy. The following configuration of SVI VLAN 100 contains three access ports g1/1, g1/2, and g1/3:

VLAN 100:

```
Switch(config-if)# ip igmp snooping limit 20
```

Port g1/1:

```
Switch(config-if)# ip igmp snooping limit 35
```

Port g1/2:

```
Switch(config-if)# no limit filter
```

Port g1/3:

```
Switch(config-if)# no limit filter
```

In this example, the limit value for g1/1 is 35, the limit value for g1/2 is 20, and the limit value for g1/3 is also 20.

Displaying IGMP Filtering

The following sections describe how to display IGMP filtering:

- Displaying IGMP Filtering Configuration, page 40-5
- Displaying IGMP Filtering Statistics, page 40-6

Displaying IGMP Filtering Configuration

To display IGMP filtering rules, perform this task:

Command	Purpose
Switch(config-if)# **show ip igmp snooping filter interface** *interface-name* [**details**]	Displays the filters configured for the specified interface.

This example shows how to display the default filters configured on the SVI:

```
Router# show ip igmp snooping filter interface vlan 20
Access-Group: Channel1-Acl
Groups/Channels Limit:100 (Exception List: Channel6-Acl)
IGMP Minimum-Version:Not Configured
```

This example shows how to display the filters configured for all ports in access mode under this SVI and for all trunk ports carrying the corresponding VLAN:

```
Router# show ip igmp snooping filter interface g3/48
  Access-Group: Channel4-Acl
  Groups/Channels Limit:10 (Exception List: Channel3-Acl)
```

This example shows how to display the filters configured for all ports in access mode under this SVI:

```
Router# show ip igmp snooping filter interface vlan 20 detail
GigabitEthernet3/47 :
  Access-Group: Not Configured
  Groups/Channels Limit: Not Configured
GigabitEthernet3/48      :
  Access-Group: Channel4-ACL
  Groups/Channels Limit: 10     (Exception-list: Channel3-Acl)
```

This example shows how to display the default trunk port filters:

```
Router# show ip igmp snooping filter interface g3/46
  Access-Group: Channel1-Acl
  Groups/Channels Limit: 10 (Exception List: Channel3-Acl)
```

This example shows how to display the per-VLAN filters for all VLANs on this trunk:

```
Router# show ip igmp snooping filter interface g3/46 detail
Vlan 10   :
```

```
  Access-Group: Not Configured
  Groups/Channels Limit: Not Configured
Vlan 20  :
  Access-Group: Not Configured
        Groups/Channels Limit: 8 (Exception List: Channel4-Acl)
```

This example shows how to display the per-VLAN filters for a specific VLAN on this trunk:

```
Router# show ip igmp snooping filter interface g3/46 vlan 20
  Access-Group: Not Configured
  Groups/Channels Limit: 8 (Exception List: Channel4-Acl)
```

Note If the port is in the shutdown state, filter status will not be displayed because it cannot be determined whether the port is in trunk mode or access mode. In this situation, you can use the **show running-config interface** *xxxx* command to view the configuration.

Displaying IGMP Filtering Statistics

Statistics are maintained on an interface basis for ports in access mode and on a per-VLAN basis for ports in trunk mode.

To display IGMP filtering statistics, perform this task:

Command	Purpose
Switch(config-if)# **show ip igmp snooping filter interface** *interface-name* [**statistics**]	Displays the filtering statistic collected for the specified interface.

This example shows how to display statistics for each port in access mode under the SVI:

```
Router# show ip igmp snooping filter interface vlan 20 statistics
GigabitEthernet3/47       :
  IGMP Filters are not configured

GigabitEthernet3/48       :
  Access-group denied : 0
  Limit denied : 2
  Limit status : 0 active out of 2 max
  Minimum-version denied : 0
```

This example shows how to display statistics for a specific port in access mode:

```
Router# show ip igmp snooping filter interface g3/48 statistics
  Access-group denied : 0
  Limit denied : 2
  Limit status : 0 active out of 2 max
  Minimum-version denied : 0
```

This example shows how to display statistics for Gigabit Ethernet port 3/47 in access mode with no default SVI filter and no port filter:

```
Router# show ip igmp snooping filter interface g3/47 statistics
IGMP Filters are not configured
```

This example shows how to display statistics for all VLANs under a trunk:

```
Router# show ip igmp snooping filter interface g3/46 statistics
```

```
Vlan 10   :
IGMP Filters are not configured

Vlan 20   :
  Access-group denied : 0
  Limit denied : 0
  Minimum-version denied : 0
```

This example shows how to display statistics for a specific VLAN under a trunk:

```
Router# show ip igmp snooping filter interface g3/46 vlan 20 statistics
  Access-group denied : 0
  Limit denied : 0
  Minimum-version denied : 0
```

This example shows how to display statistics for a specific VLAN under a trunk port with no trunk and no VLAN filter:

```
Router# show ip igmp snooping filter interface g3/46 vlan 10 statistics
IGMP Filters are not configured
```

Note If the port is in the shutdown state, filter statistics will not be displayed because it cannot be determined whether the port is in trunk mode or access mode.

Clearing IGMP Filtering Statistics

To clear IGMP filtering statistics, perform one of these tasks:

Command	Purpose
Router# **clear ip igmp snooping filter statistics**	Clears IGMP filtering statistics for all access ports and for all VLANs on all trunk ports.
Router# **clear ip igmp snooping filter statistics interface** *interface_name*	Clears statistics for one particular access port or for all VLANs on one particular trunk port.
Router# **clear ip igmp snooping filter statistics interface** *interface_name* **vlan** *vlan_ID*	Clears statistics for one particular VLAN on a trunk port.

Understanding Router Guard

These sections describe Router Guard:

- Router Guard Overview, page 40-8
- Configuring Router Guard, page 40-8
- Displaying Router Guard Configurations, page 40-9
- Displaying Router Guard Interfaces, page 40-10

Router Guard Overview

The Router Guard feature allows you to designate a specified port only as a multicast host port and not as a multicast router port. Multicast router control packets received on this port are dropped.

Any port can become a multicast router port if the switch receives one of the multicast router control packets, such as IGMP general query, PIM hello, or CGMP hello. When a port becomes a multicast router port, all multicast traffic (both known and unknown source traffic) is sent to all multicast router ports. This cannot be prevented without Router Guard.

When configured, the Router Guard feature makes the specified port a host port only. The port is prevented from becoming a router port, even if a multicast router control packets are received.

In addition, any control packets normally received from multicast routers, such as IGMP queries and PIM joins, will also be discarded by this filter.

A Router Guard command applies a user policy to a Layer 3 SVI interface, a Layer 2 port, or a particular VLAN on a Layer 2 trunk port. The Layer 2 port may be an access port or a trunk port.

The Router Guard feature does not require IGMP snooping to be enabled.

Router Guard is implemented only for IPv4.

Router Guard is typically used in access switches connected to end-user boxes in Ethernet-to-home deployment scenarios.

The IPv4 multicast Router Guard feature is SSO-compliant.

The following packet types are discarded if they are received on a port that has Router Guard enabled:

- IGMP query messages
- IPv4 PIMv2 messages
- IGMP PIM messages (PIMv1)
- IGMP DVMRP messages
- RGMP messages
- CGMP messages

When these packets are discarded, statistics are updated indicating that packets are being dropped due to Router Guard.

Configuring Router Guard

The Router Guard feature can be configured globally and per-interface. Typically, the global configuration initiates Router Guard for all Layer 2 ports in the system. The per-interface configuration can then be used to override Router Guard for specific ports, for example, the ports where multicast routers are actually connected.

The following sections describe each type of configuration:

- Enabling Router Guard Globally, page 40-9
- Clearing Router Guard Statistics, page 40-9
- Disabling Router Guard on Ports, page 40-9

Enabling Router Guard Globally

To enable Router Guard globally, perform this task:

Command	Purpose
Router# **router-guard ip multicast switchports**	Enables Router Guard globally.

Clearing Router Guard Statistics

To clear Router Guard statistics, perform this task:

	Command	Purpose
Step 1	Router# **clear ip igmp snooping filter statistics interface** *interface_name*	Clears statistics for one particular access port or for all VLANs on one particular trunk port.
Step 2	Router# **clear ip igmp snooping filter statistics interface** *interface_name* **vlan** *vlan_id*	Clears statistics for one particular VLAN on a trunk port.

Disabling Router Guard on Ports

To disable Router Guard on a Layer 2 port to which a multicast router is connected, perform this task:

Command	Purpose
Router(config-if)# **no router-guard ip multicast** [**vlan** *vlan_id*]	Disables Router Guard on a Layer 2 port. **Note** The **vlan** keyword is effective only if the port is in trunk mode. You can use this keyword to override Router Guard only for specific VLANs on the trunk port.

This example shows how to allow multicast router messages on trunk port Gigabit Ethernet 3/46, VLAN 20:

```
Router# configure terminal
Router(config)# interface gigabitethernet 3/46
Router(config-if)# no router-guard ip multicast vlan 20
```

Displaying Router Guard Configurations

To display the global Router Guard configuration and the Router Guard configuration for a specific interface, perform these tasks:

Command	Purpose
Router# **show router-guard**	Displays the global Router Guard configuration.
Router# **show router-guard interface** *interface_name*	Displays the Router Guard configuration for a specific interface.

This example shows how to display the interface command output for a port in access mode with Router Guard not active:

```
Router# show router-guard interface g3/48
  Router Guard for IP Multicast:
Globally enabled for all switch ports
Enabled on this interface
Packets denied:
  IGMP Queries:
  PIMv2 Messages:
  PIMv1 Messages:
  DVMRP Messages:
  RGMP Messages:
  CGMP Messages:
```

This example shows how to display the interface command output for a port in trunk mode:

```
Router# show router-guard interface g3/48
  Router Guard for IP Multicast:
Globally enabled for all switch ports
Disabled on this interface
```

This example shows how to verify that a trunk port is carrying VLANs 10 and 20:

```
Router# show router-guard interface g3/46
  Router Guard for IP Multicast:
Globally enabled for all switch ports
Default: Enabled for all VLANs on this interface
VLAN 10:
Enabled on this VLAN
Packets denied:
  IGMP Queries:
  PIMv2 Messages:
  PIMv1 Messages:
  DVMRP Messages:
  RGMP Messages:
  CGMP Messages:
VLAN 20 :
Disabled on this VLAN
```

Note If the port is in the shutdown state, the status will not be displayed because it cannot be determined whether the port is in trunk mode or access mode. You can use the **show running-config interface** *xxxx* command to display the Router Guard configuration.

Displaying Router Guard Interfaces

To display a list of all interfaces for which Router Guard is disabled, perform this task:

Command	Purpose
Router# **show router-guard interface** Router Guard for IP Multicast: Globally enabled for all switchports Interfaces: Gi3/46: Disabled on this port for VLANS: ALL	Displays a list of all interfaces for which Router Guard is disabled.

Clearing Router Guard Statistics

To clear Router Guard statistics, perform one of these tasks:

Command	Purpose
Router(config)# **clear router-guard ip multicast statistics**	Clears statistics for all access ports and for all VLANs on all trunk ports.
Router(config)# **clear router-guard ip multicast statistics interface** *interface_name*	Clears statistics for an access port and for all VLANs on a trunk port.
Router(config)# **clear router-guard ip multicast statistics interface** *interface_name* **vlan** *v*	Clears statistics for one particular VLAN on a trunk port.

This example shows how to clear statistics for one particular VLAN on a trunk port:

```
Router# clear router-guard ip multicast statistics interface interface_name vlan v
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring PIM Snooping

This chapter describes how to configure protocol independent multicast (PIM) snooping in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

---

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding PIM Snooping

In networks where a Layer 2 switch interconnects several routers, such as an Internet exchange point (IXP), the switch floods IP multicast packets on all multicast router ports by default, even if there are no multicast receivers downstream. With PIM snooping enabled, the switch restricts multicast packets for each IP multicast group to only those multicast router ports that have downstream receivers joined to that group. When you enable PIM snooping, the switch learns which multicast router ports need to receive the multicast traffic within a specific VLAN by listening to the PIM hello messages, PIM join and prune messages, and bidirectional PIM designated forwarder-election messages. With Release 12.2(33)SXJ2 and later releases, PIM snooping also constrains multicast traffic to VPLS interfaces.

Note To use PIM snooping, you must enable IGMP snooping on the switch. IGMP snooping restricts multicast traffic that exits through the LAN ports to which hosts are connected. IGMP snooping does not restrict traffic that exits through the LAN ports to which one or more multicast routers are connected.

The following illustrations show the flow of traffic and flooding that results in networks without PIM snooping enabled and the flow of traffic and traffic restriction when PIM snooping is enabled.

Figure 41-1 shows the flow of a PIM join message without PIM snooping enabled. In the figure, the switches flood the PIM join message intended for Router B to all connected routers.

Figure 41-1 PIM Join Message Flow without PIM Snooping

\

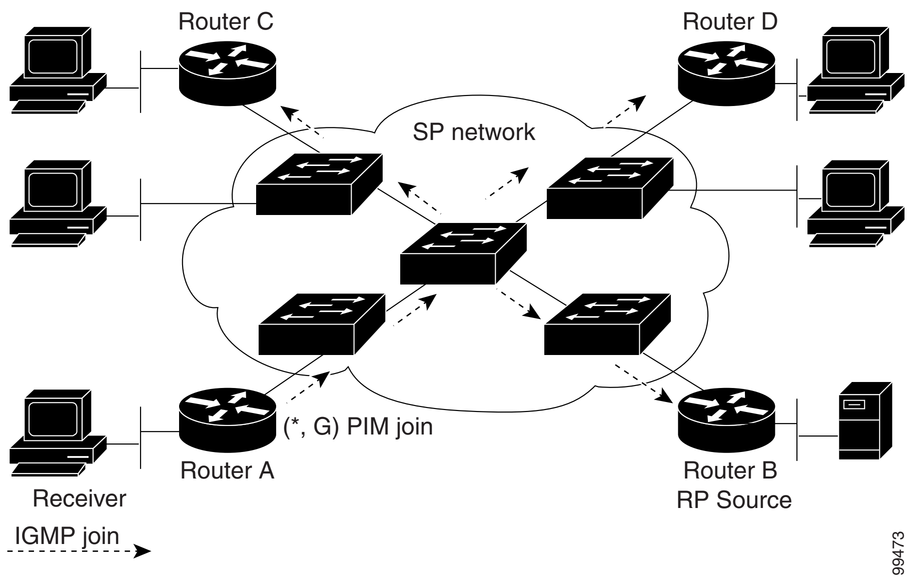


Figure 41-2 shows the flow of a PIM join message with PIM snooping enabled. In the figure, the switches restrict the PIM join message and forward it only to the router that needs to receive it (Router B).

Figure 41-2 PIM Join Message Flow with PIM Snooping

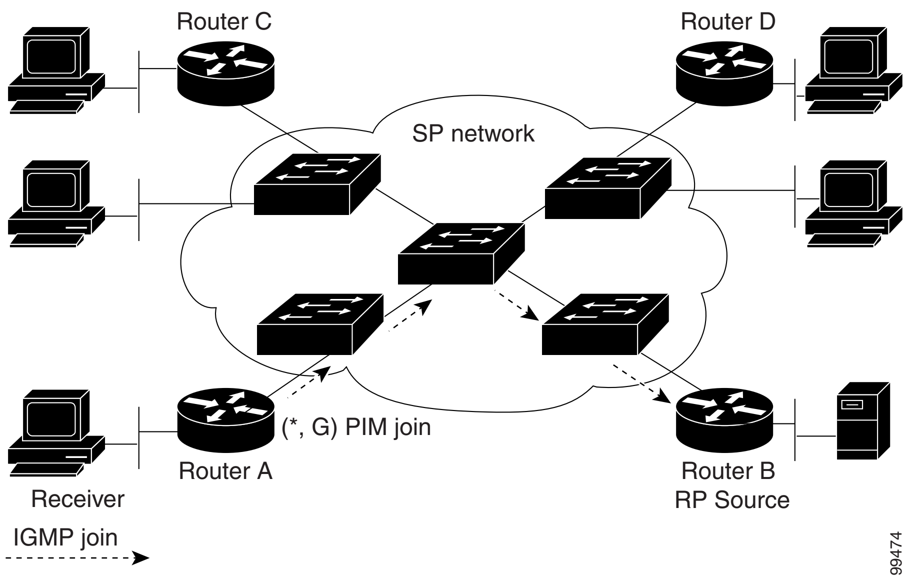


Figure 41-3 shows the flow of data traffic without PIM snooping enabled. In the figure, the switches flood the data traffic intended for Router A to all connected routers.

Figure 41-3 Data Traffic Flow without PIM Snooping

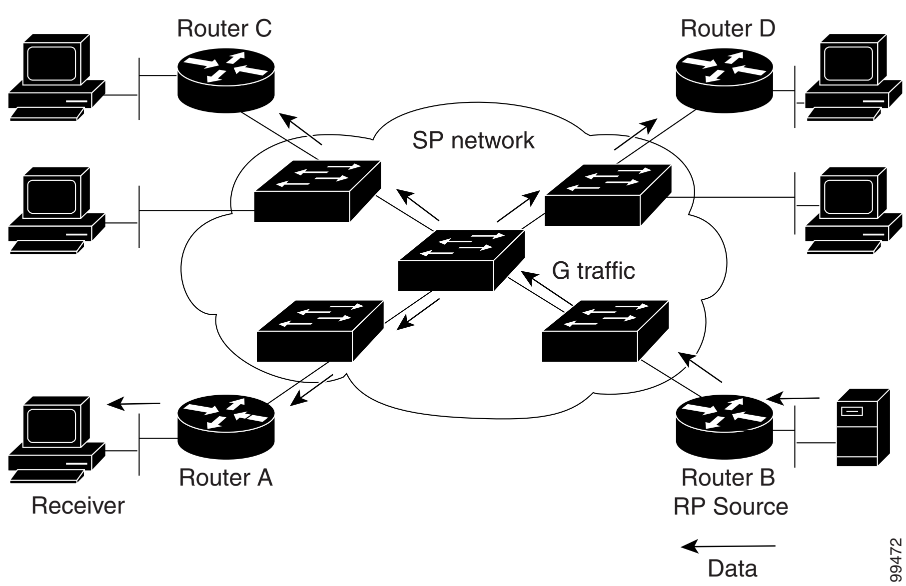
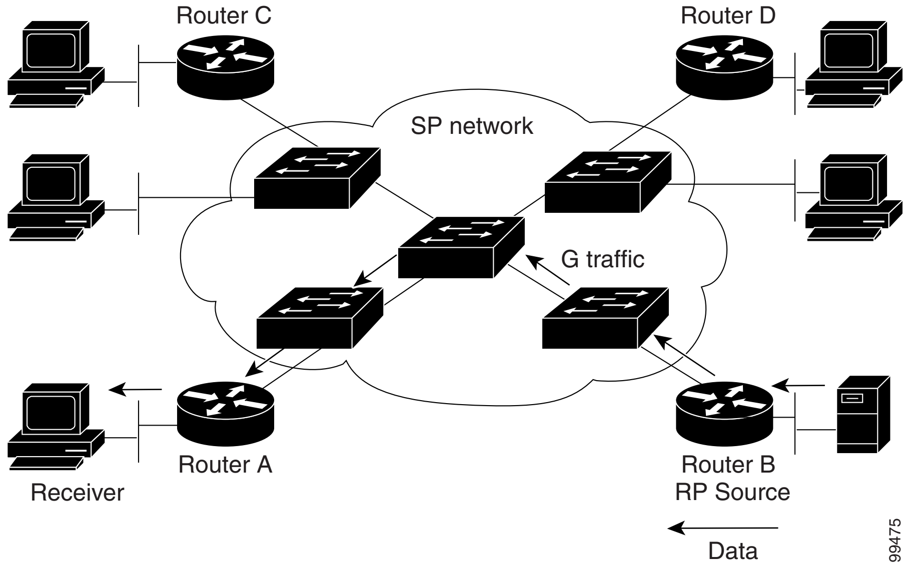


Figure 41-4 shows the flow of data traffic with PIM snooping enabled. In the figure, the switches forward the data traffic only to the router that needs to receive it (Router A).

Figure 41-4 Data Traffic Flow with PIM Snooping



Default PIM Snooping Configuration

PIM snooping is disabled by default.

PIM Snooping Configuration Guidelines and Restrictions

When configuring PIM snooping, follow these guidelines and restrictions:

- When you use the PIM-sparse mode (PIM-SM) feature, downstream routers only see traffic if they previously indicated interest through a PIM join or prune message. An upstream router only sees traffic if it was used as an upstream router during the PIM join or prune process.
- Join or prune messages are not flooded on all router ports but are sent only to the port corresponding to the upstream router mentioned in the payload of the join or prune message.
- Directly connected sources are supported for bidirectional PIM groups. Traffic from directly connected sources is forwarded to the designated router and designated forwarder for a VLAN. In some cases, a nondesignated router (NDR) can receive a downstream (S, G) join. For source-only networks, the initial unknown traffic is flooded only to the designated routers and designated forwarders.
- Dense group mode traffic is seen as unknown traffic and is dropped.
- The AUTO-RP groups (224.0.1.39 and 224.0.1.40) are always flooded.
- The switch snoops on designated forwarder election and maintains a list of all designated forwarder routers for various RPs for the VLAN. All traffic is sent to all designated forwarders which ensures that bidirectional functionality works properly.
- PIM snooping and IGMP snooping can be enabled at the same time in a VLAN. Either RGMP or PIM snooping can be enabled in a VLAN but not both.

- Any non-PIMv2 multicast router will receive all traffic.
- You can enable or disable PIM snooping on a per-VLAN basis.
- All mroute and router information is timed out based on the hold-time indicated in the PIM hello and join/prune control packets. All mroute state and neighbor information is maintained per VLAN.

Configuring PIM Snooping

These sections describe how to configure PIM snooping:

- Enabling PIM Snooping Globally, page 41-5
- Enabling PIM Snooping in a VLAN, page 41-5
- Disabling PIM Snooping Designated-Router Flooding, page 41-6

Enabling PIM Snooping Globally

To enable PIM snooping globally, perform this task:

	Command	Purpose
Step 1	Router(config)# **ip pim snooping**	Enables PIM snooping.
	Router(config)# **no ip pim snooping**	Disables PIM snooping.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show ip pim snooping**	Verifies the configuration.

This example shows how to enable PIM snooping globally and verify the configuration:

```
Router(config)# ip pim snooping
Router(config)# end
Router# show ip pim snooping
Global runtime mode: Enabled
Global admin mode  : Enabled
Number of user enabled VLANs: 1
User enabled VLANs: 10
Router#
```

Note You do not need to configure an IP address or IP PIM in order to run PIM snooping.

Enabling PIM Snooping in a VLAN

Note When you apply the **ip pim snooping** command and associated commands on any VLAN interface, the commands continue to function even if the VLAN interface is in shutdown state.

To enable PIM snooping in a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ip pim snooping`	Enables PIM snooping.
	`Router(config-if)# no ip pim snooping`	Disables PIM snooping.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show ip pim snooping`	Verifies the configuration.

This example shows how to enable PIM snooping on VLAN 10 and verify the configuration:

```
Router# interface vlan 10
Router(config-if)# ip pim snooping
Router(config-if)# end
Router# show ip pim snooping vlan 10
3 neighbors (0 DR priority incapable, 0 Bi-dir incapable)
6 mroutes, 3 mac entries
DR is 10.10.10.4
RP DF Set
Router#
```

Disabling PIM Snooping Designated-Router Flooding

Note Do not disable designated-router flooding on switches in a Layer 2 broadcast domain that supports multicast sources.

By default, switches that have PIM snooping enabled will flood multicast traffic to the designated router (DR). This method of operation can send unnecessary multicast packets to the designated router. The network must carry the unnecessary traffic, and the designated router must process and drop the unnecessary traffic.

To reduce the traffic sent over the network to the designated router, disable designated-router flooding. With designated-router flooding disabled, PIM snooping only passes to the designated-router traffic that is in multicast groups for which PIM snooping receives an explicit join from the link towards the designated router.

To disable PIM snooping designated-router flooding, perform this task:

	Command	Purpose
Step 1	`Router(config)# no ip pim snooping dr-flood`	Disables PIM snooping designated-router flooding.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show running-config | include dr-flood`	Verifies the configuration.

This example shows how to disable PIM snooping designated-router flooding:

```
Router(config)# no ip pim snooping dr-flood
Router(config)# end
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring IPv4 Multicast VPN Support

This chapter describes how to configure IPv4 Multicast Virtual Private Network (MVPN) support in Cisco IOS Release 12.2SX.

Note

- PFC3A modes does not support MVPN.
- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains these sections:

Understanding MVPN

These sections describe MVPN:

- Hardware-Assisted IPv4 Multicast, page 42-6

MVPN Overview

MVPN is a standards-based feature that transmits IPv4 multicast traffic across an MPLS VPN cloud. MVPN uses the IPv4 multicast traffic PFC hardware support to forward multicast traffic over VPNs at wire speeds. MVPN adds support for IPv4 multicast traffic over Layer 3 IPv4 VPNs to the existing IPv4 unicast support.

MVPN routes and forwards multicast packets for each individual VPN routing and forwarding (VRF) instance, as well as transmitting the multicast packets through VPN tunnels across the service provider backbone.

MVPN is an alternative to IP-in-IP generic route encapsulation (GRE) tunnels. GRE tunnels are not a readily scalable solution and they are limited in the granularity they provide to customers.

Multicast Routing and Forwarding and Multicast Domains

MVPN adds multicast routing information to the VPN routing and forwarding table. When a provider-edge (PE) router receives multicast data or control packets from a customer-edge (CE) router, forwarding is performed according to the information in the multicast VRF (MVRF).

Note MVRF is also commonly referred to as multicast over VRF-lite.

Each MVRF maintains the routing and forwarding information that is needed for its particular VRF instance. An MVRF is created and configured in the same way as existing VRFs, except multicast routing is also enabled on each MVRF.

A multicast domain constitutes the set of hosts that can send multicast traffic to each other within the MPLS network. For example, the multicast domain for a customer that wanted to send certain types of multicast traffic to all global employees would consist of all CE routers associated with that enterprise.

Multicast Distribution Trees

The MVPN feature establishes at least one multicast distribution tree (MDT) for each multicast domain. The MDT provides the information needed to interconnect the same MVRFs that exist on the different PE routers.

MVPN supports two MDT types:

- Default MDT—The default MDT is a permanent channel for PIM control messages and low-bandwidth streams between all PE routers in a particular multicast domain. All multicast traffic in the default MDT is replicated to every other PE router in the domain. Each PE router is logically seen as a PIM neighbor (one hop away) from every other PE router in the domain.
- Data MDT—Data MDTs are optional. If enabled, they are dynamically created to provide optimal paths for high-bandwidth transmissions, such as full-motion video, that do not need to be sent to every PE router. This allows for on-demand forwarding of high-bandwidth traffic between PE routers, so as to avoid flooding every PE router with every high-bandwidth stream that might be created.

To create data MDTs, each PE router that is forwarding multicast streams to the backbone periodically examines the traffic being sent in each default MDT as follows:

1. Each PE router periodically samples the multicast traffic (approximately every 10 seconds for software switching, and 90 seconds for hardware switching) to determine whether a multicast stream has exceeded the configured threshold. (Depending on when the stream is sampled, this means that in a worst-case scenario, it could take up to 180 seconds before a high-bandwidth stream is detected.)

   Note Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries.

2. If a particular multicast stream exceeds the defined threshold, the sending PE router dynamically creates a data MDT for that particular multicast traffic.
3. The sending PE router then transmits a DATA-MDT JOIN request (which is a User Datagram Protocol (UDP) message to port 3232) to the other PE routers, informing them of the new data MDT.
4. Receiving PE routers examine their VRF routing tables to determine if they have any customers interested in receiving this data stream. If so, they use the PIM protocol to transmit a PIM JOIN message for this particular data MDT group (in the global table PIM instance) to accept the stream. Routers that do not currently have any customers for this stream still cache the information, in case any customers request it later on.
5. Three seconds after sending the DATA-MDT JOIN message, the sending PE router removes the high-bandwidth multicast stream from the default MDT and begins transmitting it over the new data MDT.
6. The sending PE router continues to send a DATA-MDT JOIN message every 60 seconds, as long as the multicast stream continues to exceed the defined threshold. If the stream falls below the threshold for more than 60 seconds, the sending PE router stops sending the DATA-MDT JOIN messages, and moves the stream back to the default MDT.
7. Receiving routers age out the cache information for the default MDT when they do not receive a DATA-MDT JOIN message for more than three minutes.

Data MDTs allow for high-bandwidth sources inside the VPN while still ensuring optimal traffic forwarding in the MPLS VPN core.

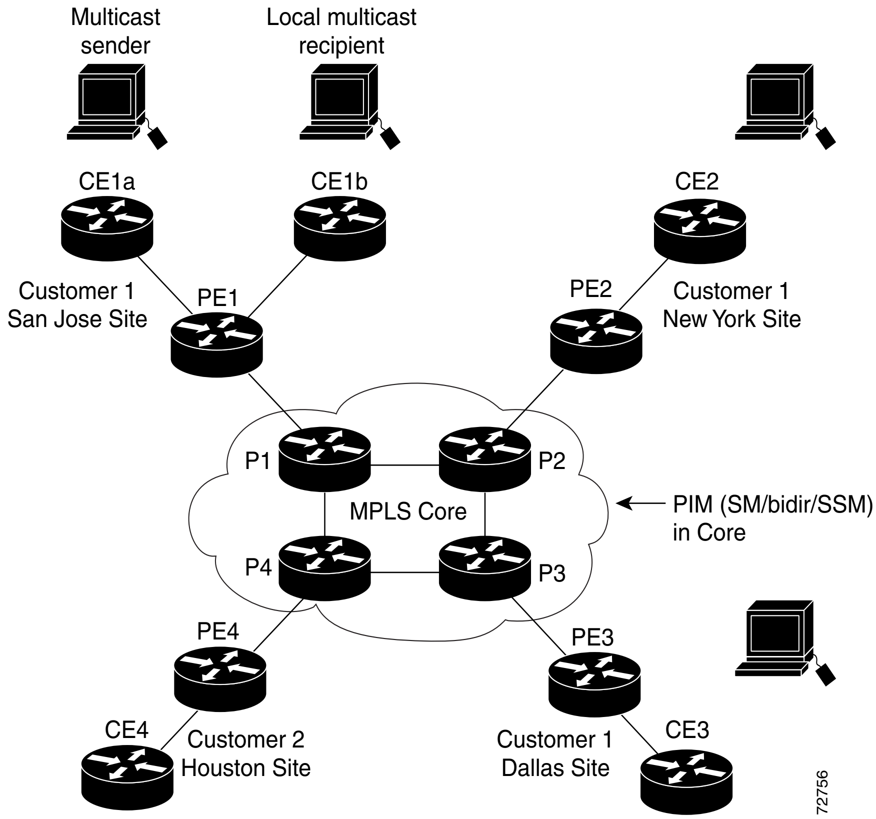
Note For technical information about the DATA-MDT JOIN message and other aspects of the data MDT creation and usage, see the Internet-Draft, *Multicast in MPLS/BGP IP VPNs*, by Eric C. Rosen et al.

In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. The San Jose site is transmitting a one-way multicast presentation. The service provider network supports all three sites associated with this customer, in addition to the Houston site of a different enterprise customer.

The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. Although PE4 is interconnected to these other routers in the MPLS core, PE4 is associated with a different customer and is therefore not part of the default MDT.

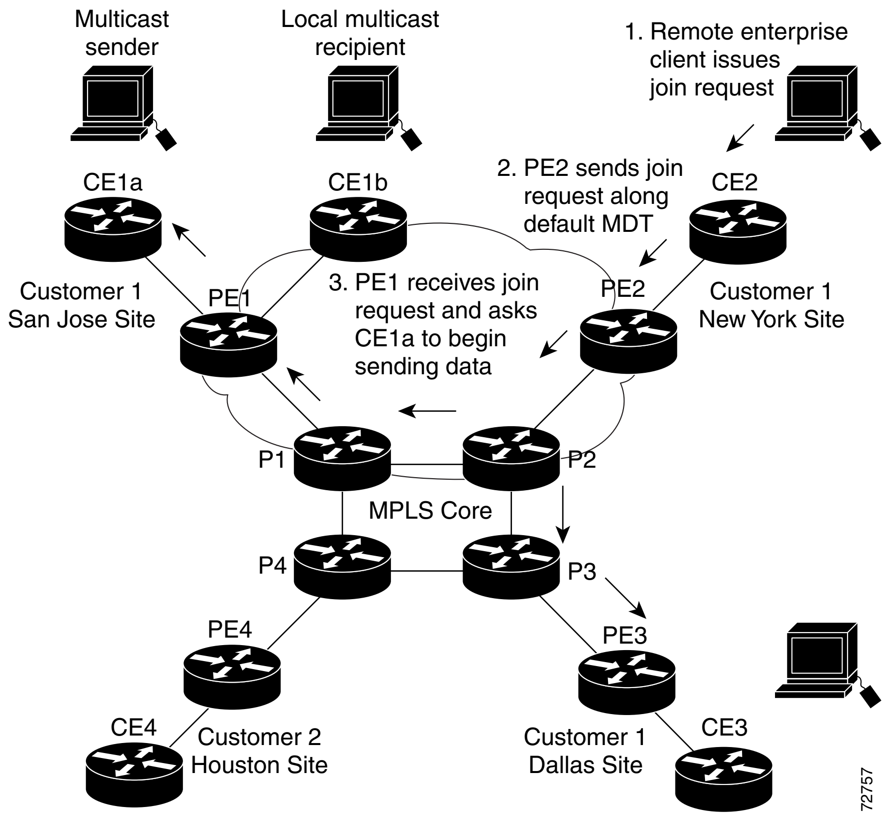
Figure 42-1 shows the situation in this network when no one outside of San Jose has joined the multicast broadcast, which means that no data is flowing along the default MDT. Each PE router maintains a PIM relationship with the other PE routers over the default MDT, as well as a PIM relationship with its directly attached PE routers.

Figure 42-1 Default Multicast Distribution Tree Overview



If an employee in New York joins the multicast session, the PE router associated for the New York site sends a join request that flows across the default MDT for the multicast domain. The PE router associated with the multicast session source (PE1) receives the request. Figure 42-2 shows how the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 42-2 Initializing the Data MDT



The CE router (CE1a) starts sending the multicast data to the associated PE router (PE1), which recognizes that the multicast data exceeds the bandwidth threshold at which a data MDT should be created. PE1 then creates a data MDT and sends a message to all routers using the default MDT that contains information about the data MDT.

Approximately three seconds later, PE1 begins sending the multicast data for that particular stream using the data MDT. Because only PE2 has receivers who are interested in this source, only PE2 joins the data MDT and receives traffic on it.

Multicast Tunnel Interfaces

The PE router creates a multicast tunnel interface (MTI) for each multicast VRF (MVRF) in the multicast domain. The MVRF uses the tunnel interface to access the multicast domain to provide a conduit that connects an MVRF and the global MVRF.

On the router, the MTI is a tunnel interface (created with the **interface tunnel** command) with a class D multicast address. All PE routers that are configured with a default MDT for this MVRF create a logical network in which each PE router appears as a PIM neighbor (one hop away) to every other PE router in the multicast domain, regardless of the actual physical distance between them.

The MTI is automatically created when an MVRF is configured. The BGP peering address is assigned as the MTI interface source address, and the PIM protocol is automatically enabled on each MTI.

When the router receives a multicast packet from the customer side of the network, it uses the incoming interface's VRF to determine which MVRFs should receive it. The router then encapsulates the packet using GRE encapsulation. When the router encapsulates the packet, it sets the source address to that of the BGP peering interface and sets the destination address to the multicast address of the default MDT, or to the source address of the data MDT if configured. The router then replicates the packet as needed for forwarding on the appropriate number of MTI interfaces.

When the router receives a packet on the MTI interface, it uses the destination address to identify the appropriate default MDT or data MDT, which in turn identifies the appropriate MVRF. It then decapsulates the packet and forwards it out the appropriate interfaces, replicating it as many times as are necessary.

Note

- Unlike other tunnel interfaces that are commonly used on Cisco routers, the MVPN MTI is classified as a LAN interface, not a point-to-point interface. The MTI interface is not configurable, but you can use the **show interface tunnel** command to display its status.
- The MTI interface is used exclusively for multicast traffic over the VPN tunnel.
- The tunnel does not carry unicast routed traffic.

PE Router Routing Table Support for MVPN

Each PE router that supports the MVPN feature uses the following routing tables to ensure that the VPN and MVPN traffic is routed correctly:

- Default routing table—Standard routing table used in all Cisco routers. This table contains the routes that are needed for backbone traffic and for non-MPLS VPN unicast and multicast traffic (including Generic Routing Encapsulation (GRE) multicast traffic).
- VPN routing/forwarding (VRF) table—Routing table created for each VRF instance. Responsible for routing the unicast traffic between VPNs in the MPLS network.
- Multicast VRF (MVRF) table—Multicast routing table and multicast routing protocol instance created for each VRF instance. Responsible for routing the multicast traffic in the multicast domain of the network. This table also includes the multicast tunnel interfaces that are used to access the multicast domain.

Multicast Distributed Switching Support

MVPN supports multicast distributed switching (MDS) for multicast support on a per-interface and a per-VRF basis. When configuring MDS, you must make sure that no interface (including loopback interfaces) has the **no ip mroute-cache** command configured.

Hardware-Assisted IPv4 Multicast

Cisco IOS Release 12.2SX supports hardware acceleration for IPv4 multicast over VPN traffic, which forwards multicast traffic to the appropriate VPNs at wire speed without increased RP CPU utilization.

In a customer VRF, PFC hardware acceleration supports multicast traffic in PIM dense, PIM sparse, PIM bidirectional, and PIM Source Specific Multicast (SSM) modes.

In the service provider core, PFC hardware acceleration supports multicast traffic in PIM sparse, PIM bidirectional, and PIM SSM modes. In the service provider core, PFC hardware acceleration does not support multicast traffic in PIM dense mode.

MVPN Configuration Guidelines and Restrictions

When configuring MVPN, follow these guidelines and restrictions:

- PFC3A mode does not support MVPN.
- All PE routers in the multicast domain need to be running a Cisco IOS software image that supports the MVPN feature. There is no requirement for MVPN support on the P and CE routers.
- Support for IPv4 multicast traffic must also be enabled on all backbone routers.
- The Border Gateway Protocol (BGP) routing protocol must be configured and operational on all routers supporting multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.
- Only ingress replication is supported when MVPN is configured. If the switch is currently configured for egress replication, it is forced into ingress replication when the first MVRF is configured.
- When the switch is acting as a PE, and receives a multicast packet from a customer router with a time-to-live (TTL) value of 2, it drops the packet instead of encapsulating it and forwarding it across the MVPN link. Because such packets would normally be dropped by the PE at the other end of the MVPN link, this does not affect traffic flow.
- If the core multicast routing uses SSM, then the data and default multicast distribution tree (MDT) groups must be configured within the SSM range of IPv4 addresses.
- The update source interface for the BGP peerings must be the same for all BGP peerings configured on the router in order for the default MDT to be configured properly. If you use a loopback address for BGP peering, then PIM sparse mode must be enabled on the loopback address.
- The **ip mroute-cache** command must be enabled on the loopback interface used as the BGP peering interface in order for distributed multicast switching to function on the platforms that support it. The **no ip mroute-cache** command must *not* be present on these interfaces.
- Data MDTs are not created for VRF PIM dense mode multicast streams because of the flood and prune nature of dense mode multicast flows and the resulting periodic bring-up and tear-down of such data MDTs.
- Data MDTs are not created for VRF PIM bidirectional mode because source information is not available.
- MVPN does not support multiple BGP peering update sources, and configuring them can break MVPN RPF checking. The source IPv4 address of the MVPN tunnels is determined by the highest IPv4 address used for the BGP peering update source. If this IPv4 address is not the IPv4 address used as the BGP peering address with the remote PE router, MVPN will not function properly.
- MDT tunnels do not carry unicast traffic.
- Although MVPN uses the infrastructure of MPLS VPN networks, you cannot apply MPLS tags or labels to multicast traffic over the VPNs.

- Each MVRF that is configured with a default MDT uses three hidden VLANs (one each for encapsulation, decapsulation, and interface), in addition to external, user-visible VLANs. This means that an absolute maximum of 1,000 MVRFs are supported on each router. (MVRFs without a configured MDT still use one internal VLAN, so unused MVRFs should be deleted to conserve VLAN allocation.)
- Because MVPN uses MPLS, MVPN supports only the RPR redundancy modes. MPLS can coexist with NSF with SSO redundancy mode, but there is no support for stateful MPLS switchover.
- If your MPLS VPN network already contains a network of VRFs, you do not need to delete them or recreate them to be able to support MVRF traffic. Instead, configure the **mdt default** and **mdt data** commands, as listed in the following procedure, to enable multicast traffic over the VRF.
- BGP should be already configured and operational on all routers that are sending or receiving multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.
- The same MVRF must be configured on each PE router that is to support a particular VPN connection.
- Each PE router that supports a particular MVRF must be configured with the same **mdt default** command.
- The switch supports only ingress replication when MVPN is configured. If a switch is currently configured for egress replication, it is forced into ingress replication when the first MVRF is configured. If a switch is currently configured for egress replication, we recommend performing this task only during scheduled maintenance periods, so that traffic disruption can be kept to a minimum.

Configuring MVPN

These sections describe how to configure MVPN:

- Forcing Ingress Multicast Replication Mode (Optional), page 42-8
- Configuring a Multicast VPN Routing and Forwarding Instance, page 42-9
- Configuring Multicast VRF Routing, page 42-15
- Configuring Interfaces for Multicast Routing to Support MVPN, page 42-19

Note These configuration tasks assume that BGP is already configured and operational on all routers that are sending or receiving the multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.

Forcing Ingress Multicast Replication Mode (Optional)

The MVPN feature supports only ingress multicast replication mode. If the switch is currently configured for egress replication, it is forced into ingress replication when the first MVRF is configured. This change in replication mode automatically purges all forwarding entries in the hardware, temporarily forcing the switch into software switching until the table entries can be rebuilt.

To avoid disrupting customer traffic, we recommend verifying that the switch is already in ingress multicast replication mode before configuring any MVRFs.

This example shows how to verify the multicast replication mode:

```
Router# show mls ip multicast capability

Current mode of replication is Ingress
auto replication mode detection is ON

 Slot           Multicast replication capability
    2                       Egress
    5                       Egress
    6                       Egress
    8                       Ingress
    9                       Ingress

Router#
```

If the current replication mode is egress or if any of the switching modules are capable of egress replication mode, configure ingress replication mode during a scheduled maintenance period to minimize the disruption of customer traffic.

To configure ingress multicast replication mode, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# mls ip multicast replication-mode ingress`	Configures ingress multicast replication mode and disables automatic detection of the replication mode (enabled by default).
Step 3	`Router(config)# do show mls ip multicast capability | include Current`	Verifies the configuration.

This example shows how to configure ingress multicast replication mode and verify the configuration:

```
Router(config)# mls ip multicast replication-mode ingress
Router(config)# do show mls ip multicast capability | include Current
Current mode of replication is Ingress
```

Configuring a Multicast VPN Routing and Forwarding Instance

These sections describe how to configure a multicast VPN routing and forwarding (MVRF) instance for each VPN connection on each PE router that is to handle the traffic for each particular VPN connection that is to transmit or receive multicast traffic:

Configuring a VRF Entry

To configure a VRF entry, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# ip vrf vrf_name`	Configures a VRF routing table entry and a Cisco Express Forwarding (CEF) table entry and enters VRF configuration mode.
Step 3	`Router(config-vrf)# do show ip vrf vrf_name`	Verifies the configuration.

This example show how to configure a VRF named blue and verify the configuration:

```
Router# configure terminal
Router(config)# ip vrf blue
Router(config-vrf)# do show ip vrf blue
Name                             Default RD          Interfaces
  blue                             <not set>
```

Configuring the Route Distinguisher

To configure the route distinguisher, perform this task:

	Command	Purpose
Step 1	`Router(config-vrf)# rd route_distinguisher`	Specifies the route distinguisher for a VPN IPv4 prefix.
Step 2	`Router(config-vrf)# do show ip vrf vrf_name`	Verifies the configuration.

When configuring the route distinguisher, enter the route distinguisher in one of the following formats:

- 16-bit AS number:your 32-bit number (101:3)
- 32-bit IPv4 address:your 16-bit number (192.168.122.15:1)

This example show how to configure 55:1111 as the route distinguisher and verify the configuration:

```
Router(config-vrf)# rd 55:1111
Router(config-vrf)# do show ip vrf blue
Name                             Default RD          Interfaces
blue                             55:1111
```

Configuring the Route-Target Extended Community

To configure the route-target extended community, perform this task:

	Command	Purpose
Step 1	`Router(config-vrf)# route-target [import	export	both] route_target_ext_community`	Configures a route-target extended community for the VRF.
Step 2	`Router(config-vrf)# do show ip vrf detail`	Verifies the configuration.

When configuring the route-target extended community, note the following information:

- **import—**Imports routing information from the target VPN extended community.
- **export—**Exports routing information to the target VPN extended community.
- **both**—Imports and exports.
- *route_target_ext_community*—Adds the 48-bit route-target extended community to the VRF. Enter the number in one of the following formats:
  - 16-bit AS number:your 32-bit number (101:3)
  - 32-bit IPv4 address:your 16-bit number (192.168.122.15:1)

This example shows how to configure 55:1111 as the import and export route-target extended community and verify the configuration:

```
Router(config-vrf)# route-target both 55:1111
Router(config-vrf)# do show ip vrf detail
VRF blue; default RD 55:1111; default VPNID <not set>
VRF Table ID = 1
  No interfaces
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:55:1111
  Import VPN route-target communities
    RT:55:1111
  No import route-map
  No export route-map
CSC is not configured.
```

Configuring the Default MDT

To configure the default MDT, perform this task:

Command	Purpose
Router(config-vrf)# **mdt default** *group_address*	Configures the default MDT.

When configuring the default MDT, note the following information:

- The *group_address* is the multicast IPv4 address of the default MDT group. This address serves as an identifier for the MVRF community, because all provider-edge (PE) routers configured with this same group address become members of the group, which allows them to receive the PIM control messages and multicast traffic that are sent by other members of the group.
- This same default MDT must be configured on each PE router to enable the PE routers to receive multicast traffic for this particular MVRF.

This example shows how to configure 239.1.1.1 as the default MDT:

```
Router(config-vrf)# mdt default 239.1.1.1
```

Configuring Data MDTs (Optional)

To configure optional data MDTs, perform this task:

Command	Purpose
`Router(config-vrf)# **mdt data** *group_address wildcard_bits* [**threshold** *threshold_value*] [**list** *access_list*]`	(Optional) Configures a data MDTs for the specified range of multicast addresses.

When configuring optional data MDTs, note the following information:

- *group_address1*—Multicast group address. The address can range from 224.0.0.1 to 239.255.255.255, but cannot overlap the address that has been assigned to the default MDT.
- *wildcard_bits*—Wildcard bit mask to be applied to the multicast group address to create a range of possible addresses. This allows you to limit the maximum number of data MDTs that each MVRF can support.
- **threshold** *threshold_value*—(Optional) Defines the threshold value in kilobits, at which multicast traffic should be switched from the default MDT to the data MDT. The *threshold_value* parameter can range from 1 through 4294967 kilobits.
- **list** *access_list*—(Optional) Specifies an access list name or number to be applied to this traffic.

This example shows how to configure a data MDT:

```
Router(config-vrf)# mdt data 239.1.2.0 0.0.0.3 threshold 10
```

Enabling Data MDT Logging

To enable data MDT logging, perform this task:

Command	Purpose
`Router(config-vrf)# **mdt log-reuse**`	(Optional) Enables the recording of data MDT reuse information, by generating a SYSLOG message whenever a data MDT is reused. Frequent reuse of a data MDT might indicate a need to increase the number of allowable data MDTs by increasing the size of the wildcard bitmask that is used in the **mdt data** command.

This example shows how to enable data MDT logging:

```
Router(config-vrf)# mdt log-reuse
```

Sample Configuration

The following excerpt from a configuration file shows typical VRF configurations for a range of VRFs. To simplify the display, only the starting and ending VRFs are shown.

```
!
ip vrf mvpn-cus1
 rd 200:1
 route-target export 200:1
 route-target import 200:1
 mdt default 239.1.1.1
```

```
!
ip vrf mvpn-cus2
 rd 200:2
 route-target export 200:2
 route-target import 200:2
 mdt default 239.1.1.2
!
ip vrf mvpn-cus3
 rd 200:3
 route-target export 200:3
 route-target import 200:3
 mdt default 239.1.1.3
!
...

ip vrf mvpn-cus249
 rd 200:249
 route-target export 200:249
 route-target import 200:249
 mdt default 239.1.1.249
 mdt data 239.1.1.128 0.0.0.7
```

Displaying VRF Information

To display all of the VRFs that are configured on the switch, use the **show ip vrf** command:

```
Router# show ip vrf

  Name                              Default RD          Interfaces
  green                             1:52                GigabitEthernet6/1
  red                               200:1               GigabitEthernet1/1
                                                        GigabitEthernet3/16
                                                        Loopback2

Router#
```

To display information about the MDTs that are currently configured for all MVRFs, use the **show ip pim mdt** command. The following example shows typical output for this command:

```
Router# show ip pim mdt

  MDT Group       Interface   Source                    VRF
* 227.1.0.1       Tunnel1     Loopback0                 BIDIR01
* 227.2.0.1       Tunnel2     Loopback0                 BIDIR02
* 228.1.0.1       Tunnel3     Loopback0                 SPARSE01
* 228.2.0.1       Tunnel4     Loopback0                 SPARSE02
```

Note To display information about a specific tunnel interface, use the **show interface tunnel** command. The IPv4 address for the tunnel interface is the multicast group address for the default MDT of the MVRF.

To display additional information about the MDTs, use the **show mls ip multicast mdt** command. The following example shows typical output for this command:

```
Router# show mls ip multicast mdt

  State: H - Hardware Installed, I - Install Pending, D - Delete Pending,
       Z - Zombie

              MMLS
  VRF         VPN-ID            MDT INFO                      MDT Type     State
```

```
   BIDIR01HWRP       1           (10.10.10.9, 227.1.0.1)       default      H
   BIDIR01SWRP       2           (10.10.10.9, 227.2.0.1)       default      H
  SPARSE01HWRP       3           (10.10.10.9, 228.1.0.1)       default      H
  SPARSE01SWRP       4           (10.10.10.9, 228.2.0.1)       default      H
           red       5              (6.6.6.6, 234.1.1.1)       default      H
           red       5           (131.2.1.2, 228.1.1.75)   data (send)      H
           red       5           (131.2.1.2, 228.1.1.76)   data (send)      H
           red       5           (131.2.1.2, 228.1.1.77)   data (send)      H
           red       5           (131.2.1.2, 228.1.1.78)   data (send)      H

Router#
```

To display routing information for a particular VRF, use the **show ip route vrf** command:

```
Router# show ip route vrf red

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     2.0.0.0/32 is subnetted, 1 subnets
C       2.2.2.2 is directly connected, Loopback2
     3.0.0.0/32 is subnetted, 1 subnets
B       3.3.3.3 [200/0] via 3.1.1.3, 00:20:09
C    21.0.0.0/8 is directly connected, GigabitEthernet3/16
B    22.0.0.0/8 [200/0] via 3.1.1.3, 00:20:09

Router#
```

To display information about the multicast routing table and tunnel interface for a particular MVRF, use the **show ip mroute vrf** command. The following example shows typical output for a MVRF named BIDIR01:

```
Router# show ip mroute vrf BIDIR01

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 228.1.0.1), 00:16:25/stopped, RP 10.10.10.12, flags: SJCF
  Incoming interface: Tunnel1, RPF nbr 10.10.10.12, Partial-SC
  Outgoing interface list:
    GigabitEthernet3/1.3001, Forward/Sparse-Dense, 00:16:25/00:02:49, H
(6.9.0.100, 228.1.0.1), 00:14:13/00:03:29, flags: FT
  Incoming interface: GigabitEthernet3/1.3001, RPF nbr 0.0.0.0, RPF-MFD
  Outgoing interface list:
    Tunnel1, Forward/Sparse-Dense, 00:14:13/00:02:46, H

Router#
```

Note In this example, the **show ip mroute vrf** command shows that Tunnel1 is the MDT tunnel interface (MTI) being used by this VRF.

Configuring Multicast VRF Routing

These sections describe how to configure multicast routing to support MVPN:

- Enabling IPv4 Multicast Routing Globally, page 42-15
- Enabling IPv4 Multicast VRF Routing, page 42-15
- Configuring a PIM VRF Register Message Source Address, page 42-16
- Specifying the PIM VRF RP Address, page 42-16
- Configuring an MSDP Peer, page 42-17
- Enabling IPv4 Multicast Header Storage, page 42-17
- Configuring the Maximum Number of Multicast Routes, page 42-17
- Sample Configuration, page 42-18
- Displaying IPv4 Multicast VRF Routing Information, page 42-19

Note BGP should be already configured and operational on all routers that are sending or receiving multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.

Enabling IPv4 Multicast Routing Globally

To enable IPv4 multicast routing globally, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`ip multicast-routing`**	Enables IPv4 multicast routing globally.

This example show how to enable IPv4 multicast routing globally:

```
Router# configure terminal
Router(config)# ip multicast-routing
```

Enabling IPv4 Multicast VRF Routing

To enable IPv4 multicast VRF routing, perform this task:

Command	Purpose
`Router(config)#` **`ip multicast-routing vrf`** *`vrf_name`* `[`**`distributed`**`]`	Enables IPv4 multicast VRF routing.

When enabling IPv4 multicast VRF routing, note the following information:

- *vrf_name*—Specifies a particular VRF for multicast routing. The *vrf_name* should see a VRF that has been previously created, as specified in the "Configuring a Multicast VPN Routing and Forwarding Instance" section on page 42-9.
- **distributed**—(Optional) Enables Multicast Distributed Switching (MDS).

This example show how to enable IPv4 multicast VRF routing:

```
Router# configure terminal
Router(config)# ip multicast-routing vrf blue
```

Configuring a PIM VRF Register Message Source Address

To configure a PIM VRF register message source address, perform this task:

Command	Purpose
Router(config)# **ip pim vrf** *vrf_name* **register-source** *interface_type interface_number*	(Optional) Configures a PIM VRF register message source address. You can configure a loopback interface as the source of the register messages.

This example show how to configure a PIM VRF register message source address:

```
Router(config)# ip pim vrf blue register-source loopback 3
```

Specifying the PIM VRF RP Address

To specify the PIM VRF rendezvous point (RP) address, perform this task:

Command	Purpose
Router(config)# **ip pim vrf** *vrf_name* **rp-address** *rp_address* [*access_list*] [**override**] [**bidir**]	Specifies the PIM RP IPv4 address for a (required for sparse PIM networks):

When specifying the PIM VRF RP address, note the following information:

- **vrf** *vrf_name*—(Optional) Specifies a particular VRF instance to be used.
- *rp_address*—Unicast IP address for the PIM RP router.
- *access_list*—(Optional) Number or name of an access list that defines the multicast groups for the RP.
- **override**—(Optional) In the event of conflicting RP addresses, this particular RP overrides any RP that is learned through Auto-RP.
- **bidir**—(Optional) Specifies that the multicast groups specified by the *access_list* argument are to operate in bidirectional mode. If this option is not specified, the groups operate in PIM sparse mode.
- Use bidirectional mode whenever possible, because it offers better scalability.

This example show how to specify the PIM VRF RP address:

```
Router(config)# ip pim vrf blue rp-address 198.196.100.33
```

Configuring an MSDP Peer

To configure a multicast source discovery protocol (MSDP) peer, perform this task:

Command	Purpose
Router(config)# **ip msdp vrf** *vrf_name* **peer** {*peer_name* | *peer_address*} [**connect-source** *interface_type interface_number*] [**remote-as** *ASN*]	(Optional) Configures an MSDP peer.

When configuring an MSDP peer, note the following information:

- **vrf** *vrf_name*—Specifies a particular VRF instance to be used.
- {*peer_name* | *peer_address*}—Domain Name System (DNS) name or IP address of the MSDP peer router.
- **connect-source** *interface_type interface_number*—Interface name and number for the interface whose primary address is used as the source IP address for the TCP connection.
- **remote-as** *ASN*—(Optional) Autonomous system number of the MSDP peer. This is for display-only purposes.

This example show how to configure an MSDP peer:

```
Router(config)# ip msdp peer router.cisco.com connect-source fastethernet 1/1 remote-as
109
```

Enabling IPv4 Multicast Header Storage

To enable IPv4 multicast header storage, perform this task:

Command	Purpose
Router(config)# **ip multicast vrf** *vrf_name* **cache-headers** [**rtp**]	(Optional) Enables a circular buffer to store IPv4 multicast packet headers.

When enabling IPv4 multicast header storage, note the following information:

- **vrf** *vrf_name*—Allocates a buffer for the specified VRF.
- **rtp**—(Optional) Also caches Real-Time Transport Protocol (RTP) headers.
- The buffers can be displayed with the **show ip mpacket** command.

This example show how to enable IPv4 multicast header storage:

```
Router(config)# ip multicast vrf blue cache-headers
```

Configuring the Maximum Number of Multicast Routes

To configure the maximum number of multicast routes, perform this task:

Command	Purpose
Router(config)# **ip multicast vrf** *vrf_name* **route-limit** *limit* [*threshold*]	(Optional) Configures the maximum number of multicast routes that can be added for multicast traffic.

When configuring the maximum number of routes, note the following information:

- **vrf** *vrf_name*— Enables route limiting for the specified VRF.
- *limit*—The number of multicast routes that can be added. The range is from 1 to 2147483647, with a default of 2147483647.
- *threshold*—(Optional) Number of multicast routes that can be added before a warning message occurs. The valid range is from 1 to the value of the *limit* parameter.

This example show how to configure the maximum number of multicast routes:

```
Router(config)# ip multicast vrf blue route-limit 200000 20000
```

Configuring IPv4 Multicast Route Filtering

To configure IPV4 multicast route filtering, perform this task:

Command	Purpose
Router(config)# **ip multicast mrinfo-filter** *access_list*	(Optional) Configures IPV4 multicast route filtering with an access list. The *access_list* parameter can be the name or number of a access list.

This example show how to configure IPV4 multicast route filtering:

```
Router(config)# ip multicast mrinfo-filter 101
```

Sample Configuration

The following excerpt from a configuration file shows the minimum configuration that is needed to support multicast routing for a range of VRFs. To simplify the display, only the starting and ending VRFs are shown.

```
!
ip multicast-routing
ip multicast-routing vrf lite
ip multicast-routing vrf vpn201
ip multicast-routing vrf vpn202

...

ip multicast-routing vrf vpn249
ip multicast-routing vrf vpn250
ip multicast cache-headers


...

ip pim rp-address 192.0.1.1
ip pim vrf lite rp-address 104.1.1.2
ip pim vrf vpn201 rp-address 192.200.1.1
ip pim vrf vpn202 rp-address 192.200.2.1

...

ip pim vrf vpn249 rp-address 192.200.49.6
ip pim vrf vpn250 rp-address 192.200.50.6
...
```

Displaying IPv4 Multicast VRF Routing Information

To display the known PIM neighbors for a particular MVRF, use the **show ip pim vrf neighbor** command:

```
Router# show ip pim vrf 98 neighbor

PIM Neighbor Table
Neighbor          Interface                 Uptime/Expires    Ver   DR
Address                                                             Prio/Mode
40.60.0.11        Tunnel96                  00:00:31/00:01:13 v2    1 / S
40.50.0.11        Tunnel96                  00:00:54/00:00:50 v2    1 / S

Router#
```

Configuring Interfaces for Multicast Routing to Support MVPN

These sections describe how to configure interfaces for multicast routing to support MVPN:

- Multicast Routing Configuration Overview, page 42-19
- Configuring PIM on an Interface, page 42-20
- Configuring an Interface for IPv4 VRF Forwarding, page 42-20
- Sample Configuration, page 42-21

Multicast Routing Configuration Overview

Protocol Independent Multicast (PIM) must be configured on all interfaces that are being used for IPv4 multicast traffic. In a VPN multicast environment, you should enable PIM on at least all of the following interfaces:

- Physical interface on a provider edge (PE) router that is connected to the backbone.
- Loopback interface that is used for BGP peering.
- Loopback interface that is used as the source for the sparse PIM rendezvous point (RP) router address.

In addition, you must also associate MVRFs with those interfaces over which they are going to forward multicast traffic.

BGP should be already configured and operational on all routers that are sending or receiving multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.

Configuring PIM on an Interface

To configure PIM on an interface, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface`** *`type`* `{`*`slot/port`* `|` *`number`*`}`	Enters interface configuration mode for the specified interface.
Step 3	`Router(config-if)#` **`ip pim`** `{`**`dense-mode`** `|` **`sparse-mode`** `|` **`sparse-dense-mode`**`}`	Enables PIM on the interface.

When configuring PIM on an interface, note the following information:

- You can use one of these interface types:
  - A physical interface on a provider edge (PE) router that is connected to the backbone.
  - A loopback interface that is used for BGP peering.
  - A loopback interface that is used as the source for the sparse PIM network rendezvous point (RP) address.
- These are the PIM modes:
  - **dense-mode**—Enables dense mode of operation.
  - **sparse-mode**—Enables sparse mode of operation.
  - **sparse-dense-mode**—Enables sparse mode if the multicast group has an RP router defined, or enables dense mode if an RP router is not defined.
- Use **sparse-mode** for the physical interfaces of all PE routers that are connected to the backbone, and on all loopback interfaces that are used for BGP peering or as the source for RP addressing.

This example shows how to configure PIM sparse mode on a physical interface:

```
Router# configure terminal
Router(config)# interface gigabitethernet 10/1
Router(config-if)# ip pim sparse-mode
```

This example shows how to configure PIM sparse mode on a loopback interface:

```
Router# configure terminal
Router(config)# interface loopback 2
Router(config-if)# ip pim sparse-mode
```

Configuring an Interface for IPv4 VRF Forwarding

To configure an interface for IPv4 VRF forwarding, perform this task:

Command	Purpose
`Router(config-if)#` **`ip vrf forwarding`** *`vrf_name`*	(Optional) Associates the specified VRF routing and forwarding tables with the interface. If this is not specified, the interface defaults to using the global routing table. **Note** Entering this command on an interface removes the IP address, so reconfigure the IP address.

This example shows how to configure the interface for VRF blue forwarding:

```
Router(config-if)# ip vrf forwarding blue
```

Sample Configuration

The following excerpt from a configuration file shows the interface configuration, along with the associated MVRF configuration, to enable multicast traffic over a single MVRF:

```
ip multicast-routing vrf blue
ip multicast-routing

ip vrf blue
 rd 100:27
 route-target export 100:27
 route-target import 100:27
 mdt default 239.192.10.2

interface GigabitEthernet1/1
 description blue connection
 ip vrf forwarding blue
 ip address 192.168.2.26 255.255.255.0
 ip pim sparse-mode

interface GigabitEthernet1/15
 description Backbone connection
 ip address 10.8.4.2 255.255.255.0
 ip pim sparse-mode

ip pim vrf blue rp-address 192.7.25.1
ip pim rp-address 10.1.1.1
```

Sample Configurations for MVPN

This section contains the following sample configurations for the MVPN feature:

- MVPN Configuration with Default MDTs Only, page 42-21
- MVPN Configuration with Default and Data MDTs, page 42-23

MVPN Configuration with Default MDTs Only

The following excerpt from a configuration file shows the lines that are related to the MVPN configuration for three MVRFs. (The required BGP configuration is not shown.)

```
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service compress-config
!
hostname MVPN Router
!
boot system flash slot0:
logging snmp-authfail
!
ip subnet-zero
!
```

```
!
no ip domain-lookup
ip host tftp 223.255.254.238
!
ip vrf mvpn-cus1
 rd 200:1
 route-target export 200:1
 route-target import 200:1
 mdt default 239.1.1.1
!
ip vrf mvpn-cus2
 rd 200:2
 route-target export 200:2
 route-target import 200:2
 mdt default 239.1.1.2
!
ip vrf mvpn-cus3
 rd 200:3
 route-target export 200:3
 route-target import 200:3
 mdt default 239.1.1.3
!
ip multicast-routing
ip multicast-routing vrf mvpn-cus1
ip multicast-routing vrf mvpn-cus2
ip multicast-routing vrf mvpn-cus3
ip multicast multipath
frame-relay switching
mpls label range 4112 262143
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls ldp explicit-null
mpls traffic-eng tunnels
tag-switching tdp discovery directed-hello accept from 1
tag-switching tdp router-id Loopback0 force
mls ip multicast replication-mode ingress
mls ip multicast flow-stat-timer 9
mls ip multicast bidir gm-scan-interval 10
mls flow ip destination
no mls flow ipv6
mls rate-limit unicast cef glean 10 10
mls qos
mls cef error action freeze

...

vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
vlan 2001-2101,3501-3700,4001,4051-4080,4093
!
!
!
interface Loopback0
 ip address 201.252.1.14 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback1
 ip address 209.255.255.14 255.255.255.255
!
interface Loopback10
 ip vrf forwarding mvpn-cus1
 ip address 210.101.255.14 255.255.255.255
!
```

```
interface Loopback11
 ip vrf forwarding mvpn-cus1
 ip address 210.111.255.14 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback12
 ip vrf forwarding mvpn-cus1
 ip address 210.112.255.14 255.255.255.255

...

!
interface GigabitEthernet3/3
 mtu 9216
 ip vrf forwarding mvpn-cus3
 ip address 172.10.14.1 255.255.255.0
 ip pim sparse-dense-mode
!

...

!
interface GigabitEthernet3/19
 ip vrf forwarding mvpn-cus2
 ip address 192.16.4.1 255.255.255.0
 ip pim sparse-dense-mode
 ip igmp static-group 229.1.1.1
 ip igmp static-group 229.1.1.2
 ip igmp static-group 229.1.1.4
!
interface GigabitEthernet3/20
 ip vrf forwarding mvpn-cus1
 ip address 192.16.1.1 255.255.255.0
 ip pim sparse-dense-mode
!
...
```

MVPN Configuration with Default and Data MDTs

The following sample configuration includes three MVRFs that have been configured for both default and data MDTs. Only the configuration that is relevant to the MVPN configuration is shown.

```
...
!
ip vrf v1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 mdt default 226.1.1.1
 mdt data 226.1.1.128 0.0.0.7 threshold 1
!
ip vrf v2
 rd 2:2
 route-target export 2:2
 route-target import 2:2
 mdt default 226.2.2.1
 mdt data 226.2.2.128 0.0.0.7
!
ip vrf v3
 rd 3:3
 route-target export 3:3
 route-target import 3:3
```

```
 mdt default 226.3.3.1
 mdt data 226.3.3.128 0.0.0.7
!
ip vrf v4
 rd 155.255.255.1:4
 route-target export 155.255.255.1:4
 route-target import 155.255.255.1:4
 mdt default 226.4.4.1
 mdt data 226.4.4.128 0.0.0.7
!
ip multicast-routing
ip multicast-routing vrf v1
ip multicast-routing vrf v2
ip multicast-routing vrf v3
ip multicast-routing vrf v4
mpls label protocol ldp
mpls ldp logging neighbor-changes
tag-switching tdp router-id Loopback1
mls ip multicast replication-mode ingress
mls ip multicast bidir gm-scan-interval 10
no mls flow ip
no mls flow ipv6
mls cef error action freeze
!
!
!
!
!

...

vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
!
interface Loopback1
 ip address 155.255.255.1 255.255.255.255
 ip pim sparse-mode
!
interface Loopback4
 ip vrf forwarding v4
 ip address 155.255.4.4 255.255.255.255
 ip pim sparse-mode
!
interface Loopback11
 ip vrf forwarding v1
 ip address 155.255.255.11 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback22
 ip vrf forwarding v2
 ip address 155.255.255.22 255.255.255.255
 ip pim sparse-mode
!
interface Loopback33
 ip vrf forwarding v3
 ip address 155.255.255.33 255.255.255.255
 ip pim sparse-mode
!
interface Loopback44
 no ip address
!
interface Loopback111
 ip vrf forwarding v1
```

```
 ip address 1.1.1.1 255.255.255.252
 ip pim sparse-dense-mode
 ip ospf network point-to-point
!
interface GigabitEthernet1/1
 description Gi1/1 - 155.50.1.155 255.255.255.0 - peer dut50 - mpls
 mtu 9216
 ip address 155.50.1.155 255.255.255.0
 ip pim sparse-mode
 tag-switching ip
!
interface GigabitEthernet1/2
 ip vrf forwarding v1
 ip address 155.1.2.254 255.255.255.0
 ip pim sparse-mode
!
interface GigabitEthernet1/3
 description Gi1/3 - 185.155.1.155/24 - vrf v1 stub peer 185.Gi1/3
 ip vrf forwarding v1
 ip address 185.155.1.155 255.255.255.0
 ip pim sparse-mode
!

...

!
interface GigabitEthernet1/48
 ip vrf forwarding v1
 ip address 157.155.1.155 255.255.255.0
 ip pim bsr-border
 ip pim sparse-dense-mode
!
interface GigabitEthernet6/1
 no ip address
 shutdown
!
interface GigabitEthernet6/2
 ip address 9.1.10.155 255.255.255.0
 media-type rj45
!
interface Vlan1
 no ip address
 shutdown
!
router ospf 11 vrf v1
 router-id 155.255.255.11
 log-adjacency-changes
 redistribute connected subnets tag 155
 redistribute bgp 1 subnets tag 155
 network 1.1.1.0 0.0.0.3 area 155
 network 155.255.255.11 0.0.0.0 area 155
 network 155.0.0.0 0.255.255.255 area 155
 network 157.155.1.0 0.0.0.255 area 0
!
router ospf 22 vrf v2
 router-id 155.255.255.22
 log-adjacency-changes
 network 155.255.255.22 0.0.0.0 area 155
 network 155.0.0.0 0.255.255.255 area 155
 network 157.155.1.0 0.0.0.255 area 0
!
router ospf 33 vrf v3
 router-id 155.255.255.33
 log-adjacency-changes
```

```
 network 155.255.255.33 0.0.0.0 area 155
!
router ospf 1
 log-adjacency-changes
 network 155.50.1.0 0.0.0.255 area 0
 network 155.255.255.1 0.0.0.0 area 155
!
router bgp 1
 bgp router-id 155.255.255.1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 175.255.255.1 remote-as 1
 neighbor 175.255.255.1 update-source Loopback1
 neighbor 185.255.255.1 remote-as 1
 neighbor 185.255.255.1 update-source Loopback1
 !
 address-family vpnv4
 neighbor 175.255.255.1 activate
 neighbor 175.255.255.1 send-community extended
 neighbor 185.255.255.1 activate
 neighbor 185.255.255.1 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf v4
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf v3
 redistribute ospf 33
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf v2
 redistribute ospf 22
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf v1
 redistribute ospf 11
 no auto-summary
 no synchronization
 exit-address-family
!
ip classless
ip route 9.255.254.1 255.255.255.255 9.1.10.254
no ip http server
ip pim bidir-enable
ip pim rp-address 50.255.2.2 MCAST.MVPN.MDT.v2 override bidir
ip pim rp-address 50.255.3.3 MCAST.MVPN.MDT.v3 override bidir
ip pim rp-address 50.255.1.1 MCAST.MVPN.MDT.v1 override bidir
ip pim vrf v1 spt-threshold infinity
ip pim vrf v1 send-rp-announce Loopback11 scope 16 group-list MCAST.GROUP.BIDIR bidir
ip pim vrf v1 send-rp-discovery Loopback11 scope 16
ip pim vrf v1 bsr-candidate Loopback111 0
ip msdp vrf v1 peer 185.255.255.11 connect-source Loopback11
ip msdp vrf v1 cache-sa-state
!
!
ip access-list standard MCAST.ANYCAST.CE
 permit 2.2.2.2
ip access-list standard MCAST.ANYCAST.PE
```

```
 permit 1.1.1.1
ip access-list standard MCAST.BOUNDARY.VRF.v1
 deny   226.192.1.1
 permit any
ip access-list standard MCAST.GROUP.BIDIR
 permit 226.192.0.0 0.0.255.255
ip access-list standard MCAST.GROUP.SPARSE
 permit 226.193.0.0 0.0.255.255
ip access-list standard MCAST.MVPN.BOUNDARY.DATA.MDT
 deny   226.1.1.128
 permit any
ip access-list standard MCAST.MVPN.MDT.v1
 permit 226.1.0.0 0.0.255.255
ip access-list standard MCAST.MVPN.MDT.v2
 permit 226.2.0.0 0.0.255.255
ip access-list standard MCAST.MVPN.MDT.v3
 permit 226.3.0.0 0.0.255.255
ip access-list standard MCAST.MVPN.RP.v4
 permit 227.0.0.0 0.255.255.255
!
access-list 1 permit 226.1.1.1
access-list 2 deny   226.1.1.1
access-list 2 permit any
...
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 8

Quality of Service

Configuring PFC QoS

This chapter describes how to configure quality of service (QoS) as implemented on the Policy Feature Card (PFC) and Distributed Forwarding Cards (DFCs) in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html
- For information about Auto-QoS, see Chapter 44, “Using AutoQoS.”
- For information about QoS and MPLS, see Chapter 45, “Configuring MPLS QoS.”
- QoS in Cisco IOS Release 12.2SX (PFC QoS) uses some Cisco IOS modular QoS CLI (MQC). Because PFC QoS is implemented in hardware, it supports only a subset of the MQC syntax.
- The PFC3 does not support Network-Based Application Recognition (NBAR).
- With releases earlier than Release 12.2(33)SXJ3, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits (CSCts82932).

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains these sections:

Understanding PFC QoS

The term “PFC QoS” refers to QoS in Cisco IOS Release 12.2SX. PFC QoS is implemented on various switch components in addition to the PFC and any DFCs. These sections describe how PFC QoS works:

- Port Types Supported by PFC QoS, page 43-2
- Overview, page 43-2
- Component Overview, page 43-5
- Understanding Classification and Marking, page 43-14
- Understanding Port-Based Queue Types, page 43-20

Port Types Supported by PFC QoS

The PFC does not provide QoS for FlexWAN module ports. See this publication for information about FlexWAN module QoS features:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexwan-config-guide.html

In all releases, PFC QoS supports *LAN ports*. LAN ports are Ethernet ports on Ethernet switching modules.

Overview

Typically, networks operate on a *best-effort* delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

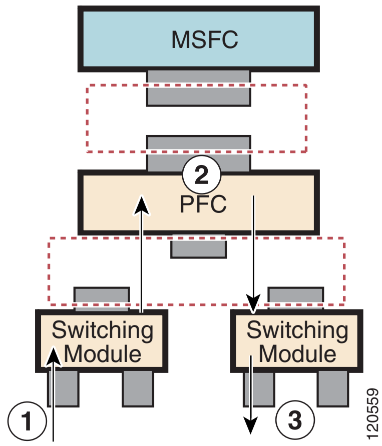
QoS makes network performance more predictable and bandwidth utilization more effective. QoS selects (classifies) network traffic, uses or assigns QoS labels to indicate priority, makes the packets comply with the configured resource usage limits (polices the traffic and marks the traffic), and provides congestion avoidance where resource contention exists.

PFC QoS classification, policing, marking, and congestion avoidance is implemented in hardware on the PFC, DFCs, and in LAN switching module port Application Specific Integrated Circuits (ASICs).

Note Cisco IOS Release 12.2SX does not support all of the MQC features (for example, Committed Access Rate (CAR)) for traffic that is Layer 3 switched or Layer 2 switched in hardware. Because queuing is implemented in the port ASICs, Cisco IOS Release 12.2SX does not support MQC-configured queuing.

Figure 43-1 shows an overview of QoS processing in a switch supported by Cisco IOS Release 12.2SX.

Figure 43-1 PFC QoS Feature Processing Overview



The PFC QoS features are applied in this order:

1. Ingress port PFC QoS features:
   - Port trust state—In PFC QoS, *trust* means to accept as valid and use as the basis of the initial internal DSCP value. Ports are untrusted by default, which sets the initial internal DSCP value to zero. You can configure ports to trust one of three types of received QoS values: CoS, IP precedence, or DSCP.
   - Layer 2 CoS remarking—PFC QoS applies Layer 2 CoS remarking, which marks the incoming frame with the port CoS value, in these situations:

     —If the traffic is not in an ISL, 802.1Q, or 802.1p frame.

     —If a port is configured as untrusted.
   - Congestion avoidance—If you configure an Ethernet LAN port to trust CoS or DSCP, QoS classifies the traffic on the basis of its Layer 2 CoS value or its Layer 3 DSCP value and assigns it to an ingress queue to provide congestion avoidance. Layer 3 DSCP-based queue mapping is available only on WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports.
2. PFC and DFC QoS features:
   - Internal DSCP—On the PFC and DFCs, QoS associates an internal DSCP value with all traffic to classify it for processing through the system. There is an initial internal DSCP based on the traffic trust state and a final internal DSCP. The final internal DSCP can be the same as the initial value or an MQC policy map can set it to a different value.
   - MQC policy maps—MQC policy maps can do one or more of these operations:

     —Change the trust state of the traffic (bases the internal DSCP value on a different QoS label)

     —Set the initial internal DSCP value (only for traffic from untrusted ports)

     —Mark the traffic

     —Police the traffic
3. Egress Ethernet LAN port QoS features:
   - Layer 3 DSCP marking with the final internal DSCP (optional)
   - Layer 2 CoS marking mapped from the final internal DSCP

- Layer 2 CoS-based and Layer 3 DSCP-based congestion avoidance. (Layer 3 DSCP-based queue mapping is available only on WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports.)

These figures provide more detail about the relationship between QoS and the switch components:

- Figure 43-2, Traffic Flow and PFC QoS Features with a PFC3
- Figure 43-3, PFC QoS Features and Component Overview

Figure 43-2 shows traffic flow and PFC QoS features with a PFC3.

Figure 43-2 Traffic Flow and PFC QoS Features with a PFC3

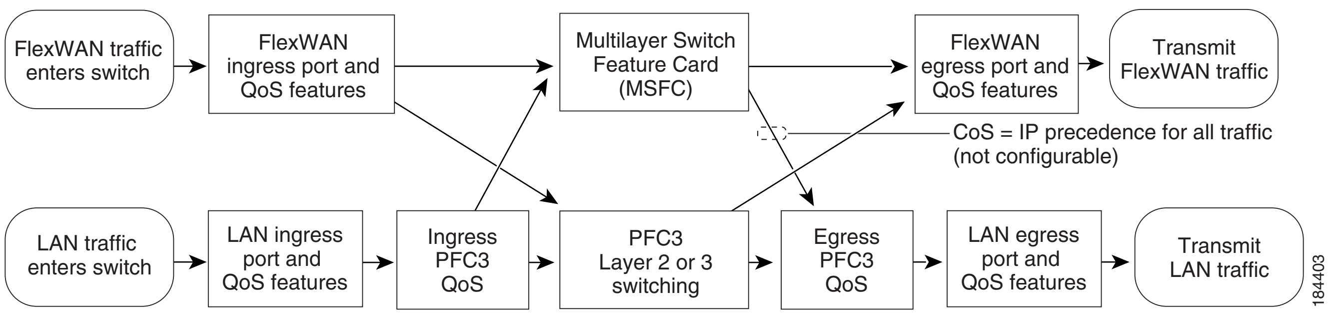
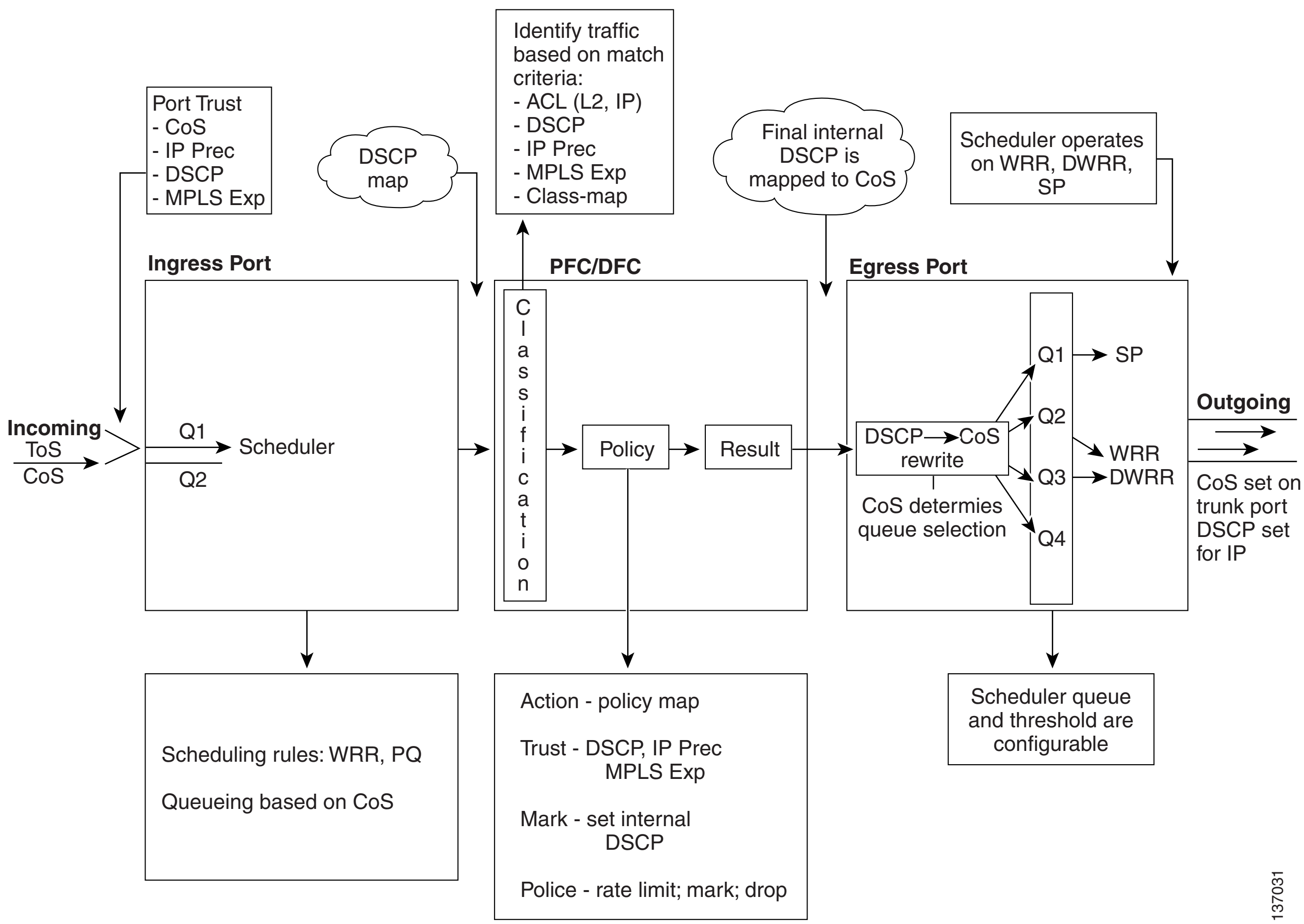


Figure 43-2 shows how traffic flows through the PFC QoS features with PFC3:

- Traffic can enter on any type of port and exit on any type of port.
- DFCs implement PFC QoS locally on switching modules.
- For FlexWAN module traffic:
  - Ingress FlexWAN QoS features can be applied to FlexWAN ingress traffic.
  - Ingress FlexWAN traffic can be Layer 3-switched by the PFC3 or routed in software by the route processor (RP).
  - Egress PFC QoS is not applied to FlexWAN ingress traffic.
  - Egress FlexWAN QoS can be applied to FlexWAN egress traffic.
- For LAN-port traffic:
  - Ingress LAN port QoS features can be applied to LAN port ingress traffic.
  - Ingress PFC QoS can be applied to LAN port ingress traffic.
  - Ingress LAN port traffic can be Layer 2- or Layer 3-switched by the PFC3 or routed in software by the RP.
  - Egress PFC QoS and egress LAN port QoS can be applied to LAN port egress traffic.

Figure 43-3 PFC QoS Features and Component Overview



Component Overview

These sections provide more detail about the role of the following components in PFC QoS decisions and processes:

- Ingress LAN Port PFC QoS Features, page 43-5
- PFC and DFC QoS Features, page 43-7
- PFC QoS Egress Port Features, page 43-11

Ingress LAN Port PFC QoS Features

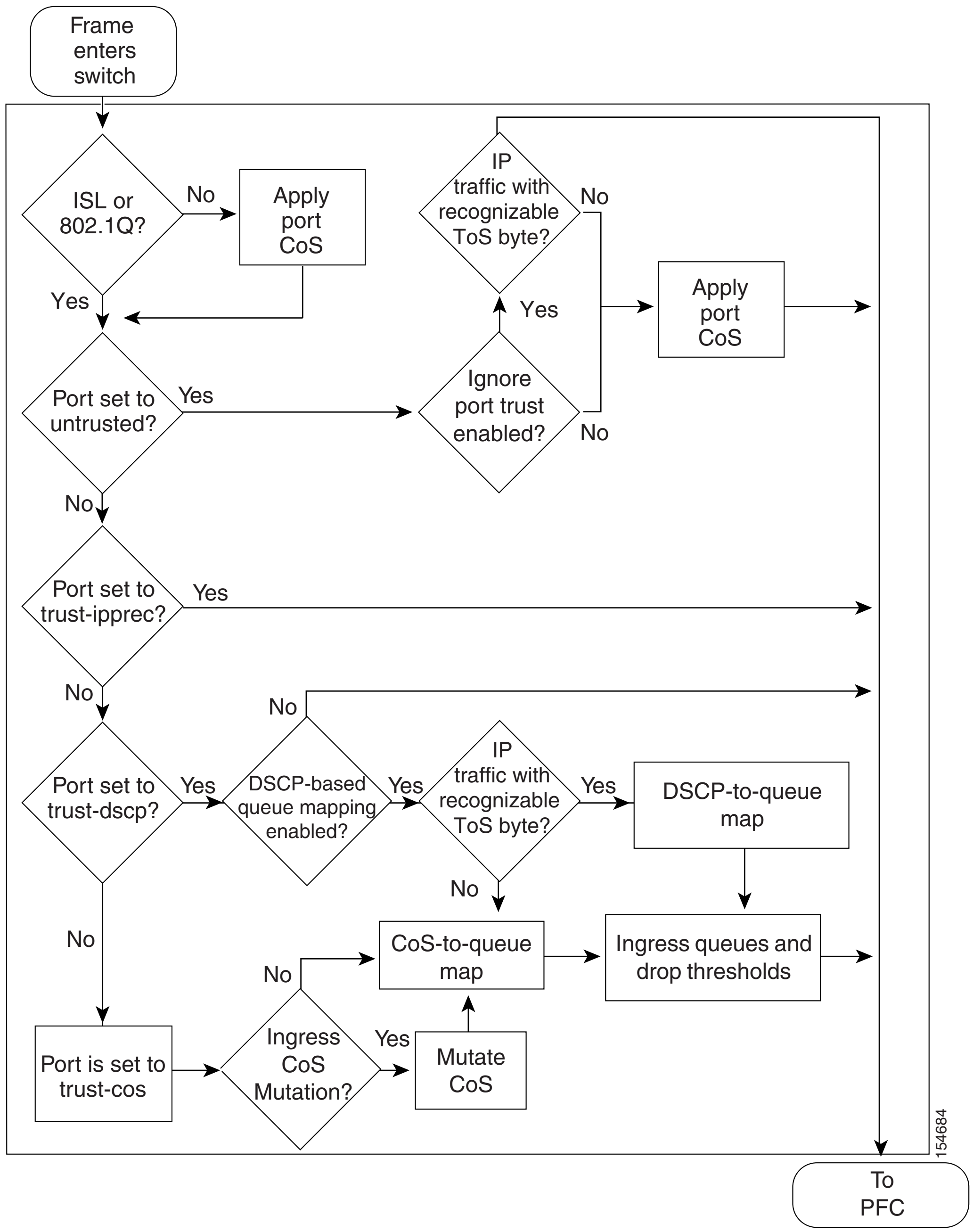
These sections provide an overview of the ingress port QoS features:

- Flowchart of Ingress LAN Port PFC QoS Features, page 43-6
- Port Trust, page 43-7
- Ingress Congestion Avoidance, page 43-7

Flowchart of Ingress LAN Port PFC QoS Features

Figure 43-4 shows how traffic flows through the ingress LAN port PFC QoS features.

Figure 43-4 Ingress LAN Port PFC QoS Features



Note

- Ingress CoS mutation is supported only on 802.1Q tunnel ports.
- DSCP-based queue mapping is supported only on WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports.

Port Trust

In PFC QoS, *trust* means to accept as valid and use as the basis of the initial internal DSCP value. You can configure ports as untrusted or you can configure them to trust these QoS values:

- Layer 2 CoS
  - A port configured to trust CoS is called a trust CoS port.
  - Traffic received through a trust CoS port or configured by a policy map to trust CoS is called trust CoS traffic.

Note Not all traffic carries a CoS value. Only ISL, 802.1Q, and 802.1P traffic carries a CoS value. PFC QoS applies the port CoS value to any traffic that does not carry a CoS value. On untrusted ports, PFC QoS applies the port CoS value to all traffic, overwriting any received CoS value.

- IP precedence
  - A port configured to trust IP precedence is called a trust IP precedence port.
  - Traffic received through a trust IP precedence port or configured by a policy map to trust IP precedence is called trust IP precedence traffic.
- DSCP
  - A port configured to trust DSCP is called a trust DSCP port.
  - Traffic received through a trust DSCP port or configured by a policy map to trust DSCP is called trust DSCP traffic.

Traffic received through an untrusted port is called untrusted traffic.

Ingress Congestion Avoidance

PFC QoS implements congestion avoidance on trust CoS ports. On a trust CoS port, QoS classifies the traffic on the basis of its Layer 2 CoS value and assigns it to an ingress queue to provide congestion avoidance. You can configure WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE trust DSCP ports to use received DSCP values for congestion avoidance. See the "Ingress Classification and Marking at Trust CoS LAN Ports" section on page 43-15 for more information about ingress congestion avoidance.

PFC and DFC QoS Features

These sections describe PFCs and DFCs as they relate to QoS:

Supported Policy Feature Cards

The policy feature card (PFC) is a daughter card on the supervisor engine. The PFC provides QoS in addition to other functionality. The following PFCs are supported in Cisco IOS Release 12.2SX:

- PFC3A on the Supervisor Engine 720
- PFC3B on the Supervisor Engine 720 and Supervisor Engine 32
- PFC3BXL on the Supervisor Engine 720
- PFC3C on the Cisco ME 6500 Series Ethernet switches and the Supervisor Engine 720-10GE
- PFC3CXL on the Supervisor Engine 720-10GE

Supported Distributed Forwarding Cards

The PFC sends a copy of the QoS policies to the distributed forwarding card (DFC) to provide local support for the QoS policies, which enables the DFCs to support the same QoS features that the PFC supports.

These DFCs are supported on the Catalyst 6500 series switches:

- For use on dCEF256 and CEF256 modules with a Supervisor Engine 720:
  - WS-F6K-DFC3A
  - WS-F6K-DFC3B
  - WS-F6K-DFC3BXL
- For use on CEF720 modules with a Supervisor Engine 720:
  - WS-F6700-DFC3A
  - WS-F6700-DFC3B
  - WS-F6700-DFC3BXL
- For use on CEF720 modules with Supervisor Engine 720 and Supervisor Engine 720-10GE:
  - WS-F6700-DFC3CXL
  - WS-F6700-DFC3C

PFC and DFC QoS Feature List and Flowchart

Table 43-1 lists the QoS features supported on the different versions of PFCs and DFCs.

Table 43-1 QoS Features Supported on PFCs and DFCs

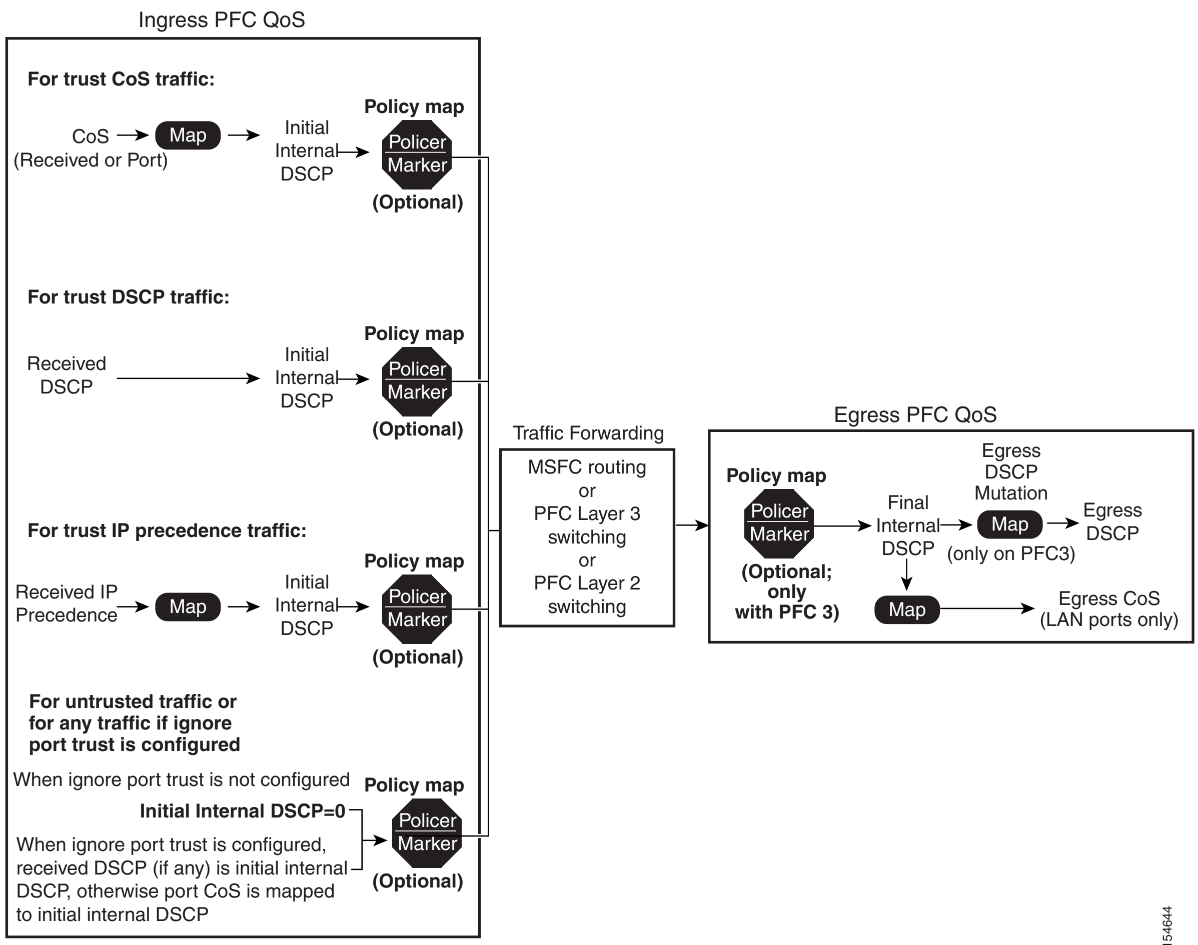
Feature	PFC3A and DFC3A	PFC3B and DFC3B	PFC3BXL and DFC3BXL	PFC3C and DFC3C	PFC3CXL and DFC3CXL
Support for DFCs	Yes	Yes	Yes	Yes	Yes
Flow granularity	Source Destination	Source Destination	Source Destination	Source Destination	Source Destination
QoS ACLs	IP, MAC	IP, MAC	IP, MAC	IP, MAC	IP, MAC
DSCP transparency **Note** Enabling DSCP transparency disables egress ToS rewrite.	Optional	Optional	Optional	Optional	Optional
Egress ToS rewrite	Optional	Optional	Optional	Optional	Optional

Table 43-1 QoS Features Supported on PFCs and DFCs (continued)

Feature	PFC3A and DFC3A	PFC3B and DFC3B	PFC3BXL and DFC3BXL	PFC3C and DFC3C	PFC3CXL and DFC3CXL
Policing:					
Ingress aggregate policers	Yes	Yes	Yes	Yes	Yes
Egress aggregate policers	Yes	Yes	Yes	Yes	Yes
Number of aggregate policers	1023 configurable	1023 configurable	1023 configurable	1023 configurable	1023 configurable
Microflow policers	64 rates	64 rates	64 rates	64 rates	64 rates
Number of flows per Microflow policer	64,000	110,000	240,000	110,000	240,000
Unit of measure for policer statistics	Bytes	Bytes	Bytes	Bytes	Bytes
Basis of policer operation	Layer 2 length	Layer 2 length	Layer 2 length	Layer 2 length	Layer 2 length

Figure 43-5 shows how traffic flows through the QoS features on the PFC and DFCs.

Figure 43-5 QoS Features on the PFC and DFCs



Note The DSCP transparency feature makes writing the egress DSCP value into the Layer 3 ToS byte optional.

Internal DSCP Values

During processing, PFC QoS represents the priority of all traffic (including non-IP traffic) with an internal DSCP value.

Initial Internal DSCP Value

On the PFC, before any marking or policing takes place, PFC QoS derives the initial internal DSCP value as follows:

- For untrusted traffic, when ignore port trust is not enabled, PFC QoS sets the initial internal DSCP value to zero for both tagged and untagged untrusted traffic.
- For untrusted traffic, when ignore port trust is enabled, PFC QoS does the following:
  - For IP traffic, PFC QoS uses the received DSCP value as the initial internal DSCP value.
  - For traffic without a recognizable ToS byte, PFC QoS maps the port CoS value to the initial internal DSCP value.
- For trust CoS traffic, when ignore port trust is enabled, PFC QoS does the following:
  - For IP traffic, PFC QoS uses the received DSCP value as the initial internal DSCP value.

    Note For trust CoS traffic, when ignore port trust is enabled, PFC QoS does not use the received CoS value in tagged IP traffic. When ignore port trust is disabled, PFC QoS uses the received CoS value in tagged IP traffic.

  - For tagged traffic without a recognizable ToS byte, PFC QoS maps the received CoS value to the initial internal DSCP value.
  - For untagged traffic without a recognizable ToS byte, PFC QoS maps the port CoS value to the initial internal DSCP value.
- For trust IP precedence traffic, PFC QoS does the following:
  - For IP traffic, PFC QoS maps the received IP precedence value to the initial internal DSCP value.
  - For tagged traffic without a recognizable ToS byte, PFC QoS maps the received CoS value to the initial internal DSCP value.
  - For untagged traffic without a recognizable ToS byte, PFC QoS maps the port CoS value to the initial internal DSCP value.
- For trust DSCP traffic, PFC QoS, PFC QoS does the following:
  - For IP traffic, PFC QoS uses the received DSCP value as the initial internal DSCP value.
  - For tagged traffic without a recognizable ToS byte, PFC QoS maps the received CoS value to the initial internal DSCP value.
  - For untagged traffic without a recognizable ToS byte, PFC QoS maps the port CoS value to the initial internal DSCP value.

For trust CoS traffic and trust IP precedence traffic, PFC QoS uses configurable maps to derive the initial internal 6-bit DSCP value from CoS or IP precedence, which are 3-bit values.

Final Internal DSCP Value

Policy marking and policing on the PFC can change the initial internal DSCP value to a final internal DSCP value, which is then used for all subsequently applied QoS features.

Port-Based PFC QoS and VLAN-Based PFC QoS

You can configure each ingress LAN port for either physical port-based PFC QoS (default) or VLAN-based PFC QoS and attach a policy map to the selected interface.

On ports configured for port-based PFC QoS, you can attach a policy map to the ingress LAN port as follows:

- On a nontrunk ingress LAN port configured for port-based PFC QoS, all traffic received through the port is subject to the policy map attached to the port.
- On a trunking ingress LAN port configured for port-based PFC QoS, traffic in all VLANs received through the port is subject to the policy map attached to the port.

On a nontrunk ingress LAN port configured for VLAN-based PFC QoS, traffic received through the port is subject to the policy map attached to the port's VLAN.

On a trunking ingress LAN port configured for VLAN-based PFC QoS, traffic received through the port is subject to the policy map attached to the traffic's VLAN.

Session-Based PFC QoS

In Cisco IOS Software Release 12.2(33)SXI and later releases, you can configure the dynamic delivery of a policy map to an interface by the AAA server when a user authenticates on that interface. This feature allows per-user or per-session QoS at the interface level, so that a user who connects by different interfaces at different times will always receive the same QoS treatment.

For each user of session-based QoS, you must set these attribute-value (AV) pairs on the AAA server by using RADIUS *cisco-av-pair* vendor-specific attributes (VSAs):

- cisco-avpair = "ip:sub-policy-In=*in_policy_name*"
- cisco-avpair = "ip:sub-policy-Out=*out_policy_name*"

The *in_policy_name* and *out_policy_name* arguments are the names of the ingress and egress QoS policy maps to be applied to an interface when a user authenticates on that interface. The policy maps will be removed from the interface when the user logs off and the session is terminated.

For session-based QoS configuration information, see the "Configuring Dynamic Per-Session Attachment of a Policy Map" section on page 43-83.

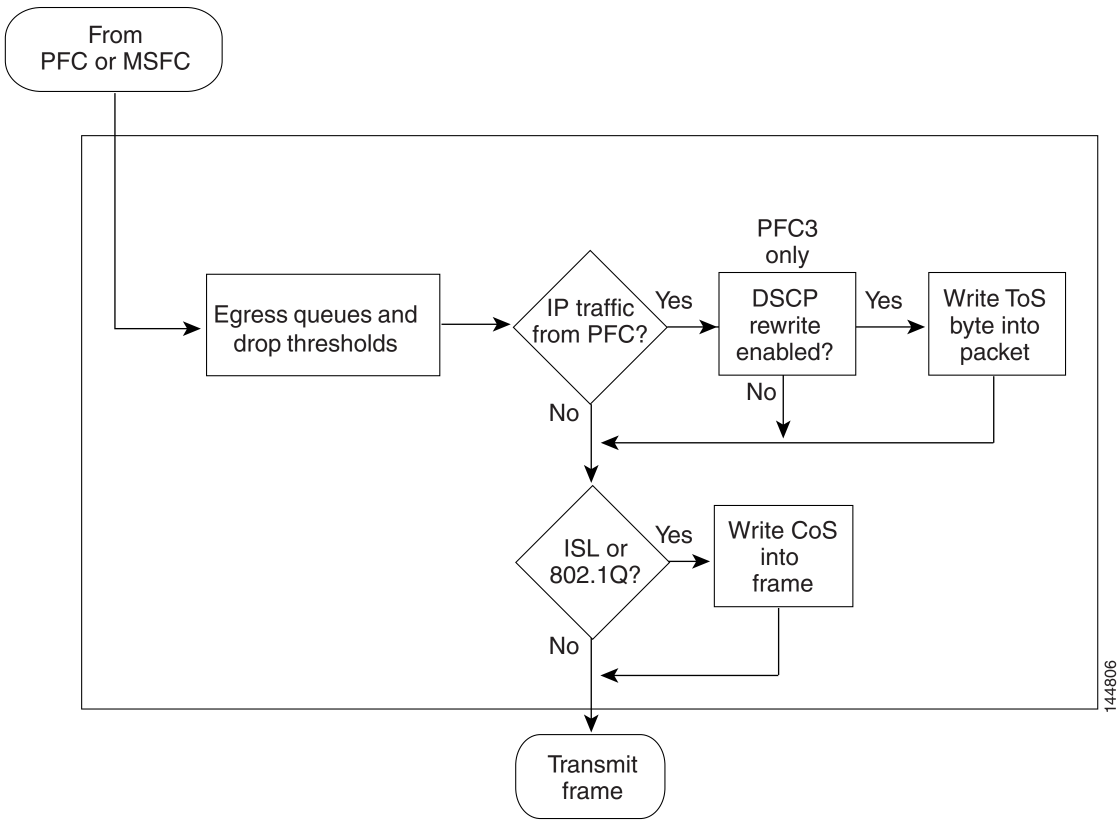
PFC QoS Egress Port Features

These sections describe PFC QoS egress port features:

Flowchart of PFC QoS Egress LAN Port Features

Figure 43-6 shows how traffic flows through the QoS features on egress LAN ports.

Figure 43-6 Egress LAN Port Scheduling, Congestion Avoidance, and Marking



Egress CoS Values

For all egress traffic, PFC QoS uses a configurable map to derive a CoS value from the final internal DSCP value associated with the traffic. PFC QoS sends the derived CoS value to the egress LAN ports for use in classification and congestion avoidance and to be written into ISL and 802.1Q frames.

Note You can configure WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports to use the final internal DSCP value for egress LAN port classification and congestion avoidance (see the "Configuring DSCP-Based Queue Mapping" section on page 43-101).

Egress DSCP Mutation

You can configure 15 egress DSCP mutation maps to mutate the internal DSCP value before it is written in the egress ToS byte. You can attach egress DSCP mutation maps to any interface on which PFC QoS supports egress QoS.

Note If you configure egress DSCP mutation, PFC QoS does not derive the egress CoS value from the mutated DSCP value.

Egress ToS Byte

Except when DSCP transparency is enabled, PFC QoS creates a ToS byte for egress IP traffic from the final internal or mutated DSCP value and sends it to the egress port to be written into IP packets. For trust DSCP and untrusted IP traffic, the ToS byte includes the original two least-significant bits from the received ToS byte.

The internal or mutated DSCP value can mimic an IP precedence value (see the "IP Precedence and DSCP Values" section on page 43-58).

Egress PFC QoS Interfaces

You can attach an output policy map to a Layer 3 interface (either a LAN port configured as a Layer 3 interface or a VLAN interface) to apply a policy map to egress traffic.

Note

- Output policies do not support microflow policing.
- You cannot apply microflow policing to ARP traffic.
- You cannot set a trust state in an output policy.

Egress ACL Support for Remarked DSCP

Note Egress ACL support for remarked DSCP is also known as packet recirculation.

The PFC3 supports egress ACL support for remarked DSCP, which enables IP precedence-based or DSCP-based egress QoS filtering to use any IP precedence or DSCP policing or marking changes made by ingress PFC QoS.

Without egress ACL support for remarked DSCP, egress QoS filtering uses received IP precedence or DSCP values; it does not use any IP precedence or DSCP changes made by ingress PFC QoS as the result of policing or marking.

The PFC3 provides egress PFC QoS only for Layer 3-switched and routed traffic on egress Layer 3 interfaces (either LAN ports configured as Layer 3 interfaces or VLAN interfaces).

You configure egress ACL support for remarked DSCP on ingress Layer 3 interfaces (either LAN ports configured as Layer 3 interfaces or VLAN interfaces).

On interfaces where egress ACL support for remarked DSCP is configured, the PFC3 processes each QoS-filtered IP packet twice: once to apply ingress PFC QoS and once to apply egress PFC QoS.

Caution If the switch is operating in PFC3A mode with egress ACL support for remarked DSCP configured, when the PFC3 processes traffic to apply ingress PFC QoS, it applies ingress PFC QoS filtering and ingress PFC QoS, and incorrectly applies any egress QoS filtering and egress PFC QoS configured on the ingress interface, which results in unexpected behavior if QoS filtering is configured on an interface where egress ACL support for remarked DSCP is enabled. This problem does not occur in other PFC3 modes.

After packets have been processed by ingress PFC QoS and any policing or marking changes have been made, the packets are processed again on the ingress interface by any configured Layer 2 features (for example, VACLs) before being processed by egress PFC QoS.

On an interface where egress ACL support for remarked DSCP is configured, if a Layer 2 feature matches the ingress-QoS-modified IP precedence or DSCP value, the Layer 2 feature might redirect or drop the matched packets, which prevents them from being processed by egress QoS.

After packets have been processed by ingress PFC QoS and any policing or marking changes have been made, the packets are processed on the ingress interface by any configured Layer 3 features (for example, ingress Cisco IOS ACLs, policy-based routing (PBR), etc.) before being processed by egress PFC QoS.

The Layer 3 features configured on an interface where egress ACL support for remarked DSCP is configured might redirect or drop the packets that have been processed by ingress PFC QoS, which would prevent them from being processed by egress PFC QoS.

Understanding Classification and Marking

The following sections describe where and how classification and marking occur in Cisco IOS Release 12.2SX:

- Classification and Marking at Trusted and Untrusted Ingress Ports, page 43-14
- Classification and Marking on the PFC Using Service Policies and Policy Maps, page 43-16
- Classification and Marking on the RP, page 43-17

Classification and Marking at Trusted and Untrusted Ingress Ports

The trust state of an ingress port determines how the port marks, schedules, and classifies received Layer 2 frames, and whether or not congestion avoidance is implemented. These are the port trust states:

- Untrusted (default)
- Trust IP precedence
- Trust DSCP
- Trust CoS

Ingress LAN port classification, marking, and congestion avoidance can use Layer 2 CoS values and do not set Layer 3 IP precedence or DSCP values. You can configure WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports to use received DSCP values for ingress LAN port classification and congestion avoidance (see the "Configuring DSCP-Based Queue Mapping" section on page 43-101). Ingress LAN port classification, marking, and congestion avoidance on other ports use Layer 2 CoS values only.

The following sections describe classification and marking at trusted and untrusted ingress ports:

- Classification and Marking at Untrusted Ingress Ports, page 43-15
- Ingress Classification and Marking at Trusted Ports, page 43-15

Classification and Marking at Untrusted Ingress Ports

PFC QoS Layer 2 remarking marks all frames received through untrusted ports with the port CoS value (the default is zero).

To map the port CoS value that was applied to untrusted ingress traffic to the initial internal DSCP value, configure a trust CoS policy map that matches the ingress traffic.

Ingress Classification and Marking at Trusted Ports

You should configure ports to trust only if they receive traffic that carries valid QoS labels. QoS uses the received QoS labels as the basis of initial internal DSCP value. After the traffic enters the switch, you can apply a different trust state to traffic with a policy map. For example, traffic can enter the switch through a trust CoS port, and then you can use a policy map to trust IP precedence or DSCP, which uses the trusted value as the basis of the initial internal DSCP value, instead of the QoS label that was trusted at the port.

These sections describe classification and marking at trusted ingress ports:

- Ingress Classification and Marking at Trust CoS LAN Ports, page 43-15
- Ingress Classification and Marking at Trust IP Precedence Ports, page 43-15
- Ingress Classification and Marking at Trust DSCP Ports, page 43-15

Ingress Classification and Marking at Trust CoS LAN Ports

You should configure LAN ports to trust CoS only if they receive traffic that carries valid Layer 2 CoS.

When an ISL frame enters the switch through a trusted ingress LAN port, PFC QoS accepts the three least significant bits in the User field as a CoS value. When an 802.1Q frame enters the switch through a trusted ingress LAN port, PFC QoS accepts the User Priority bits as a CoS value. PFC QoS Layer 2 remarking marks all traffic received in untagged frames with the ingress port CoS value.

On ports configured to trust CoS, PFC QoS does the following:

- PFC QoS maps the received CoS value in tagged trust CoS traffic to the initial internal DSCP value.
- PFC QoS maps the ingress port CoS value applied to untagged trusted traffic to the initial internal DSCP value.
- PFC QoS enables the CoS-based ingress queues and thresholds to provide congestion avoidance. See the "Understanding Port-Based Queue Types" section on page 43-20 for more information about ingress queues and thresholds.

Ingress Classification and Marking at Trust IP Precedence Ports

You should configure ports to trust IP precedence only if they receive traffic that carries valid Layer 3 IP precedence. For traffic from trust IP precedence ports, PFC QoS maps the received IP precedence value to the initial internal DSCP value. Because the ingress port queues and thresholds use Layer 2 CoS, PFC QoS does not implement ingress port congestion avoidance on ports configured to trust IP precedence. PFC does not mark any traffic on ingress ports configured to trust IP precedence.

Ingress Classification and Marking at Trust DSCP Ports

You should configure ports to trust DSCP only if they receive traffic that carries valid Layer 3 DSCP.

You can enable DSCP-based ingress queues and thresholds on WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports to provide congestion avoidance (see the "Configuring DSCP-Based Queue Mapping" section on page 43-101). The ingress port queues and thresholds on other ports use only Layer 2 CoS.

For traffic from trust DSCP ports, PFC QoS uses the received DSCP value as the initial internal DSCP value. PFC QoS does not mark any traffic on ingress ports configured to trust received DSCP.

Classification and Marking on the PFC Using Service Policies and Policy Maps

PFC QoS supports classification and marking with service policies that attach one policy map to these interface types to apply ingress PFC QoS:

- Each ingress port (except FlexWAN interfaces)
- Each EtherChannel port-channel interface
- Each VLAN interface

Note
- With releases earlier than Release 12.2(33)SXI, VSS mode does not support ingress service policies on Layer 2 ports.
- With Release 12.2(33)SXI and later releases, VSS mode supports ingress service policies on Layer 2 ports.

You can attach one policy map to each Layer 3 interface (except FlexWAN interfaces) to apply egress PFC QoS.

Each policy map can contain multiple policy-map classes. You can configure a separate policy-map class for each type of traffic handled by the interface. There are two ways to configure filtering in policy-map classes:

- Access control lists (ACLs)
- Class-map **match** commands for IP precedence and DSCP values

Policy-map classes specify actions with the following optional commands:

- Policy-map **set** commands—For untrusted traffic or if ignore port trust is enabled, PFC QoS can use configured IP precedence or DSCP values as the final internal DSCP value. The "IP Precedence and DSCP Values" section on page 43-58 shows the bit values for IP precedence and DSCP.
- Policy-map class **trust** commands—PFC QoS applies the policy-map class trust state to matched ingress traffic, which then uses the trusted value as the basis of its initial internal DSCP value, instead of the QoS label that was trusted at the port (if any). In a policy map, you can trust CoS, IP precedence, or DSCP.

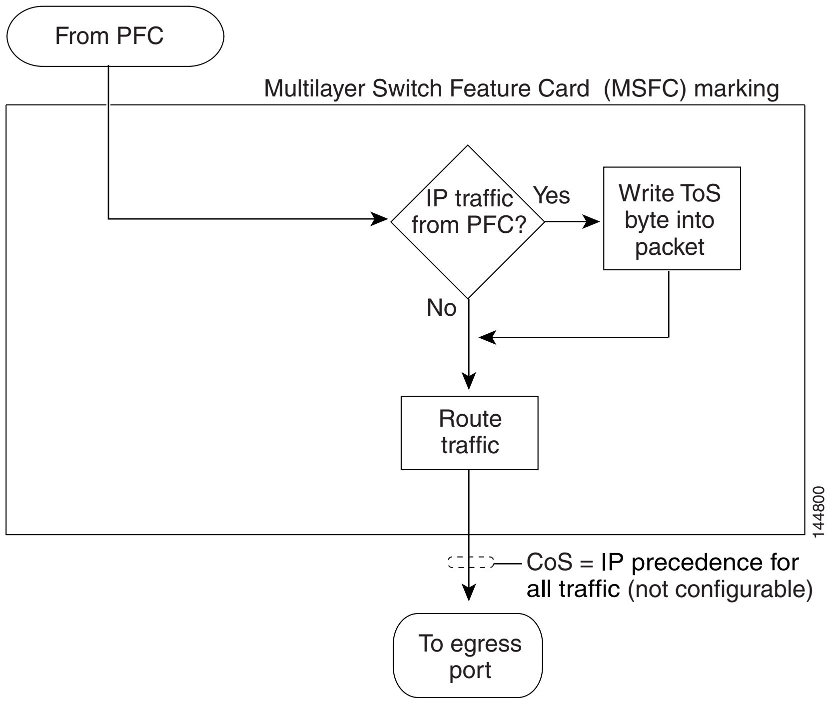
  Note A trust CoS policy map cannot restore received CoS in traffic from untrusted ports. Traffic from untrusted ports always has the port CoS value.

- Aggregate and microflow policers—PFC QoS can use policers to either mark or drop both conforming and nonconforming traffic.

Classification and Marking on the RP

PFC QoS sends IP traffic to the RP with the final internal DSCP values. CoS is equal to IP precedence in all traffic sent from the RP to egress ports.

Figure 43-7 RP Marking



Note Traffic that is Layer 3 switched on the PFC does not go through the RP and retains the CoS value assigned by the PFC.

Policers

These sections describe policers:

- Overview of Policers, page 43-17
- Aggregate Policers, page 43-18
- Microflow Policers, page 43-19

Overview of Policers

Policing allows you to rate limit incoming and outgoing traffic so that it adheres to the traffic forwarding rules defined by the QoS configuration. Sometimes these configured rules for how traffic should be forwarded through the system are referred to as a contract. If the traffic does not adhere to this contract, it is marked down to a lower DSCP value or dropped.

Policing does not buffer out-of-profile packets. As a result, policing does not affect transmission delay. In contrast, traffic shaping works by buffering out-of-profile traffic, which moderates the traffic bursts. (PFC QoS does not support shaping.)

The PFC3 supports ingress and egress PFC QoS, which includes ingress and egress policing. Traffic shaping is supported on some WAN modules.

Note Policers can act on ingress traffic per-port or per-VLAN. For egress traffic, the policers can act per-VLAN only.

You can create policers to do the following:

- Mark traffic
- Limit bandwidth utilization and mark traffic

Aggregate Policers

PFC QoS applies the bandwidth limits specified in an aggregate policer cumulatively to all flows in matched traffic. For example, if you configure an aggregate policer to allow 1 Mbps for all TFTP traffic flows on VLAN 1 and VLAN 3, it limits the TFTP traffic for all flows combined on VLAN 1 and VLAN 3 to 1 Mbps.

- You define per-interface aggregate policers in a policy map class with the **police** command. If you attach a per-interface aggregate policer to multiple ingress ports, it polices the matched traffic on each ingress port separately.
- You create named aggregate policers with the **mls qos aggregate-policer** command. If you attach a named aggregate policer to multiple ingress ports, it polices the matched traffic from all the ingress ports to which it is attached.
- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Each PFC or DFC polices independently, which might affect QoS features being applied to traffic that is distributed across the PFC and any DFCs. Examples of these QoS feature are:
  - Policers applied to a port channel interface.
  - Policers applied to a switched virtual interface.
  - Egress policers applied to either a Layer 3 interface or an SVI. Note that PFC QoS performs egress policing decisions at the ingress interface, on the PFC or ingress DFC.

  Policers affected by this restriction deliver an aggregate rate that is the sum of all the independent policing rates.

Microflow Policers

PFC QoS applies the bandwidth limit specified in a microflow policer separately to each flow in matched traffic. For example, if you configure a microflow policer to limit the TFTP traffic to 1 Mbps on VLAN 1 and VLAN 3, then 1 Mbps is allowed for each flow in VLAN 1 and 1 Mbps for each flow in VLAN 3. In other words, if there are three flows in VLAN 1 and four flows in VLAN 3, the microflow policer allows each of these flows 1 Mbps.

You can configure PFC QoS to apply the bandwidth limits in a microflow policer as follows:

- You can create microflow policers with up to 63 different rate and burst parameter combinations.
- You create microflow policers in a policy map class with the **police flow** command.
- You can configure a microflow policer to use only source addresses, which applies the microflow policer to all traffic from a source address regardless of the destination addresses.
- You can configure a microflow policer to use only destination addresses, which applies the microflow policer to all traffic to a destination address regardless of the source addresses.
- For MAC-Layer microflow policing, PFC QoS considers MAC-Layer traffic with the same protocol and the same source and destination MAC-Layer addresses to be part of the same flow, including traffic with different EtherTypes. You can configure MAC ACLs to filter IPX traffic.

Note With Release 12.2(33)SXI4 and later releases, when appropriate for the configuration of the policer, microflow policers use the interface-full flow mask, which can reduce flowmask conflicts. Releases earlier than Release 12.2(33)SXI4 use the full flow mask.

- By default, microflow policers only affect traffic routed by the RP. To enable microflow policing of other traffic, including traffic in bridge groups, enter the **mls qos bridged** command.
- You cannot apply microflow policing to ARP traffic.
- You cannot apply microflow policing to IPv6 multicast traffic.

You can include both an aggregate policer and a microflow policer in each policy map class to police a flow based on both its own bandwidth utilization and on its bandwidth utilization combined with that of other flows.

Note If traffic is both aggregate and microflow policed, then the aggregate and microflow policers must both be in the same policy-map class and each must use the same **conform-action** and **exceed-action** keyword option: **drop**, **set-dscp-transmit**, **set-prec-transmit**, or **transmit**.

For example, you could create a microflow policer with a bandwidth limit suitable for individuals in a group, and you could create a named aggregate policer with bandwidth limits suitable for the group as a whole. You could include both policers in policy map classes that match the group's traffic. The combination would affect individual flows separately and the group aggregately.

For policy map classes that include both an aggregate and a microflow policer, PFC QoS responds to an out-of-profile status from either policer and, as specified by the policer, applies a new DSCP value or drops the packet. If both policers return an out-of-profile status, then if either policer specifies that the packet is to be dropped, it is dropped; otherwise, PFC QoS applies a marked-down DSCP value.

Note To avoid inconsistent results, ensure that all traffic policed by the same aggregate policer has the same trust state.

Policing uses the Layer 2 frame size. You specify the bandwidth utilization limit as a committed information rate (CIR). You can also specify a higher peak information rate (PIR). Packets that exceed a rate are "out of profile" or "nonconforming."

In each policer, you specify if out-of-profile packets are to be dropped or to have a new DSCP value applied to them (applying a new DSCP value is called "markdown"). Because out-of-profile packets do not retain their original priority, they are not counted as part of the bandwidth consumed by in-profile packets.

If you configure a PIR, the PIR out-of-profile action cannot be less severe than the CIR out-of-profile action. For example, if the CIR out-of-profile action is to mark down the traffic, then the PIR out-of-profile action cannot be to transmit the traffic.

For all policers, PFC QoS uses a configurable global table that maps the internal DSCP value to a marked-down DSCP value. When markdown occurs, PFC QoS gets the marked-down DSCP value from the table. You cannot specify marked-down DSCP values in individual policers.

Note

- Policing with the **conform-action transmit** keywords supersedes the ingress LAN port trust state of matched traffic with trust DSCP or with the trust state defined by a **trust** policy-map class command.
- By default, the markdown table is configured so that no markdown occurs: the marked-down DSCP values are equal to the original DSCP values. To enable markdown, configure the table appropriately for your network.
- When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown.

Understanding Port-Based Queue Types

Port-based queue types are determined by the ASICs that control the ports. The following sections describe the queue types, drop thresholds, and buffers that are supported on the LAN switching modules:

- Ingress and Egress Buffers and Layer 2 CoS-Based Queues, page 43-20
- Ingress Queue Types, page 43-22
- Egress Queue Types, page 43-23
- Module to Queue Type Mappings, page 43-24

Ingress and Egress Buffers and Layer 2 CoS-Based Queues

The Ethernet port ASICs have buffers that are divided into a fixed number of queues. When congestion avoidance is enabled, PFC QoS uses the traffic's Layer 2 CoS value to assign traffic to the queues. The buffers and queues store frames temporarily as they transit the switch. PFC QoS allocates the port ASIC memory as buffers for each queue on each port.

The Ethernet ports support the following types of queues:

- Standard queues
- Strict-priority queues

The Ethernet ports support the following types of scheduling algorithms between queues:

- Shaped round robin (SRR)—SRR allows a queue to use only the allocated bandwidth.
- Deficit weighted round robin (DWRR)—DWRR keeps track of any lower-priority queue under-transmission caused by traffic in a higher-priority queue and compensates in the next round.
- Weighted Round Robin (WRR)—WRR does not explicitly reserve bandwidth for the queues. Instead, the amount of bandwidth assigned to each queue is user configurable. The percentage or weight allocated to a queue defines the amount of bandwidth allocated to the queue.
- Strict-priority queueing—Strict priority queueing allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued, giving delay-sensitive data preferential treatment over other traffic. The switch services traffic in the strict-priority transmit queue before servicing the standard queues. After transmitting a packet from a standard queue, the switch checks for traffic in the strict-priority queue. If the switch detects traffic in the strict-priority queue, it suspends its service of the standard queue and completes service of all traffic in the strict-priority queue before returning to the standard queue.

The Ethernet ports provide congestion avoidance with these types of thresholds within a queue:

- Weighted Random Early Detection (WRED)—On ports with WRED drop thresholds, frames with a given QoS label are admitted to the queue based on a random probability designed to avoid buffer congestion. The probability of a frame with a given QoS label being admitted to the queue or discarded depends on the weight and threshold assigned to that QoS label.

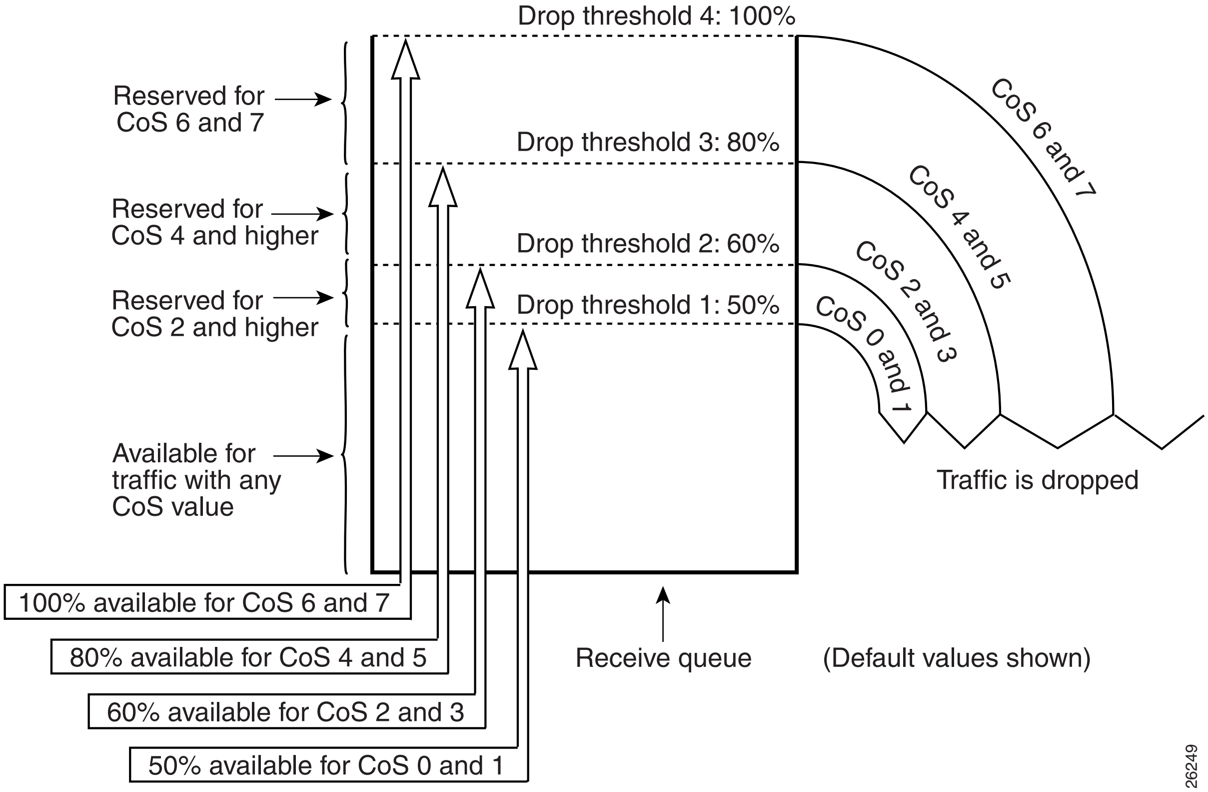
  For example, if CoS 2 is assigned to queue 1, threshold 2, and the threshold 2 levels are 40 percent (low) and 80 percent (high), then frames with CoS 2 will not be dropped until queue 1 is at least 40 percent full. As the queue depth approaches 80 percent, frames with CoS 2 have an increasingly higher probability of being discarded rather than being admitted to the queue. Once the queue is over 80 percent full, all CoS 2 frames are dropped until the queue is less than 80 percent full. The frames the switch discards when the queue level is between the low and high thresholds are picked out at random, rather than on a per-flow basis or in a FIFO manner. This method works well with protocols such as TCP that can adjust to periodic packet drops by backing off and adjusting their transmission window size.
- Tail-drop thresholds—On ports with tail-drop thresholds, frames with a given QoS label are admitted to the queue until the drop threshold associated with that QoS label is exceeded; subsequent frames of that QoS label are discarded until the threshold is no longer exceeded. For example, if CoS 1 is assigned to queue 1, threshold 2, and the threshold 2 watermark is 60 percent, then frames with CoS 1 will not be dropped until queue 1 is 60 percent full. All subsequent CoS 1 frames will be dropped until the queue is less than 60 percent full. With some port types, you can configure the standard receive queue to use both a tail-drop and a WRED-drop threshold by mapping a CoS value to the queue or to the queue and a threshold. The switch uses the tail-drop threshold for traffic carrying CoS values mapped only to the queue. The switch uses WRED-drop thresholds for traffic carrying CoS values mapped to the queue and a threshold. All LAN ports of the same type use the same drop-threshold configuration.

Note You can enable DSCP-based queues and thresholds on WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports (see the "Configuring DSCP-Based Queue Mapping" section on page 43-101).

The combination of multiple queues and the scheduling algorithms associated with each queue allows the switch to provide congestion avoidance.

Figure 43-8 illustrates the drop thresholds for a **1q4t** ingress LAN port. Drop thresholds in other configurations function similarly.

Figure 43-8 Receive Queue Drop Thresholds



Ingress Queue Types

To see the queue structure of a LAN port, enter the **show queueing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command. The command displays one of the following architectures:

- **1q2t** indicates one standard queue with one configurable tail-drop threshold and one nonconfigurable tail-drop threshold.
- **1q4t** indicates one standard queue with four configurable tail-drop thresholds.
- **1q8t** indicates one standard queue with eight configurable tail-drop thresholds.
- **2q8t** indicates two standard queues, each with eight configurable tail-drop thresholds.
- **8q4t** indicates eight standard queues, each with four thresholds, each configurable as either WRED-drop or tail-drop.

Note With releases earlier than Release 12.2(33)SXJ3, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits (CSCts82932).

- **8q8t** indicates eight standard queues, each with eight thresholds, each configurable as either WRED-drop or tail-drop.

- **1p1q4t** indicates:
  - One strict-priority queue
  - One standard queue with four configurable tail-drop thresholds.
- **1p1q0t** indicates:
  - One strict-priority queue
  - One standard queue with no configurable threshold (effectively a tail-drop threshold at 100 percent).
- **1p1q8t** indicates the following:
  - One strict-priority queue
  - One standard queue with these thresholds:

    —Eight thresholds, each configurable as either WRED-drop or tail-drop

    —One nonconfigurable (100 percent) tail-drop threshold

Egress Queue Types

To see the queue structure of an egress LAN port, enter the **show queueing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command.

The command displays one of the following architectures:

- **2q2t** indicates two standard queues, each with two configurable tail-drop thresholds.
- **1p2q2t** indicates the following:
  - One strict-priority queue
  - Two standard queues, each with two configurable WRED-drop thresholds
- **1p3q1t** indicates the following:
  - One strict-priority queue
  - Three standard queues with these thresholds:

    —One threshold configurable as either WRED-drop or tail-drop

    —One nonconfigurable (100 percent) tail-drop threshold
- **1p2q1t** indicates the following:
  - One strict-priority queue
  - Two standard queues with these thresholds:

    —One WRED-drop threshold

    —One nonconfigurable (100 percent) tail-drop threshold
- **1p3q8t** indicates the following:
  - One strict-priority queue
  - Three standard queues, each with eight thresholds, each threshold configurable as either WRED-drop or tail-drop
- **1p7q2t** indicates the following:
  - One strict-priority queue
  - Seven standard queues, each with two thresholds, each threshold configurable as either WRED-drop or tail-drop

- **1p7q4t** indicates the following:
  - One strict-priority queue
  - Seven standard queues, each with four thresholds, each threshold configurable as either WRED-drop or tail-drop
- **1p7q8t** indicates the following:
  - One strict-priority queue
  - Seven standard queues, each with eight thresholds, each threshold configurable as either WRED-drop or tail-drop

Module to Queue Type Mappings

The following tables show the module to queue structure mapping:

- Table 43-2—Supervisor Engine Module QoS Queue Structures
- Table 43-3—10-Gigabit Ethernet Modules
- Table 43-4—Gigabit and 10/100/1000 Ethernet Modules
- Table 43-5—Ethernet and Fast Ethernet Module Queue Structures

Table 43-2 Supervisor Engine Module QoS Queue Structures

Supervisor Engines	Ingress Queue and Drop Thresholds	Ingress Queue Scheduler	Egress Queue and Drop Thresholds	Egress Queue Scheduler	Total Buffer Size	Ingress Buffer Size	Egress Buffer Size
VS-S720-10G-3CXL VS-S720-10G-3C						128 MB	112 MB
With Gigabit Ethernet ports enabled	2q4t	WRR	1p3q4t	DWRR or SRR			
	Does not support DSCP-based queueing.						
With Gigabit Ethernet ports disabled	8q4t	WRR	1p7q4t	DWRR or SRR			
	• Supports DSCP-based queueing. • With releases earlier than Release 12.2(33)SXJ3, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits (CSCts82932).						
WS-SUP720	1p1q4t	—	1p2q2t	WRR	512 KB	73 KB	439 KB
WS-SUP720-3B							
WS-SUP720-3BXL							
WS-SUP32-10GE	2q8t	WRR	1p3q8t	DWRR or SRR			
10-Gigabit Ethernet ports					193 MB	105 MB	88 MB
Gigabit Ethernet port					17.7 MB	9.6 MB	8.1 MB
WS-SUP32-GE					17.7 MB	9.6 MB	8.1 MB

Note To disable the Supervisor Engine 720-10GE Gigabit Ethernet ports, enter **shutdown** interface configuration mode commands for the Supervisor Engine 720-10GE Gigabit Ethernet ports, and then enter the **mls qos 10g-only** global configuration command, which disables the Gigabit Ethernet ports on the Supervisor Engine 720-10GE.

Table 43-3 10-Gigabit Ethernet Modules

Modules	Ingress Queue and Drop Thresholds	Ingress Queue Scheduler	Egress Queue and Drop Thresholds	Egress Queue Scheduler	Total Buffer Size	Ingress Buffer Size	Egress Buffer Size
WS-X6716-10GE, WS-X6716-10T (supports DSCP-based queueing)							
Note With releases earlier than Release 12.2(33)SXJ3, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits (CSCts82932).							
Performance mode	8q4t	DWRR	1p7q4t	DWRR or SRR	198 MB	108 MB per port	90 MB per port
Oversubscription mode	1p7q2t				91 MB	90 MB per port	1 MB per port group
WS-X6708-10GE (supports DSCP-based queueing)					200 MB	108 MB	90 MB
Note With releases earlier than Release 12.2(33)SXJ3, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits (CSCts82932).							
Performance mode	8q4t	DWRR	1p7q4t	DWRR or SRR			
Oversubscription mode	1p7q2t						
WS-X6704-10GE with DFC3	8q8t	WRR	1p7q8t	DWRR	16 MB	2 MB	14 MB
WS-X6704-10GE with CFC	1q8t	—					
WS-X6502-10GE	1p1q8t	—	1p2q1t	DWRR	64.2 MB	256 KB	64 MB

Table 43-4 Gigabit and 10/100/1000 Ethernet Modules

Modules	Ingress Queue and Drop Thresholds	Ingress Queue Scheduler	Egress Queue and Drop Thresholds	Egress Queue Scheduler	Total Buffer Size	Ingress Buffer Size	Egress Buffer Size
WS-X6816-GBIC	1p1q4t	—	1p2q2t	WRR	512 KB	73 KB	439 KB
WS-X6748-GE-TX with DFC3	2q8t	WRR	1p3q8t	DWRR	1.3 MB	166 KB	1.2 MB
WS-X6748-GE-TX with CFC	1q8t	—					
WS-X6748-SFP with DFC3	2q8t	WRR					
WS-X6748-SFP with CFC	1q8t	—					
WS-X6724-SFP with DFC3	2q8t	WRR					
WS-X6724-SFP with CFC	1q8t	—					

Table 43-4 Gigabit and 10/100/1000 Ethernet Modules (continued)

Modules	Ingress Queue and Drop Thresholds	Ingress Queue Scheduler	Egress Queue and Drop Thresholds	Egress Queue Scheduler	Total Buffer Size	Ingress Buffer Size	Egress Buffer Size
WS-X6548-GE-TX	1q2t	—	1p2q2t	WRR	1.4 MB	185 KB	1.2 MB
WS-X6548V-GE-TX							
WS-X6548-GE-45AF							
WS-X6516-GBIC	1p1q4t	—	1p2q2t	WRR	512 KB	73 KB	439 KB
WS-X6516A-GBIC				WRR	1 MB	135 KB	946 KB
WS-X6516-GE-TX				WRR	512 KB	73 KB	439 KB
WS-X6408-GBIC	1q4t	—	2q2t	WRR		80 KB	432 KB
WS-X6408A-GBIC	1p1q4t	—	1p2q2t	WRR		73 KB	439 KB
WS-X6416-GBIC							
WS-X6416-GE-MT							
WS-X6316-GE-TX							
WS-X6148-GE-TX	1q2t	—			1.4 MB	185 KB	1.2 MB
WS-X6148V-GE-TX							
WS-X6148-GE-45AF							

Table 43-5 Ethernet and Fast Ethernet Module Queue Structures

Modules	Ingress Queue and Drop Thresholds	Ingress Queue Scheduler	Egress Queue and Drop Thresholds	Egress Queue Scheduler	Total Buffer Size	Ingress Buffer Size	Egress Buffer Size
WS-X6524-100FX-MM	1p1q0t	—	1p3q1t	DWRR	1,116 KB	28 KB	1,088 KB
WS-X6548-RJ-21							
WS-X6548-RJ-45							

Table 43-5 Ethernet and Fast Ethernet Module Queue Structures (continued)

Modules	Ingress Queue and Drop Thresholds	Ingress Queue Scheduler	Egress Queue and Drop Thresholds	Egress Queue Scheduler	Total Buffer Size	Ingress Buffer Size	Egress Buffer Size
WS-X6324-100FX-MM	1q4t	—	2q2t	WRR	128 KB	16 KB	112 KB
WS-X6324-100FX-SM							
WS-X6348-RJ-45							
WS-X6348-RJ-45V							
WS-X6348-RJ-21V							
WS-X6224-100FX-MT					64 KB	8 KB	56 KB
WS-X6248-RJ-45							
WS-X6248-TEL							
WS-X6248A-TEL					128 KB	16 KB	112 KB
WS-X6148-RJ-45							
WS-X6148-RJ-45V							
WS-X6148-45AF							
WS-X6148-RJ-21							
WS-X6148-RJ-21V							
WS-X6148-21AF							
WS-X6148X2-RJ-45	1p1q0t	—	1p3q1t	DWRR	1,116 KB	28 KB	1,088 KB
WS-X6148X2-45AF							
WS-X6024-10FL-MT	1q4t	—	2q2t	WRR	64 KB	8 KB	56 KB

PFC QoS Default Configuration

These sections describe the PFC QoS default configuration:

- PFC QoS Global Settings, page 43-27
- Default Values with PFC QoS Enabled, page 43-28
- Default Values with PFC QoS Disabled, page 43-52

PFC QoS Global Settings

The following global PFC QoS settings apply:

Feature	Default Value
PFC QoS global enable state	Disabled
PFC QoS port enable state	Enabled when PFC QoS is globally enabled
Port CoS value	0

Feature	Default Value
Microflow policing	Enabled
IntraVLAN microflow policing	Disabled
Port-based or VLAN-based PFC QoS	Port-based
Received CoS to initial internal DSCP map (initial internal DSCP set from received CoS values)	CoS 0 = DSCP 0 CoS 1 = DSCP 8 CoS 2 = DSCP 16 CoS 3 = DSCP 24 CoS 4 = DSCP 32 CoS 5 = DSCP 40 CoS 6 = DSCP 48 CoS 7 = DSCP 56
Received IP precedence to initial internal DSCP map (initial internal DSCP set from received IP precedence values)	IP precedence 0 = DSCP 0 IP precedence 1 = DSCP 8 IP precedence 2 = DSCP 16 IP precedence 3 = DSCP 24 IP precedence 4 = DSCP 32 IP precedence 5 = DSCP 40 IP precedence 6 = DSCP 48 IP precedence 7 = DSCP 56
Final internal DSCP to egress CoS map (egress CoS set from final internal DSCP values)	DSCP 0–7 = CoS 0 DSCP 8–15 = CoS 1 DSCP 16–23 = CoS 2 DSCP 24–31 = CoS 3 DSCP 32–39 = CoS 4 DSCP 40–47 = CoS 5 DSCP 48–55 = CoS 6 DSCP 56–63 = CoS 7
Marked-down DSCP from DSCP map	Marked-down DSCP value equals original DSCP value (no markdown)
Policers	None
Policy maps	None
Protocol-independent MAC ACL filtering	Disabled
VLAN-based MAC ACL QoS filtering	Disabled

Default Values with PFC QoS Enabled

These sections list the default values that apply when PFC QoS is enabled:

- Receive-Queue Limits, page 43-29
- Transmit-Queue Limits, page 43-29
- Bandwidth Allocation Ratios, page 43-30
- Default Drop-Threshold Percentages and CoS Value Mappings, page 43-30

Note The ingress LAN port trust state defaults to untrusted with QoS enabled.

Receive-Queue Limits

Feature	Default Value
2q8t	Low priority: 80%
	High priority: 20%
8q4t	Low priority: 80%
	Intermediate queues: 0%
	High priority: 20%
8q8t	Lowest priority: 80%
	Intermediate queues: 0%
	Highest priority: 20%

Transmit-Queue Limits

Feature	Default Value
2q2t	Low priority: 80%
	High priority: 20%
1p2q2t	Low priority: 70%
	High priority: 15%
	Strict priority 15%
1p2q1t	Low priority: 70%
	High priority: 15%
	Strict priority 15%
1p3q8t	Low priority: 50%
	Medium priority: 20%
	High priority: 15%
	Strict priority 15%
1p7q2t	Standard queue 1 (lowest priority): 50%
	Standard queue 2: 20%
	Standard queue 3: 15%
	Standard queues 4 through 7: 0%
	Strict priority 15%
1p7q4t	Standard queue 1 (lowest priority): 50%
	Standard queue 2: 20%
	Standard queue 3: 15%
	Standard queues 4 through 7: 0%
	Strict priority 15%

Feature	Default Value
1p7q8t	Standard queue 1 (lowest priority): 50%
	Standard queue 2: 20%
	Standard queue 3: 15%
	Standard queues 4 through 7: 0%
	Strict priority 15%

Bandwidth Allocation Ratios

Feature	Default Value
2q8t	90:10
8q4t	90:0:0:0:0:0:0:10
8q8t	90:0:0:0:0:0:0:10
1p3q8t	100:150:200
1p7q2t	5:255
1p7q4t	100:150:200:0:0:0:0:0
1p7q8t	100:150:200:0:0:0:0
1p2q1t	100:255
2q2t, **1p2q2t**, and **1p2q1t**	5:255
1p3q1t	100:150:255

Default Drop-Threshold Percentages and CoS Value Mappings

The following tables list the default drop-thresholds values and CoS mappings for different queue types:

- 1q2t Receive Queues, page 43-31
- 1q4t Receive Queues, page 43-31
- 1p1q4t Receive Queues, page 43-32
- 1p1q0t Receive Queues, page 43-32
- 1p1q8t Receive Queues, page 43-33
- 1q8t Receive Queues, page 43-34
- 2q8t Receive Queues, page 43-35
- 8q4t Receive Queues, page 43-36
- 8q8t Receive Queues, page 43-42
- 2q2t Transmit Queues, page 43-42
- 1p2q2t Transmit Queues, page 43-43
- 1p3q8t Transmit Queues, page 43-44
- 1p7q2t Receive Queues, page 43-45
- 1p7q4t Transmit Queues, page 43-47

Note The receive queue values shown are the values in effect when the port is configured to trust CoS or DSCP. When the port is untrusted, the receive queue values are the same as when QoS is globally disabled.

1q2t Receive Queues

Feature			Default Value
Standard receive queue	Threshold 1	CoS	0, 1, 2, 3, and 4
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 2	CoS	5, 6, and 7
		Tail-drop	100% (not configurable)
		WRED-drop	Not supported

1q4t Receive Queues

Feature			Default Value
Standard receive queue	Threshold 1	CoS	0 and 1
		Tail-drop	50%
		WRED-drop	Not supported
	Threshold 2	CoS	2 and 3
		Tail-drop	60%
		WRED-drop	Not supported
	Threshold 3	CoS	4 and 5
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 4	CoS	6 and 7
		Tail-drop	100%
		WRED-drop	Not supported

1p1q4t Receive Queues

Feature			Default Value
Standard receive queue	Threshold 1	CoS	0 and 1
		Tail-drop	50%
		WRED-drop	Not supported
	Threshold 2	CoS	2 and 3
		Tail-drop	60%
		WRED-drop	Not supported
	Threshold 3	CoS	4 and 6
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 4	CoS	7
		Tail-drop	100%
		WRED-drop	Not supported
Strict-priority receive queue		CoS	5
		Tail-drop	100% (nonconfigurable)

1p1q0t Receive Queues

Feature		Default Value
Standard receive queue	CoS	0, 1, 2, 3, 4, 6, and 7
	Tail-drop	100% (nonconfigurable)
	WRED-drop	Not supported
Strict-priority receive queue	CoS	5
	Tail-drop	100% (nonconfigurable)

1p1q8t Receive Queues

Feature			Default Value
Standard receive queue	Threshold 1	CoS	0
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	1
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 3	CoS	2
		Tail-drop	Disabled; 80%
		WRED-drop	Enabled; 50% low, 80% high
	Threshold 4	CoS	3
		Tail-drop	Disabled; 80%
		WRED-drop	Enabled; 50% low, 80% high
	Threshold 5	CoS	4
		Tail-drop	Disabled; 90%
		WRED-drop	Enabled; 60% low, 90% high
	Threshold 6	CoS	6
		Tail-drop	Disabled; 90%
		WRED-drop	Enabled; 60% low, 90% high
	Threshold 7	CoS	7
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled;70% low, 100% high
Strict-priority receive queue		CoS	5
		Tail-drop	100% (nonconfigurable)

1q8t Receive Queues

Feature			Default Value
Standard receive queue	Threshold 1	CoS	0
		Tail-drop	50%
		WRED-drop	Not supported
	Threshold 2	CoS	None
		Tail-drop	50%
		WRED-drop	Not supported
	Threshold 3	CoS	1, 2, 3, 4
		Tail-drop	60%
		WRED-drop	Not supported
	Threshold 4	CoS	None
		Tail-drop	60%
		WRED-drop	Not supported
	Threshold 5	CoS	6 and 7
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 6	CoS	None
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 7	CoS	5
		Tail-drop	100%
		WRED-drop	Not supported
	Threshold 8	CoS	None
		Tail-drop	100%
		WRED-drop	Not supported

2q8t Receive Queues

Feature			Default Value
Standard receive queue 1 (low priority)	Threshold 1	CoS	0 and 1
		Tail-drop	70%
		WRED-drop	Not supported
	Threshold 2	CoS	2 and 3
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 3	CoS	4
		Tail-drop	90%
		WRED-drop	Not supported
	Threshold 4	CoS	6 and 7
		Tail-drop	100%
		WRED-drop	Not supported
	Thresholds 5–8	CoS	None
		Tail-drop	100%
		WRED-drop	Not supported
Standard receive queue 2 (high priority)	Threshold 1	CoS	5
		Tail-drop	100%
		WRED-drop	Not supported
	Thresholds 2–8	CoS	None
		Tail-drop	100%
		WRED-drop	Not supported

8q4t Receive Queues

> **Note** With releases earlier than Release 12.2(33)SXJ3, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits (CSCts82932).

Feature			Default Value
Standard receive queue 1 (lowest priority)	Threshold 1	CoS	0 and 1
		DSCP	Release 12.2(33)SXJ3 and later releases: 0–9, 11, 13–17, 19, 21–25, 27, 29–39, 48–63 Releases earlier than Release 12.2(33)SXJ3: 0–9, 11, 13, 15–17, 19, 21, 23, 25, 27, 29, 31, 33, 39, 41–45, 47
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	2 and 3
		DSCP	Release 12.2(33)SXJ3 and later releases: 12, 20, 28 Releases earlier than Release 12.2(33)SXJ3: None.
		Tail-drop	Disabled; 80%
		WRED-drop	Enabled; 40% low, 80% high
	Threshold 3	CoS	4
		DSCP	Release 12.2(33)SXJ3 and later releases: 10, 18, 26 Releases earlier than Release 12.2(33)SXJ3: None.
		Tail-drop	Disabled; 90%
		WRED-drop	Enabled; 50% low, 90% high
	Threshold 4	CoS	6 and 7
		DSCP	None.
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 50% low, 100% high

Feature (continued)			Default Value
Standard receive queue 2 (intermediate priority)	Threshold 1	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 14
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 12
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 10
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

Feature (continued)			Default Value
Standard receive queue 3 (intermediate priority)	Threshold 1	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 22
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 20
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 18
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

Feature (continued)			Default Value
Standard receive queue 4 (intermediate priority)	Threshold 1	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 24 and 30
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 28
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 26
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

Feature (continued)			Default Value
Standard receive queue 5 (intermediate priority)	Threshold 1	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 32, 34–38
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Standard receive queue 6 (intermediate priority)	Threshold 1	CoS	None
		DSCP	Release 12.2(33)SXJ3 and later releases: None. Releases earlier than Release 12.2(33)SXJ3: 48–63
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

Feature (continued)			Default Value
Standard receive queue 7 (intermediate priority)	Threshold 1	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Standard receive queue 8 (high priority)	Threshold 1	CoS	5
		DSCP	Release 12.2(33)SXJ3 and later releases: 40–47 Releases earlier than Release 12.2(33)SXJ3: 40 and 46
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

8q8t Receive Queues

Feature			Default Value
Standard receive queue 1 (lowest priority)	Threshold 1	CoS	0 and 1
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	2 and 3
		Tail-drop	Disabled; 80%
		WRED-drop	Enabled; 40% low, 80% high
	Threshold 3	CoS	4
		Tail-drop	Disabled; 90%
		WRED-drop	Enabled; 50% low, 90% high
	Threshold 4	CoS	6 and 7
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 50% low, 100% high
	Thresholds 5–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 50% low, 100% high
Standard receive queues 2–7 (intermediate priorities)	Thresholds 1–8	CoS	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Standard receive queue 8 (highest priority)	Threshold 1	CoS	5
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Thresholds 2–8	CoS	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

2q2t Transmit Queues

Feature			Default Value
Standard transmit queue 1 (low priority)	Threshold 1	CoS	0 and 1
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 2	CoS	2 and 3
		Tail-drop	100%
		WRED-drop	Not supported

Feature			Default Value
Standard transmit queue 2 (high priority)	Threshold 1	CoS	4 and 5
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 2	CoS	6 and 7
		Tail-drop	100%
		WRED-drop	Not supported

1p2q2t Transmit Queues

Feature			Default Value
Standard transmit queue 1 (low priority)	Threshold 1	CoS	0 and 1
		Tail-drop	Not supported
		WRED-drop	40% low, 70% high
	Threshold 2	CoS	2 and 3
		Tail-drop	Not supported
		WRED-drop	70% low, 100% high
Standard transmit queue 2 (high priority)	Threshold 1	CoS	4 and 6
		Tail-drop	Not supported
		WRED-drop	40% low, 70% high
	Threshold 2	CoS	7
		Tail-drop	Not supported
		WRED-drop	70% low, 100% high
Strict-priority transmit queue		CoS	5
		Tail-drop	100% (nonconfigurable)

1p3q8t Transmit Queues

Feature			Default Value
Standard transmit queue 1 (lowest priority)	Threshold 1	CoS	0
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	1
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 3	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 4	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Thresholds 5–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 50% low, 100% high
Standard transmit queue 2 (medium priority)	Threshold 1	CoS	2
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	3 and 4
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Thresholds 3–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard transmit queue 3 (high priority)	Threshold 1	CoS	6 and 7
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Thresholds 2–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Strict-priority transmit queue		CoS	5
		Tail-drop	100% (nonconfigurable)

1p7q2t Receive Queues

Feature			Default Value
Standard receive queue 1 (lowest priority)	Threshold 1	CoS	0
		DSCP	0–9, 11, 13, 15–17, 19, 21, 23, 25, 27, 29, 31, 33, 39, 41–45, 47
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	1
		DSCP	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard receive queue 2 (intermediate priority)	Threshold 1	CoS	2
		DSCP	14
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	3 and 4
		DSCP	10 and 12
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard receive queue 3 (intermediate priority)	Threshold 1	CoS	6 and 7
		DSCP	22
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 2	CoS	None
		DSCP	18 and 20
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard receive queue 4 (intermediate priority)	Threshold 1	CoS	None
		DSCP	24 and 30
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	26 and 28
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

Feature (continued)			Default Value
Standard receive queue 5 (intermediate priority)	Threshold 1	CoS	None
		DSCP	32, 34–38
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Standard receive queue 6 (intermediate priority)	Threshold 1	CoS	None
		DSCP	48–63
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Standard receive queue 7 (intermediate priority)	Threshold 1	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Strict-priority transmit queue		CoS	5
		DSCP	40 and 46
		Tail-drop	100% (nonconfigurable)

1p7q4t Transmit Queues

Feature			Default Value
Standard transmit queue 1 (lowest priority)	Threshold 1	CoS	0 and 1
		DSCP	0–9, 11, 13, 15–17, 19, 21, 23, 25, 27, 29, 31, 33, 39, 41–45, 47
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	2 and 3
		DSCP	
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 3	CoS	4
		DSCP	
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 4	CoS	6 and 7
		DSCP	
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard transmit queue 2 (intermediate priority)	Threshold 1	CoS	None
		DSCP	14
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	None
		DSCP	12
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 3	CoS	None
		DSCP	10
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high

Feature (continued)			Default Value
Standard transmit queue 3 (intermediate priority)	Threshold 1	CoS	None
		DSCP	22
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 2	CoS	None
		DSCP	20
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 3	CoS	None
		DSCP	18
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard transmit queue 4 (intermediate priority)	Threshold 1	CoS	None
		DSCP	24 and 30
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	28
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	26
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

Feature (continued)			Default Value
Standard transmit queue 5 (intermediate priority)	Threshold 1	CoS	None
		DSCP	32, 34–38
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Standard transmit queue 6 (intermediate priority)	Threshold 1	CoS	None
		DSCP	48–63
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high

Feature (continued)			Default Value
Standard transmit queue 7 (intermediate priority)	Threshold 1	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 2	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 3	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 4	CoS	None
		DSCP	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Strict-priority transmit queue		CoS	5
		DSCP	40 and 46
		Tail-drop	100% (nonconfigurable)

1p7q8t Transmit Queues

Feature			Default Value
Standard transmit queue 1 (lowest priority)	Threshold 1	CoS	0
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	1
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Thresholds 3–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high

Feature (continued)			Default Value
Standard transmit queue 2 (intermediate priority)	Threshold 1	CoS	2
		Tail-drop	Disabled; 70%
		WRED-drop	Enabled; 40% low, 70% high
	Threshold 2	CoS	3 and 4
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Thresholds 3–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard transmit queue 3 (intermediate priority)	Threshold 1	CoS	6 and 7
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
	Thresholds 2–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 100% low, 100% high
Standard transmit queues 4–7 (intermediate priorities)	Thresholds 1–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 100% low, 100% high
Strict-priority transmit queue		CoS	5
		Tail-drop	100% (nonconfigurable)

1p3q1t Transmit Queues

Feature			Default Value
Standard transmit queue 1 (lowest priority)	Threshold 1	CoS	0 and 1
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard transmit queue 2 (medium priority)	Threshold 1	CoS	2, 3, and 4
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Standard transmit queue 3 (high priority)	Threshold 1	CoS	6 and 7
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 70% low, 100% high
Strict-priority transmit queue		CoS	5
		Tail-drop	100% (nonconfigurable)

1p2q1t Transmit Queues

Feature			Default Value
Standard transmit queue 1 (lowest priority)	Threshold 1	CoS	0, 1, 2, and 3
		Tail-drop	Not supported
		WRED-drop	Enabled; 70% low, 100% high
Standard transmit queue 3 (high priority)	Threshold 1	CoS	4, 6, and 7
		Tail-drop	Not supported
		WRED-drop	Enabled; 70% low, 100% high
Strict-priority transmit queue		CoS	5
		Tail-drop	100% (nonconfigurable)

Default Values with PFC QoS Disabled

Feature	Default Value
Ingress LAN port trust state	Trust DSCP.
Receive-queue drop-threshold percentages	All thresholds set to 100%.
Transmit-queue drop-threshold percentages	All thresholds set to 100%.
Transmit-queue bandwidth allocation ratio	255:1.
Transmit-queue size ratio	Low priority: 100% (other queues not used).
CoS value and drop threshold mapping	All QoS labels mapped to the low-priority queue.

PFC QoS Configuration Guidelines and Restrictions

When configuring PFC QoS, follow these guidelines and restrictions:

- General Guidelines, page 43-53
- Class Map Command Restrictions, page 43-56
- Policy Map Command Restrictions, page 43-56
- Policy Map Class Command Restrictions, page 43-56
- Supported Granularity for CIR and PIR Rate Values, page 43-56
- Supported Granularity for CIR and PIR Token Bucket Sizes, page 43-57
- IP Precedence and DSCP Values, page 43-58

General Guidelines

- PFC QoS cannot be applied to IGMP, MLD, or PIM traffic.
- Configure the same trust mode on all ports supported by an ASIC. Mismatched trust modes cause inconsistent bandwidth and queue-limit ratios.
- With a Supervisor Engine 720-10GE that is using 10G mode, only one port per-ASIC is available because the 1-Gigabit uplinks are shutdown, making the behavior similar to that of WS-X6708-10GE.
- When you configure a port on a Supervisor Engine 720-10GE as a member of the VSL, the **mls qos trust cos** command is automatically added to the port configuration.
- The **match ip precedence** and **match ip dscp** commands filter only IPv4 traffic.
- The **match precedence** and **match dscp** commands filter IPv4 and IPv6 traffic.
- The **set ip dscp** and **set ip precedence** commands are saved in the configuration file as **set dscp** and **set precedence** commands.
- PFC QoS supports the **set dscp** and **set precedence** policy map class commands for IPv4 and IPv6 traffic.
- The flowmask requirements of QoS, NetFlow, and NetFlow data export (NDE) might conflict, especially if you configure microflow policing.
- With egress ACL support for remarked DSCP and VACL capture both configured on an interface, VACL capture might capture two copies of each packet, and the second copy might be corrupt.
- You cannot configure egress ACL support for remarked DSCP on tunnel interfaces.
- Egress ACL support for remarked DSCP supports IP unicast traffic.
- Egress ACL support for remarked DSCP is not relevant to multicast traffic. PFC QoS applies ingress QoS changes to multicast traffic before applying egress QoS.
- NetFlow and NetFlow data export (NDE) do not support interfaces where egress ACL support for remarked DSCP is configured.
- When egress ACL support for remarked DSCP is configured on any interface, you must configure an interface-specific flowmask to enable NetFlow and NDE support on interfaces where egress ACL support for remarked DSCP is not configured. Enter either the **mls flow ip interface-destination-source** or the **mls flow ip interface-full** global configuration mode command.
- Interface counters are not accurate on interfaces where egress ACL support for remarked DSCP is configured.
- You cannot apply microflow policing to IPv6 multicast traffic.
- You cannot apply microflow policing to traffic that has been permitted by egress ACL support for remarked DSCP.
- Traffic that has been permitted by egress ACL support for remarked DSCP cannot be tagged as MPLS traffic. (The traffic can be tagged as MPLS traffic on another network device.)
- When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown. (CSCea23571)
- If traffic is both aggregate and microflow policed, then the aggregate and microflow policers must both be in the same policy-map class and each must use the same **conform-action** and **exceed-action** keyword option: **drop**, **set-dscp-transmit**, **set-prec-transmit**, or **transmit**.

- You cannot configure PFC QoS features on tunnel interfaces.
- PFC QoS does not rewrite the payload ToS byte in tunnel traffic.
- PFC QoS filters only by ACLs, dscp values, or IP precedence values.
- For these commands, PFC QoS applies identical configuration to all LAN ports controlled by the same application-specific integrated circuit (ASIC):
  - **rcv-queue cos-map**
  - **wrr-queue cos-map**
- Except for WS-X6716-10T, WS-X6716-10GE, WS-X6708-10GE, WS-X6704-10GE, WS-X6748-SFP, WS-X6724-SFP, WS-X6748-GE-TX modules, PFC QoS applies identical configuration to all LAN ports controlled by the same application-specific integrated circuit (ASIC) for these commands:
  - **rcv-queue random-detect**
  - **rcv-queue queue-limit**
  - **wrr-queue queue-limit**
  - **wrr-queue bandwidth**
  - **priority-queue cos-map**
  - **wrr-queue threshold**
  - **rcv-queue threshold**
  - **wrr-queue random-detect**
  - **wrr-queue random-detect min-threshold**
  - **wrr-queue random-detect max-threshold**
- Configure these commands only on physical ports. Do not configure these commands on logical interfaces:
  - **priority-queue cos-map**
  - **wrr-queue cos-map**
  - **wrr-queue random-detect**
  - **wrr-queue random-detect max-threshold**
  - **wrr-queue random-detect min-threshold**
  - **wrr-queue threshold**
  - **wrr-queue queue-limit**
  - **wrr-queue bandwidth**
  - **rcv-queue cos-map**
  - **rcv-queue bandwidth**
  - **rcv-queue random-detect**
  - **rcv-queue random-detect max-threshold**
  - **rcv-queue random-detect min-threshold**
  - **rcv-queue queue-limit**
  - **rcv-queue cos-map**
  - **rcv-queue threshold**

Note IP multicast switching using egress packet replication is not compatible with QoS. In some cases, egress replication can result in the incorrect COS or DSCP marking of packets. If you are using QoS and your switching modules are capable of egress replication, enter the **mls ip multicast replication-mode ingress** command to force ingress replication.

- All versions of the PFC3 support QoS for IPv6 unicast and multicast traffic.
- To display information about IPv6 PFC QoS, enter the **show mls qos ipv6** command.
- The QoS features implemented in the port ASICs (queue architecture and dequeuing algorithms) support IPv4 and IPv6 traffic.
- The PFC3 supports IPv6 named extended ACLs and named standard ACLs.
- The PFC3 supports the **match protocol ipv6** command.
- Because of conflicting TCAM lookup flow key bit requirements, you cannot configure IPv6 DSCP-based filtering and IPv6 Layer 4 range-based filtering on the same interface. For example:
  - If you configure both a DSCP value and a Layer 4 "greater than" (gt) or "less than" (lt) operator in an IPv6 ACE, you cannot use the ACL for PFC QoS filtering.
  - If you configure a DSCP value in one IPv6 ACL and a Layer 4 "greater than" (gt) or "less than" (lt) operator in another IPv6 ACL, you cannot use both ACLs in different class maps on the same interface for PFC QoS filtering.
- You can apply aggregate and microflow policers to IPv6 traffic, but you cannot apply microflow policing to IPv6 multicast traffic.
- With egress ACL support for remarked DSCP configured, the PFC3 does not provide hardware-assistance for these features:
  - Cisco IOS reflexive ACLs
  - TCP intercept
  - Network Address Translation (NAT)
- You cannot apply microflow policing to ARP traffic.
- The PFC3 does not apply egress policing to traffic that is being bridged to the RP.
- The PFC3 does not apply egress policing or egress DSCP mutation to multicast traffic from the RP.
- With a PFC3A, PFC QoS does not rewrite the ToS byte in bridged multicast traffic.
- The PFC3 supports up to 1023 configurable aggregate policers, but some PFC QoS commands other than the **police** command will be included in this count. By default, any policy using a **set** or **trust** command will be included in the aggregate policer count. You can disable the addition of the **set** or **trust** commands to the aggregate policer count by entering the **no mls qos marking statistics** command, but you will then be unable to collect statistics for the classmaps associated with these commands. You can view the aggregate policer count in the QoS Policer Resources section of the output of the **show platform hardware capacity qos** command.

Class Map Command Restrictions

- PFC QoS supports a single **match** command in **class-map match-all** class maps, except that the **match protocol** command can be configured in a class map with the **match dscp** or **match precedence** command.
- PFC QoS supports multiple **match** commands in **class-map match-any** class maps.
- PFC QoS does not support these class map commands:
  - **match cos**
  - **match classmap**
  - **match destination-address**
  - **match input-interface**
  - **match qos-group**
  - **match source-address**

Policy Map Command Restrictions

PFC QoS does not support these policy map commands:

- **class** *class_name* **destination-address**
- **class** *class_name* **input-interface**
- **class** *class_name* **protocol**
- **class** *class_name* **qos-group**
- **class** *class_name* **source-address**

Policy Map Class Command Restrictions

PFC QoS does not support these policy map class commands:

- **bandwidth**
- **priority**
- **queue-limit**
- **random-detect**
- **set qos-group**
- **service-policy**

Supported Granularity for CIR and PIR Rate Values

PFC QoS has the following hardware granularity for CIR and PIR rate values:

CIR and PIR Rate Value Range	Granularity
32768 to 2097152 (2 Mbs)	32768 (32 Kb)
2097153 to 4194304 (4 Mbs)	65536 (64 Kb)
4194305 to 8388608 (8 Mbs)	131072 (128 Kb)
8388609 to 16777216 (16 Mbs)	262144 (256 Kb)
16777217 to 33554432 (32 Mbs)	524288 (512 Kb)
33554433 to 67108864 (64 Mbs)	1048576 (1 Mb)
67108865 to 134217728 (128 Mbs)	2097152 (2 Mb)
134217729 to 268435456 (256 Mbs)	4194304 (4 Mb)
268435457 to 536870912 (512 Mbs)	8388608 (8 Mb)
536870913 to 1073741824 (1 Gbs)	16777216 (16 Mb)
1073741825 to 2147483648 (2 Gbs)	33554432 (32 Mb)
2147483649 to 4294967296 (4 Gbs)	67108864 (64 Mb)
4294967296 to 8589934592 (8 Gbs)	134217728 (128 Mb)
8589934593 to 17179869184 (16 Gbs)	268435456 (256 Mb)
17179869185 to 34359738368 (32 Gbs)	536870912 (512 Mb)
34359738369 to 68719476736 (64 Gbs)	1073741824 (1024 Mb)

Within each range, PFC QoS programs the PFC with rate values that are multiples of the granularity values.

Supported Granularity for CIR and PIR Token Bucket Sizes

PFC QoS has the following hardware granularity for CIR and PIR token bucket (burst) sizes:

CIR and PIR Token Bucket Size Range	Granularity
1 to 32768 (32 KB)	1024 (1 KB)
32769 to 65536 (64 KB)	2048 (2 KB)
65537 to 131072 (128 KB)	4096 (4 KB)
131073 to 262144 (256 KB)	8196 (8 KB)
262145 to 524288 (512 KB)	16392 (16 KB)
524289 to 1048576 (1 MB)	32768 (32 KB)
1048577 to 2097152 (2 MB)	65536 (64 KB)
2097153 to 4194304 (4 MB)	131072 (128 KB)
4194305 to 8388608 (8 MB)	262144 (256 KB)
8388609 to 16777216 (16 MB)	524288 (512 KB)
16777217 to 33554432 (32 MB)	1048576 (1 MB)

Within each range, PFC QoS programs the PFC with token bucket sizes that are multiples of the granularity values.

IP Precedence and DSCP Values

3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP
	8	7	6	5	4	3	
0	0	0	0	0	0	0	0
	0	0	0	0	0	1	1
	0	0	0	0	1	0	2
	0	0	0	0	1	1	3
	0	0	0	1	0	0	4
	0	0	0	1	0	1	5
	0	0	0	1	1	0	6
	0	0	0	1	1	1	7
1	0	0	1	0	0	0	8
	0	0	1	0	0	1	9
	0	0	1	0	1	0	10
	0	0	1	0	1	1	11
	0	0	1	1	0	0	12
	0	0	1	1	0	1	13
	0	0	1	1	1	0	14
	0	0	1	1	1	1	15
2	0	1	0	0	0	0	16
	0	1	0	0	0	1	17
	0	1	0	0	1	0	18
	0	1	0	0	1	1	19
	0	1	0	1	0	0	20
	0	1	0	1	0	1	21
	0	1	0	1	1	0	22
	0	1	0	1	1	1	23
3	0	1	1	0	0	0	24
	0	1	1	0	0	1	25
	0	1	1	0	1	0	26
	0	1	1	0	1	1	27
	0	1	1	1	0	0	28
	0	1	1	1	0	1	29
	0	1	1	1	1	0	30
	0	1	1	1	1	1	31

3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP
	8	7	6	5	4	3	
4	1	0	0	0	0	0	32
	1	0	0	0	0	1	33
	1	0	0	0	1	0	34
	1	0	0	0	1	1	35
	1	0	0	1	0	0	36
	1	0	0	1	0	1	37
	1	0	0	1	1	0	38
	1	0	0	1	1	1	39
5	1	0	1	0	0	0	40
	1	0	1	0	0	1	41
	1	0	1	0	1	0	42
	1	0	1	0	1	1	43
	1	0	1	1	0	0	44
	1	0	1	1	0	1	45
	1	0	1	1	1	0	46
	1	0	1	1	1	1	47
6	1	1	0	0	0	0	48
	1	1	0	0	0	1	49
	1	1	0	0	1	0	50
	1	1	0	0	1	1	51
	1	1	0	1	0	0	52
	1	1	0	1	0	1	53
	1	1	0	1	1	0	54
	1	1	0	1	1	1	55
7	1	1	1	0	0	0	56
	1	1	1	0	0	1	57
	1	1	1	0	1	0	58
	1	1	1	0	1	1	59
	1	1	1	1	0	0	60
	1	1	1	1	0	1	61
	1	1	1	1	1	0	62
	1	1	1	1	1	1	63

1. MSb = most significant bit

Configuring PFC QoS

These sections describe how to configure PFC QoS in Cisco IOS Release 12.2SX:

- Enabling PFC QoS Globally, page 43-59
- Enabling Ignore Port Trust, page 43-60

Note PFC QoS processes both unicast and multicast traffic.

Enabling PFC QoS Globally

To enable PFC QoS globally, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`mls qos`**	Enables PFC QoS globally on the switch.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.
Step 3	`Router#` **`show mls qos`** `[`**`ipv6`**`]`	Verifies the configuration.

This example shows how to enable PFC QoS globally:

```
Router# configure terminal
Router(config)# mls qos
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos
  QoS is enabled globally
  Microflow QoS is enabled globally

QoS global counters:
```

```
    Total packets: 544393
    IP shortcut packets: 1410
    Packets dropped by policing: 0
    IP packets with TOS changed by policing: 467
    IP packets with COS changed by policing: 59998
    Non-IP packets with COS changed by policing: 0

Router#
```

Enabling Ignore Port Trust

The ignore port trust feature allows an ingress policy to apply a configured IP precedence or DSCP value to any traffic, rather than only to untrusted traffic.

To enable ignore port trust, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos marking ignore port-trust**	Enables ignore port trust globally on the switch.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show mls qos | include ignores**	Verifies the configuration.

Note For untrusted traffic, when ignore port trust is enabled, PFC QoS does the following:

- For IP traffic, PFC QoS uses the received DSCP value as the initial internal DSCP value.
- For traffic without a recognizable ToS byte, PFC QoS maps the port CoS value to the initial internal DSCP value.

This example shows how to enable ignore port trust and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos marking ignore port-trust
Router(config)# end
Router# show mls qos | include ignores
  Policy marking ignores port_trust
Router#
```

Configuring DSCP Transparency

Note

- In addition to support for other IP traffic, PFC3C and PFC3CXL mode support the **no mls qos rewrite ip dscp** command for these traffic types:
  - MPLS traffic
  - Traffic in IP in IP tunnels
  - Traffic in GRE tunnels
- In addition to support for other IP traffic, PFC3B and PFC3BXL mode support the **no mls qos rewrite ip dscp** command for these traffic types:
  - Except on PE routers, MPLS traffic
  - Traffic in IP in IP tunnels
  - Traffic in GRE tunnels
- PFC3A mode does not support the **no mls qos rewrite ip dscp** command for MPLS traffic, traffic in IP in IP tunnels, and traffic in GRE tunnels.

To enable DSCP transparency, which preserves the received Layer 3 ToS byte, perform this task:

	Command	Purpose
Step 1	`Router(config)# no mls qos rewrite ip dscp`	Disables egress ToS byte rewrite globally on the switch.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos	include rewrite`	Verifies the configuration.

When you preserve the received Layer 3 ToS byte, QoS uses the marked or marked-down CoS value for egress queueing and in egress tagged traffic.

This example shows how to preserve the received Layer 3 ToS byte and verify the configuration:

```
Router# configure terminal
Router(config)# no mls qos rewrite ip dscp
Router(config)# end
Router# show mls qos | include rewrite
  QoS ip packet dscp rewrite disabled globally
Router#
```

Enabling Queueing-Only Mode

To enable queueing-only mode on the switch, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos queueing-only`	Enables queueing-only mode on the switch.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos`	Verifies the configuration.

When you enable queueing-only mode, the switch does the following:

- Disables marking and policing globally
- Configures all ports to trust Layer 2 CoS

Note The switch applies the port CoS value to untagged ingress traffic and to traffic that is received through ports that cannot be configured to trust CoS.

This example shows how to enable queueing-only mode:

```
Router# configure terminal
Router(config)# mls qos queueing-only
Router(config)# end
Router#
```

Enabling Microflow Policing of Bridged Traffic

By default, microflow policers affect only routed traffic. To enable microflow policing of bridged traffic on specified VLANs, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {{**vlan** *vlan_ID*} | {*type*[1] *slot/port*}}	Selects the interface to configure.
Step 2	Router(config-if)# **mls qos bridged**	Enables microflow policing of bridged traffic, including bridge groups, on the VLAN.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show mls qos**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable microflow policing of bridged traffic on VLANs 3 through 5:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface range vlan 3 - 5
Router(config-if)# mls qos bridged
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos | begin Bridged QoS
Bridged QoS is enabled on the following interfaces:
    Vl3 Vl4 Vl5
<...output truncated...>
Router#
```

Enabling VLAN-Based PFC QoS on Layer 2 LAN Ports

Note You can attach policy maps to Layer 3 interfaces for application of PFC QoS to egress traffic. VLAN-based or port-based PFC QoS on Layer 2 ports is not relevant to application of PFC QoS to egress traffic on Layer 3 interfaces.

By default, PFC QoS uses policy maps attached to LAN ports. For ports configured as Layer 2 LAN ports with the **switchport** keyword, you can configure PFC QoS to use policy maps attached to a VLAN. Ports not configured with the **switchport** keyword are not associated with a VLAN.

To enable VLAN-based PFC QoS on a Layer 2 LAN port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`*`type`*[1] *`slot/port`*`}	{`**`port-channel`** *`number`*`}}`	Selects the interface to configure.
Step 2	`Router(config-if)#` **`mls qos vlan-based`**	Enables VLAN-based PFC QoS on a Layer 2 LAN port or a Layer 2 EtherChannel.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show mls qos`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable VLAN-based PFC QoS on Fast Ethernet port 5/42:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/42
Router(config-if)# mls qos vlan-based
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show mls qos | begin QoS is vlan-based
  QoS is vlan-based on the following interfaces:
    Fa5/42
<...Output Truncated...>
```

Note Configuring a Layer 2 LAN port for VLAN-based PFC QoS preserves the policy map port configuration. The **no mls qos vlan-based** port command reenables any previously configured port commands.

Enabling Egress ACL Support for Remarked DSCP

To enable egress ACL support for remarked DSCP on an ingress interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`vlan`** *`vlan_ID`*`}	{`*`type`*[1] *`slot/port`*`}	{`**`port-channel`** *`number`*`}}`	Selects the ingress interface to configure.
Step 2	`Router(config-if)#` **`platform ip features sequential`** `[`**`access-group`** *`IP_acl_name_or_number`*`]`	Enables egress ACL support for remarked DSCP on the ingress interface.

	Command	Purpose
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show running-config interface** ({*type*[1] *slot/port*} | {**port-channel** *number*})	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring egress ACL support for remarked DSCP on an ingress interface, note the following information:

- To enable egress ACL support for remarked DSCP only for the traffic filtered by a specific standard, extended named, or extended numbered IP ACL, enter the IP ACL name or number.
- If you do not enter an IP ACL name or number, egress ACL support for remarked DSCP is enabled for all IP ingress IP traffic on the interface.

This example shows how to enable egress ACL support for remarked DSCP on Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# platform ip features sequential
Router(config-if)# end
```

Creating Named Aggregate Policers

To create a named aggregate policer, perform this task:

Command	Purpose
Router(config)# **mls qos aggregate-policer** *policer_name bits_per_second normal_burst_bytes* [*maximum_burst_bytes*] [**pir** *peak_rate_bps*] [[[**conform-action** {**drop** | **set-dscp-transmit**[1] *dscp_value* | **set-prec-transmit**[1] *ip_precedence_value* | **transmit**}] **exceed-action** {**drop** | **policed-dscp** | **transmit**}] **violate-action** {**drop** | **policed-dscp** | **transmit**}]	Creates a named aggregate policer.

1. The **set-dscp-transmit** and **set-prec-transmit** keywords are only supported for IP traffic.

When creating a named aggregate policer, note the following information:

- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Each PFC or DFC polices independently, which might affect QoS features being applied to traffic that is distributed across the PFC and any DFCs. Examples of these QoS feature are:
  - Policers applied to a port channel interface.
  - Policers applied to a switched virtual interface.
  - Egress policers applied to either a Layer 3 interface or an SVI. Note that PFC QoS performs egress policing decisions at the ingress interface, on the PFC or ingress DFC.

Policers affected by this restriction deliver an aggregate rate that is the sum of all the independent policing rates.

- You can apply aggregate policers to IPv6 traffic.
- Policing uses the Layer 2 frame size.
- See the "PFC QoS Configuration Guidelines and Restrictions" section on page 43-52 for information about rate and burst size granularity.
- The valid range of values for the CIR *bits_per_second* parameter is as follows:
  - Minimum—32 kilobits per second, entered as 32000
  - Maximum with Release 12.2(33)SXH1 and earlier—10 gigabits per second, entered as 10000000000
  - Maximum with Release 12.2(33)SXH2 and later—64 gigabits per second, entered as 64000000000
- The *normal_burst_bytes* parameter sets the CIR token bucket size.
- The *maximum_burst_bytes* parameter sets the PIR token bucket size.
- When configuring the size of a token bucket, note the following information:
  - Because the token bucket must be large enough to hold at least one frame, configure the token bucket size to be larger than the maximum size of the traffic being policed.
  - For TCP traffic, configure the token bucket size as a multiple of the TCP window size, with a minimum value at least twice as large as the maximum size of the traffic being policed.
  - The *maximum_burst_bytes* parameter must be set larger than the *normal_burst_bytes* parameter
  - To sustain a specific rate, set the token bucket size to be at least the rate value divided by 2000.

Note To provide compatibility with QoS as implemented in the Catalyst operating system, Release 12.2(33)SXI and later releases support 1 byte as the minimum token bucket size, entered as 1.

  - With Release 12.2(33)SXI and later releases, the minimum token bucket size is 1 byte, entered as 1.
  - With releases earlier than Release 12.2(33)SXI, the minimum token bucket size is 1000 bytes, entered as 1000.
  - The maximum token bucket size is 512 megabytes, entered as 512000000.
- The valid range of values for the **pir** *bits_per_second* parameter is as follows:
  - Minimum—32 kilobits per second, entered as 32000 (the value cannot be smaller than the CIR *bits_per_second* parameters)
  - Maximum with Release 12.2(33)SXH1 and earlier—10 gigabits per second, entered as 10000000000
  - Maximum with Release 12.2(33)SXH2 and later—64 gigabits per second, entered as 64000000000
- (Optional) You can specify a conform action for matched in-profile traffic as follows:
  - The default conform action is **transmit**, which sets the policy map class trust state to *trust DSCP* unless the policy map class contains a **trust** command.

- To set PFC QoS labels in untrusted traffic, enter the **set-dscp-transmit** keyword to mark matched untrusted traffic with a new DSCP value or enter the **set-prec-transmit** keyword to mark matched untrusted traffic with a new IP precedence value. The **set-dscp-transmit** and **set-prec-transmit** keywords are only supported for IP traffic. PFC QoS sets egress ToS and CoS from the configured value.
- Enter the **drop** keyword to drop all matched traffic.

Note When you configure **drop** as the conform action, PFC QoS configures **drop** as the exceed action and the violate action.

- (Optional) For traffic that exceeds the CIR, you can specify an exceed action as follows:
  - The default exceed action is **drop**, except with a *maximum_burst_bytes* parameter (**drop** is not supported with a *maximum_burst_bytes* parameter).

Note When the exceed action is **drop**, PFC QoS ignores any configured violate action.

  - Enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map.

Note When you create a policer that does not use the **pir** keyword and the *maximum_burst_bytes* parameter is equal to the *normal_burst_bytes* parameter (which is the case if you do not enter the *maximum_burst_bytes* parameter), the **exceed-action policed-dscp-transmit** keywords cause PFC QoS to mark traffic down as defined by the **policed-dscp max-burst** markdown map.

- (Optional) For traffic that exceeds the PIR, you can specify a violate action as follows:
  - To mark traffic without policing, enter the **transmit** keyword to transmit all matched out-of-profile traffic.
  - The default violate action is equal to the exceed action.
  - Enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map.
  - For marking without policing, enter the **transmit** keyword to transmit all matched out-of-profile traffic.

Note When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown.

This example shows how to create a named aggregate policer with a 1-Mbps rate limit and a 10-MB burst size that transmits conforming traffic and marks down out-of-profile traffic:

```
Router(config)# mls qos aggregate-policer aggr-1 1000000 10000000 conform-action transmit
exceed-action policed-dscp-transmit
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos aggregate-policer aggr-1
ag1 1000000 1000000 conform-action transmit exceed-action policed-dscp-transmit AgId=0
[pol4]
Router#
```

The output displays the following:

- The **AgId** parameter displays the hardware policer ID.
- The policy maps that use the policer are listed in the square brackets ([]).

Configuring a PFC QoS Policy

These sections describe PFC QoS policy configuration:

- PFC QoS Policy Configuration Overview, page 43-67
- Configuring MAC ACLs, page 43-68
- Configuring ARP ACLs for QoS Filtering, page 43-72
- Configuring a Class Map, page 43-72
- Verifying Class Map Configuration, page 43-75
- Configuring a Policy Map, page 43-75
- Verifying Policy Map Configuration, page 43-81
- Attaching a Policy Map to an Interface, page 43-82
- Configuring Dynamic Per-Session Attachment of a Policy Map, page 43-83

Note PFC QoS policies process both unicast and multicast traffic.

PFC QoS Policy Configuration Overview

Note To mark traffic without limiting bandwidth utilization, create a policer that uses the **transmit** keywords for both conforming and nonconforming traffic.

These commands configure traffic classes and the policies to be applied to those traffic classes and attach the policies to ports:

- **access-list** (Optional for IP traffic. You can filter IP traffic with **class-map** commands.):
  - PFC QoS supports these ACL types:

Protocol	Numbered ACLs	Extended ACLs	Named ACLs
IPv4	Yes: 1 to 99 1300 to 1999	Yes: 100 to 199 2000 to 2699	Yes
IPv6	—	Yes (named)	Yes

Protocol	Numbered ACLs	Extended ACLs	Named ACLs
MAC Layer	No	No	Yes
ARP	No	No	Yes

  – The PFC3 supports IPv6 named extended ACLs and named standard ACLs.
  – The PFC3 supports ARP ACLs.

Note —The PFC3 does not apply IP ACLs to ARP traffic.

—You cannot apply microflow policing to ARP traffic.

  – The PFC3 does not support IPX ACLs. With the PFC3, you can configure MAC ACLs to filter IPX traffic.
  – PFC QoS supports time-based Cisco IOS ACLs.
  – Except for MAC ACLs and ARP ACLs, see the *Cisco IOS Security Configuration Guide*, Release 12.2, "Traffic Filtering and Firewalls," at this URL:

    http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfacls.html
  – See Chapter 47, "Configuring Network Security," for additional information about ACLs in Cisco IOS Release 12.2SX.
- **class-map** (optional)—Enter the **class-map** command to define one or more traffic classes by specifying the criteria by which traffic is classified.
- **policy-map**—Enter the **policy-map** command to define the following:
  – Policy map class trust mode
  – Aggregate policing and marking
  – Microflow policing and marking
- **service-policy**—Enter the **service-policy** command to attach a policy map to an interface.

Configuring MAC ACLs

These sections describe MAC ACL configuration:

Note You can use MAC ACLs with VLAN ACLs (VACLs). For more information, see Chapter 51, "Configuring Port ACLs and VLAN ACLs."

Configuring Protocol-Independent MAC ACL Filtering

Note PFC3A mode does not support protocol-independent MAC ACL filtering.

Protocol-independent MAC ACL filtering applies MAC ACLs to all ingress traffic types (for example, IPv4 traffic, IPv6 traffic, and MPLS traffic, in addition to MAC-layer traffic).

You can configure these interface types for protocol-independent MAC ACL filtering:

- VLAN interfaces without IP addresses
- Physical LAN ports configured to support EoMPLS
- Logical LAN subinterfaces configured to support EoMPLS

Ingress traffic permitted or denied by a MAC ACL on an interface configured for protocol-independent MAC ACL filtering is processed by egress interfaces as MAC-layer traffic. You cannot apply egress IP ACLs to traffic that was permitted or denied by a MAC ACL on an interface configured for protocol-independent MAC ACL filtering.

To configure protocol-independent MAC ACL filtering, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`vlan`** *`vlan_ID`*`} | {`*`type`*[1] *`slot/port`*`[.`*`subinterface`*`]} | {`**`port-channel`** *`number`*`[.`*`subinterface`*`]}}`	Selects the interface to configure.
Step 2	`Router(config-if)#` **`mac packet-classify`**	Enables protocol-independent MAC ACL filtering on the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring protocol-independent MAC ACL filtering, note the following information:

- Do not configure protocol-independent MAC ACL filtering on VLAN interfaces where you have configured an IP address.
- Do not configure protocol-independent MAC ACL filtering with microflow policing when the permitted traffic would be bridged or Layer 3 switched in hardware by the PFC.
- Protocol-independent MAC ACL filtering supports microflow policing when the permitted traffic is routed in software by the RP.

This example shows how to configure VLAN interface 4018 for protocol-independent MAC ACL filtering and how to verify the configuration:

```
Router(config)# interface vlan 4018
Router(config-if)# mac packet-classify
Router(config-if)# end
Router# show running-config interface vlan 4018 | begin 4018
interface Vlan4018
mtu 9216
ipv6 enable
mac packet-classify
end
```

This example shows how to configure Gigabit Ethernet interface 6/1 for protocol-independent MAC ACL filtering and how to verify the configuration:

```
Router(config)# interface gigabitethernet 6/1
Router(config-if)# mac packet-classify
Router(config-if)# end
Router# show running-config interface gigabitethernet 6/1 | begin 6/1
interface GigabitEthernet6/1
mtu 9216
no ip address
mac packet-classify
```

```
mpls l2transport route 4.4.4.4 4094
end
```

This example shows how to configure Gigabit Ethernet interface 3/24, subinterface 4000, for protocol-independent MAC ACL filtering and how to verify the configuration:

```
Router(config)# interface gigabitethernet 3/24.4000
Router(config-if)# mac packet-classify
Router(config-if)# end
Router# show running-config interface gigabitethernet 3/24.4000 | begin 3/24.4000
interface GigabitEthernet3/24.4000
encapsulation dot1Q 4000
mac packet-classify
mpls l2transport route 4.4.4.4 4000
end
```

Enabling VLAN-Based MAC QoS Filtering

You can globally enable or disable VLAN-based QoS filtering in MAC ACLs. VLAN-based QoS filtering in MAC ACLs is disabled by default.

To enable VLAN-based QoS filtering in MAC ACLs, perform this task:

Command	Purpose
Router(config)# **mac packet-classify use vlan**	Enables VLAN-based QoS filtering in MAC ACLs.

To disable VLAN-based QoS filtering in MAC ACLs, perform this task:

Command	Purpose
Router(config)# **no mac packet-classify use vlan**	Disables VLAN-based QoS filtering in MAC ACLs.

Configuring MAC ACLs

You can configure named ACLs that filter IPX, DECnet, AppleTalk, VINES, or XNS traffic based on MAC addresses.

You can configure MAC ACLs that do VLAN-based filtering or CoS-based filtering or both.

You can globally enable or disable VLAN-based QoS filtering in MAC ACLs (disabled by default).

To configure a MAC ACL, perform this task:

	Command	Purpose
Step 1	Router(config)# **mac host** *name mac_addr*	(Optional) Assigns a name to a MAC address.
Step 2	Router(config)# **mac access-list extended** *list_name*	Configures a MAC ACL.
Step 3	Router(config-ext-macl)# {**permit** | **deny**} {*src_mac_mask* | {**host name** *src_mac_name*} | **any**} {*dest_mac_mask* | {**host name** *dst_mac_name*} | **any**} [{*protocol_keyword* | {*ethertype_number ethertype_mask*}} [**vlan** *vlan_ID*] [**cos** *cos_value*]]	Configures an access control entry (ACE) in a MAC ACL. The source and destination MAC addresses can be specified by MAC address masks or by names created with the **mac host** command.

When configuring an entry in a MAC-Layer ACL, note the following information:

- The PFC3 supports the **ipx-arpa** and **ipx-non-arpa** keywords.
- Cisco IOS Release 12.2SX supports the **vlan** and **cos** keywords.
- The **vlan** and **cos** keywords are not supported in MAC ACLs used for VACL filtering.
- The **vlan** keyword for VLAN-based QoS filtering in MAC ACLs can be globally enabled or disabled and is disabled by default.
- You can enter MAC addresses as three 2-byte values in dotted hexadecimal format. For example, 0030.9629.9f84.
- You can enter MAC address masks as three 2-byte values in dotted hexadecimal format. Use 1 bits as wildcards. For example, to match an address exactly, use 0000.0000.0000 (can be entered as 0.0.0).
- You can enter an EtherType and an EtherType mask as hexadecimal values.
- Entries without a protocol parameter match any protocol.
- ACL entries are scanned in the order you enter them. The first matching entry is used. To improve performance, place the most commonly used entries near the beginning of the ACL.
- An implicit **deny any any** entry exists at the end of an ACL unless you include an explicit **permit any any** entry at the end of the list.
- All new entries to an existing list are placed at the end of the list. You cannot add entries to the middle of a list.
- This list shows the EtherType values and their corresponding protocol keywords:
  - 0x0600—xns-idp—Xerox XNS IDP
  - 0x0BAD—vines-ip—Banyan VINES IP
  - 0x0baf—vines-echo—Banyan VINES Echo
  - 0x6000—etype-6000—DEC unassigned, experimental
  - 0x6001—mop-dump—DEC Maintenance Operation Protocol (MOP) Dump/Load Assistance
  - 0x6002—mop-console—DEC MOP Remote Console
  - 0x6003—decnet-iv—DEC DECnet Phase IV Route
  - 0x6004—lat—DEC Local Area Transport (LAT)
  - 0x6005—diagnostic—DEC DECnet Diagnostics
  - 0x6007—lavc-sca—DEC Local-Area VAX Cluster (LAVC), SCA
  - 0x6008—amber—DEC AMBER
  - 0x6009—mumps—DEC MUMPS
  - 0x0800—ip—Malformed, invalid, or deliberately corrupt IP frames
  - 0x8038—dec-spanning—DEC LANBridge Management
  - 0x8039—dsm—DEC DSM/DDP
  - 0x8040—netbios—DEC PATHWORKS DECnet NETBIOS Emulation
  - 0x8041—msdos—DEC Local Area System Transport
  - 0x8042—etype-8042—DEC unassigned
  - 0x809B—appletalk—Kinetics EtherTalk (AppleTalk over Ethernet)
  - 0x80F3—aarp—Kinetics AppleTalk Address Resolution Protocol (AARP)

This example shows how to create a MAC-Layer ACL named mac_layer that denies dec-phase-iv traffic with source address 0000.4700.0001 and destination address 0000.4700.0009, but permits all other traffic:

```
Router(config)# mac access-list extended mac_layer
Router(config-ext-macl)# deny 0000.4700.0001 0.0.0 0000.4700.0009 0.0.0 dec-phase-iv
Router(config-ext-macl)# permit any any
```

Configuring ARP ACLs for QoS Filtering

Note
- The PFC3 does not apply IP ACLs to ARP traffic.
- You cannot apply microflow policing to ARP traffic.

You can configure named ACLs that filter ARP traffic (EtherType 0x0806) for QoS.

To configure an ARP ACL for QoS filtering, perform this task:

	Command	Purpose
Step 1	`Router(config)# arp access-list list_name`	Configures an ARP ACL for QoS filtering.
Step 2	`Router(config-arp-nacl)# {permit | deny} {ip {any | host sender_ip | sender_ip sender_ip_wildcardmask} mac any`	Configures an access control entry (ACE) in an ARP ACL for QoS filtering.

When configuring an entry in an ARP ACL for QoS filtering, note the following information:

- This publication describes the ARP ACL syntax that is supported in hardware by the PFC3. Any other ARP ACL syntax displayed by the CLI help when you enter a question mark ("?") is not supported and cannot be used to filter ARP traffic for QoS.
- ACLs entries are scanned in the order you enter them. The first matching entry is used. To improve performance, place the most commonly used entries near the beginning of the ACL.
- An implicit **deny ip any mac any** entry exists at the end of an ACL unless you include an explicit **permit ip any mac any** entry at the end of the list.
- All new entries to an existing list are placed at the end of the list. You cannot add entries to the middle of a list.

This example shows how to create an ARP ACL named arp_filtering that only permits ARP traffic from IP address 1.1.1.1:

```
Router(config)# arp access-list arp_filtering
Router(config-arp-nacl)# permit ip host 1.1.1.1 mac any
```

Configuring a Class Map

These sections describe class map configuration:

- Creating a Class Map, page 43-73
- Class Map Filtering Guidelines and Restrictions, page 43-73
- Configuring Filtering in a Class Map, page 43-74

Creating a Class Map

To create a class map, perform this task:

Command	Purpose
`Router(config)# class-map [match-all | match-any] class_name`	Creates a class map. **Note** If you do not enter a **match** keyword, the default is **match-all**.

Class Map Filtering Guidelines and Restrictions

When configuring class map filtering, follow these guidelines and restrictions:

- PFC QoS supports a single **match** command in **class-map match-all** class maps, except that the **match protocol** command can be configured in a class map with the **match dscp** or **match precedence** command.
- PFC QoS supports multiple **match** commands in **class-map match-any** class maps.
- When multiple **match access-group** ACLs are included in a **match-any** class map, and one ACL contains a **deny ip any any** entry, all match criteria after the **deny ip any any** entry (either in the same ACL or in different ACLs) will not be installed in the TCAM.

  In the following example, ACLs acl4 and acl5 will not be installed because they follow acl3, which contains a **deny ip any any** entry:

  ```
  ip access-list ext acl3
    deny ip any any

  class-map cmap1
    match access-group acl1
    match access-group acl2
    match access-group acl3
    match access-group acl4
    match access-group acl5
  ```

  You can use either of the following workarounds to avoid this issue:

  – Move the **deny ip any any** entry to the end of the ACL and move that ACL to the end of the class map.

  – Configure all ACLs that must follow the **deny ip any any** entry into different class maps.
- The PFC3 supports the **match protocol ipv6** command.
- Because of conflicting TCAM lookup flow key bit requirements, you cannot configure IPv6 DSCP-based filtering and IPv6 Layer 4 range-based filtering on the same interface. For example:

  – If configure both a DSCP value and a Layer 4 greater than (gt) or less than (lt) operator in an IPv6 ACE, you cannot use the ACL for PFC QoS filtering.

  – If configure a DSCP value in one IPv6 ACL and a Layer 4 greater than (gt) or less than (lt) operator in another IPv6 ACL, you cannot use both ACLs in different class maps on the same interface for PFC QoS filtering.
- The IPv6 address matching against Layer 4 ports is ignored if the IPv6 address in the ACE is not compressible. The IPv6 source and destination addresses are matched, but the configured source or destination UDP or TCP ports will be ignored. To force Layer 4 port matching, use the **mls ipv6 acl compress address unicast** command.

- PFC QoS supports the **match protocol ip** command for IPv4 traffic.
- PFC QoS does not support the **match cos**, **match classmap**, **match destination-address**, **match input-interface**, **match qos-group**, and **match source-address** class map commands.
- Cisco IOS Release 12.2SX does not detect the use of unsupported commands until you attach a policy map to an interface.
- Filtering based on IP precedence or DSCP for egress QoS uses the received IP precedence or DSCP. Egress QoS filtering is not based on any IP precedence or DSCP changes made by ingress QoS.

Note This chapter includes the following ACL documentation:

- Configuring MAC ACLs, page 43-68
- Configuring ARP ACLs for QoS Filtering, page 43-72

Other ACLs are not documented in this publication. See the references under **access-list** in the "PFC QoS Policy Configuration Overview" section on page 43-67.

Configuring Filtering in a Class Map

To configure filtering in a class map, perform one of these tasks:

Command	Purpose
`Router(config-cmap)# **match access-group name** *acl_index_or_name*`	(Optional) Configures the class map to filter using an ACL.
`Router (config-cmap)# **match protocol ipv6**`	(Optional—for IPv6 traffic) Configures the class map to filter IPv6 traffic.
`Router (config-cmap)# **match precedence** *ipp_value1* [*ipp_value2* [*ipp_valueN*]]`	(Optional—for IPv4 or IPv6 traffic) Configures the class map to filter based on up to eight IP precedence values. **Note** Does not support source-based or destination-based microflow policing.
`Router (config-cmap)# **match dscp** *dscp_value1* [*dscp_value2* [*dscp_valueN*]]`	(Optional—for IPv4 or IPv6 traffic only) Configures the class map to filter based on up to eight DSCP values. **Note** Does not support source-based or destination-based microflow policing.
`Router (config-cmap)# **match ip precedence** *ipp_value1* [*ipp_value2* [*ipp_valueN*]]`	(Optional—for IPv4 traffic) Configures the class map to filter based on up to eight IP precedence values. **Note** Does not support source-based or destination-based microflow policing.
`Router (config-cmap)# **match ip dscp** *dscp_value1* [*dscp_value2* [*dscp_valueN*]]`	(Optional—for IPv4 traffic) Configures the class map to filter based on up to eight DSCP values. **Note** Does not support source-based or destination-based microflow policing.

Verifying Class Map Configuration

To verify class map configuration, perform this task:

	Command	Purpose
Step 1	`Router (config-cmap)# end`	Exits configuration mode.
Step 2	`Router# show class-map class_name`	Verifies the configuration.

This example shows how to create a class map named **ipp5** and how to configure filtering to match traffic with IP precedence 5:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# class-map ipp5
Router(config-cmap)# match ip precedence 5
Router(config-cmap)# end
```

This example shows how to verify the configuration:

```
Router# show class-map ipp5
 Class Map match-all ipp5 (id 1)
   Match ip precedence 5
```

Configuring a Policy Map

You can attach only one policy map to an interface. Policy maps can contain one or more policy map classes, each with different policy map commands.

Configure a separate policy map class in the policy map for each type of traffic that an interface receives. Put all commands for each type of traffic in the same policy map class. PFC QoS does not attempt to apply commands from more than one policy map class to matched traffic.

These sections describe policy map configuration:

- Creating a Policy Map, page 43-75
- Policy Map Class Configuration Guidelines and Restrictions, page 43-75
- Creating a Policy Map Class and Configuring Filtering, page 43-76
- Configuring Policy Map Class Actions, page 43-76

Creating a Policy Map

To create a policy map, perform this task:

Command	Purpose
`Router(config)# policy-map policy_name`	Creates a policy map.

Policy Map Class Configuration Guidelines and Restrictions

When you configuring policy map classes, follow the guidelines and restrictions:

- PFC QoS does not support the **class** *class_name* **destination-address**, **class** *class_name* **input-interface**, **class** *class_name* **qos-group**, and **class** *class_name* **source-address** policy map commands.

- PFC QoS supports the **class default** policy map command.
- PFC QoS does not detect the use of unsupported commands until you attach a policy map to an interface.

Creating a Policy Map Class and Configuring Filtering

To create a policy map class and configure it to filter with a class map, perform this task:

Command	Purpose
`Router(config-pmap)# class class_name`	Creates a policy map class and configures it to filter with a class map. **Note** • PFC QoS supports a single **match** command in class maps configured with the **match-all** keyword. • PFC QoS supports multiple **match** commands in class maps configured with the **match-any** keyword.

Configuring Policy Map Class Actions

When configuring policy map class actions, note the following information:

- Policy maps can contain one or more policy map classes.
- Put all trust-state and policing commands for each type of traffic in the same policy map class.
- PFC QoS only applies commands from one policy map class to traffic. After traffic has matched the filtering in one policy map class, QoS does apply the filtering configured in other policy map classes.
- For hardware-switched traffic, PFC QoS does not support the **bandwidth**, **priority**, **queue-limit**, or **random-detect** policy map class commands. You can configure these commands because they can be used for software-switched traffic.
- PFC QoS does not support the **set qos-group** policy map class commands.
- PFC QoS supports the **set ip dscp** and **set ip precedence** policy map class commands for IPv4 traffic.
  - You can use the **set ip dscp** and **set ip precedence** commands on non-IP traffic to mark the internal DSCP value, which is the basis of the egress Layer 2 CoS value.
  - The **set ip dscp** and **set ip precedence** commands are saved in the configuration file as **set dscp** and **set precedence** commands.
- PFC QoS supports the **set dscp** and **set precedence** policy map class commands for IPv4 and IPv6 traffic.
- You cannot do all three of the following in a policy map class:
  - Mark traffic with the **set** commands
  - Configure the trust state
  - Configure policing

  In a policy map class, you can either mark traffic with the **set** commands or do one or both of the following:
  - Configure the trust state

- Configure policing

Note When configure policing, you can mark traffic with policing keywords.

These sections describe policy map class action configuration:

- Configuring Policy Map Class Marking, page 43-77
- Configuring the Policy Map Class Trust State, page 43-77
- Configuring Policy Map Class Policing, page 43-77

Configuring Policy Map Class Marking

When the ignore port trust feature is enabled, PFC QoS supports policy map class marking for all traffic with **set** policy map class commands.

In all releases, PFC QoS supports policy map class marking for untrusted traffic with **set** policy map class commands.

To configure policy map class marking, perform this task:

Command	Purpose
`Router(config-pmap-c)#` **`set`** `{`**`dscp`** *`dscp_value`* `|` **`precedence`** *`ip_precedence_value`*`}`	Configures the policy map class to mark matched untrusted traffic with the configured DSCP or IP precedence value.

Configuring the Policy Map Class Trust State

Note You cannot attach a policy map that configures a trust state with the **service-policy output** command.

To configure the policy map class trust state, perform this task:

Command	Purpose
`Router(config-pmap-c)#` **`trust`** `{`**`cos`** `|` **`dscp`** `|` **`ip-precedence`**`}`	Configures the policy map class trust state, which selects the value that PFC QoS uses as the source of the initial internal DSCP value.

When configuring the policy map class trust state, note the following information:

- Enter the **no trust** command to use the trust state configured on the ingress port (this is the default).
- With the **cos** keyword, PFC QoS sets the internal DSCP value from received or ingress port CoS.
- With the **dscp** keyword, PFC QoS uses received DSCP.
- With the **ip-precedence** keyword, PFC QoS sets DSCP from received IP precedence.

Configuring Policy Map Class Policing

When you configure policy map class policing, note the following information:

- PFC QoS does not support the **set-qos-transmit** policer keyword.
- PFC QoS does not support the **set-dscp-transmit** or **set-prec-transmit** keywords as arguments to the **exceed-action** keyword.

- PFC QoS does not detect the use of unsupported keywords until you attach a policy map to an interface.

These sections describe configuration of policy map class policing:

- Using a Named Aggregate Policer, page 43-78
- Configuring a Per-Interface Policer, page 43-78

Note Policing with the **conform-action transmit** keywords sets the port trust state of matched traffic to trust DSCP or to the trust state configured by a **trust** command in the policy map class.

Using a Named Aggregate Policer

To use a named aggregate policer, perform this task:

Command	Purpose
Router(config-pmap-c)# **police aggregate** *aggregate_name*	Configures the policy map class to use a previously defined named aggregate policer.

Configuring a Per-Interface Policer

To configure a per-interface policer, perform this task:

Command	Purpose
Router(config-pmap-c)# **police** [**flow** [**mask** {**src-only** | **dest-only** | **full-flow**}]] *bits_per_second* *normal_burst_bytes* [*maximum_burst_bytes*] [**pir** *peak_rate_bps*] [[[**conform-action** {**drop** | **set-dscp-transmit** *dscp_value* | **set-prec-transmit** *ip_precedence_value* | **transmit**}] **exceed-action** {**drop** | **policed-dscp** | **transmit**}] **violate-action** {**drop** | **policed-dscp** | **transmit**}]	Creates a per-interface policer and configures the policy-map class to use it.

When configuring a per-interface policer, note the following information:

- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Each PFC or DFC polices independently, which might affect QoS features being applied to traffic that is distributed across the PFC and any DFCs. Examples of these QoS feature are:
  - Policers applied to a port channel interface.
  - Policers applied to a switched virtual interface.
  - Egress policers applied to either a Layer 3 interface or an SVI. Note that PFC QoS performs egress policing decisions at the ingress interface, on the PFC or ingress DFC.

  Policers affected by this restriction deliver an aggregate rate that is the sum of all the independent policing rates.

- When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown.
- You can apply aggregate and microflow policers to IPv6 traffic.
- Policing uses the Layer 2 frame size.
- See the "PFC QoS Configuration Guidelines and Restrictions" section on page 43-52 for information about rate and burst size granularity.
- You can enter the **flow** keyword to define a microflow policer (you cannot apply microflow policing to ARP traffic). When configuring a microflow policer, note the following information:
  - You can enter the **mask src-only** keywords to base flow identification only on source addresses, which applies the microflow policer to all traffic from each source address. PFC QoS supports the **mask src-only** keywords for both IP traffic and MAC traffic.
  - You can enter the **mask dest-only** keywords to base flow identification only on destination addresses, which applies the microflow policer to all traffic to each source address. PFC QoS supports the **mask dest-only** keywords for both IP traffic and MAC traffic.
  - By default and with the **mask full-flow** keywords, PFC QoS bases IP flow identification on source IP address, destination IP address, the Layer 3 protocol, and Layer 4 port numbers.
  - PFC QoS considers MAC-Layer traffic with the same protocol and the same source and destination MAC-Layer addresses to be part of the same flow, including traffic with different EtherTypes.
  - Microflow policers do not support the *maximum_burst_bytes* parameter, the **pir** *bits_per_second* keyword and parameter, or the **violate-action** keyword.

Note The flowmask requirements of microflow policing, NetFlow, and NetFlow data export (NDE) might conflict.

- The valid range of values for the CIR *bits_per_second* parameter is as follows:
  - Minimum—32 kilobits per second, entered as 32000
  - Maximum with Release 12.2(33)SXH1 and earlier—10 gigabits per second, entered as 10000000000
  - Maximum with Release 12.2(33)SXH2 and later—64 gigabits per second, entered as 64000000000
- The *normal_burst_bytes* parameter sets the CIR token bucket size.
- The *maximum_burst_bytes* parameter sets the PIR token bucket size (not supported with the **flow** keyword).
- When configuring the size of a token bucket, note the following information:
  - Because the token bucket must be large enough to hold at least one frame, configure the token bucket size to be larger than the maximum size of the traffic being policed.
  - For TCP traffic, configure the token bucket size as a multiple of the TCP window size, with a minimum value at least twice as large as the maximum size of the traffic being policed.
  - The *maximum_burst_bytes* parameter must be set larger than the *normal_burst_bytes* parameter.
  - To sustain a specific rate, set the token bucket size to be at least the rate value divided by 2000.

> **Note** To provide compatibility with QoS as implemented in the Catalyst operating system, Release 12.2(33)SXI and later releases support 1 byte as the minimum token bucket size, entered as 1.

  - With Release 12.2(33)SXI and later releases, the minimum token bucket size is 1 byte, entered as 1.
  - With releases earlier than Release 12.2(33)SXI, the minimum token bucket size is 1000 bytes, entered as 1000.
  - The maximum token bucket size is 512 megabytes, entered as 512000000.
- The valid range of values for the **pir** *bits_per_second* parameter (not supported with the **flow** keyword) is as follows:
  - Minimum—32 kilobits per second, entered as 32000 (the value cannot be smaller than the CIR *bits_per_second* parameters)
  - Maximum with Release 12.2(33)SXH1 and earlier—10 gigabits per second, entered as 10000000000
  - Maximum with Release 12.2(33)SXH2 and later—64 gigabits per second, entered as 64000000000
- (Optional) You can specify a conform action for matched in-profile traffic as follows:
  - The default conform action is **transmit**, which sets the policy map class trust state to *trust DSCP* unless the policy map class contains a **trust** command.
  - To set PFC QoS labels in untrusted traffic, you can enter the **set-dscp-transmit** keyword to mark matched untrusted traffic with a new DSCP value or enter the **set-prec-transmit** keyword to mark matched untrusted traffic with a new IP precedence value. The **set-dscp-transmit** and **set-prec-transmit** keywords are only supported for IP traffic. PFC QoS sets egress ToS and CoS from the configured value.
  - You can enter the **drop** keyword to drop all matched traffic.
  - Ensure that aggregate and microflow policers that are applied to the same traffic each specify the same conform-action behavior.
- (Optional) For traffic that exceeds the CIR, you can specify an exceed action as follows:
  - For marking without policing, you can enter the **transmit** keyword to transmit all matched out-of-profile traffic.
  - The default exceed action is **drop**, except with a *maximum_burst_bytes* parameter (**drop** is not supported with a *maximum_burst_bytes* parameter).

> **Note** When the exceed action is **drop**, PFC QoS ignores any configured violate action.

  - You can enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map.

Note When you create a policer that does not use the **pir** keyword and the *maximum_burst_bytes* parameter is equal to the *normal_burst_bytes* parameter (which is the case if you do not enter the *maximum_burst_bytes* parameter), the **exceed-action policed-dscp-transmit** keywords cause PFC QoS to mark traffic down as defined by the **policed-dscp max-burst** markdown map.

- (Optional—Not supported with the **flow** keyword) for traffic that exceeds the PIR, you can specify a violate action as follows:
  - For marking without policing, you can enter the **transmit** keyword to transmit all matched out-of-profile traffic.
  - The default violate action is equal to the exceed action.
  - You can enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map.

This example shows how to create a policy map named **max-pol-ipp5** that uses the class-map named **ipp5**, which is configured to trust received IP precedence values and is configured with a maximum-capacity aggregate policer and with a microflow policer:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# policy-map max-pol-ipp5
Router(config-pmap)# class ipp5
Router(config-pmap-c)# trust ip-precedence
Router(config-pmap-c)# police 2000000000 2000000 conform-action set-prec-transmit 6
exceed-action policed-dscp-transmit
Router(config-pmap-c)# police flow 10000000 10000 conform-action set-prec-transmit 6
exceed-action policed-dscp-transmit
Router(config-pmap-c)# end
```

Verifying Policy Map Configuration

To verify policy map configuration, perform this task:

	Command	Purpose
Step 1	`Router(config-pmap-c)# end`	Exits policy map class configuration mode. **Note** Enter additional **class** commands to create additional classes in the policy map.
Step 2	`Router# show policy-map policy_name`	Verifies the configuration.

This example shows how to verify the configuration:

```
Router# show policy-map max-pol-ipp5
 Policy Map max-pol-ipp5
  class  ipp5

  class ipp5
    police flow 10000000 10000 conform-action set-prec-transmit 6 exceed-action
policed-dscp-transmit
    trust precedence
    police 2000000000 2000000 2000000 conform-action set-prec-transmit 6 exceed-action
policed-dscp-transmit

Router#
```

Attaching a Policy Map to an Interface

To attach a policy map to an interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{vlan vlan_ID}	{type[1] slot/port[.subinterface]}	{port-channel number[.subinterface]}}`	Selects the interface to configure.
Step 2	`Router(config-if)# service-policy [input	output] policy_map_name`	Attaches a policy map to the interface.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show policy-map interface {{vlan vlan_ID}	{type[1] slot/port}	{port-channel number}}`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When attaching a policy map to an interface, note the following information:

- Do not attach a service policy to a port that is a member of an EtherChannel.
- PFC QoS supports the **output** keyword only on Layer 3 interfaces (either LAN ports configured as Layer 3 interfaces or VLAN interfaces).You can attach both an input and an output policy map to a Layer 3 interface.
- VLAN-based or port-based PFC QoS on Layer 2 ports is not relevant to policies attached to Layer 3 interfaces with the **output** keyword.
- Policies attached with the **output** keyword do not support microflow policing.
- You cannot attach a policy map that configures a trust state with the **service-policy output** command.
- Filtering based on IP precedence or DSCP in policies attached with the **output** keyword uses the received IP precedence or DSCP values. Filtering based on IP precedence or DSCP in policies attached with the **output** keyword is not based on any IP precedence or DSCP changes made by ingress QoS.
- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Each PFC or DFC polices independently, which might affect QoS features being applied to traffic that is distributed across the PFC and any DFCs. Examples of these QoS feature are:
  - Policers applied to a port channel interface.
  - Policers applied to a switched virtual interface.
  - Egress policers applied to either a Layer 3 interface or an SVI. Note that PFC QoS performs egress policing decisions at the ingress interface, on the PFC or ingress DFC.

  Policers affected by this restriction deliver an aggregate rate that is the sum of all the independent policing rates.
- When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown.

This example shows how to attach the policy map named **pmap1** to Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# service-policy input pmap1
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show policy-map interface fastethernet 5/36
 FastEthernet5/36
  service-policy input: pmap1
    class-map: cmap1 (match-all)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 5
  class cmap1
    police 8000 8000 conform-action transmit exceed-action drop
    class-map: cmap2 (match-any)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 2
        0 packets, 0 bytes
        5 minute rate 0 bps
  class cmap2
    police 8000 10000 conform-action transmit exceed-action drop
Router#
```

Configuring Dynamic Per-Session Attachment of a Policy Map

To configure and enable per-session QoS, perform these steps:

Step 1 Define the ingress and egress QoS policy maps to be assigned when users are authenticated.

Step 2 Configure identity policies to specify the policy maps to be assigned.

Step 3 In the user profiles on the RADIUS server, configure the Cisco vendor-specific attributes (VSAs) to specify which ingress and egress QoS policy maps will be assigned to each user.

To define the policy maps and associate them with an identity policy, follow these steps:

	Command	Purpose
Step 1	`Router(config)# policy-map in_policy_name`	Configures an ingress QoS policy map.
Step 2	`Router(config-pmap)# class class_map_name` ...	Configures policy map class.
Step 3	`Router(config-pmap-c)# exit`	Exits policy map class configuration submode.
Step 4	`Router(config)# policy-map out_policy_name`	Configures an egress QoS policy map.
Step 5	`Router(config-pmap)# class class_map_name` ...	Configures policy map class.
Step 6	`Router(config-pmap-c)# exit`	Exits policy map class configuration submode.
Step 7	`Router(config)# identity policy policy1`	Creates an identity policy, and enters identity policy configuration submode.

	Command	Purpose
Step 8	`Router(config-identity-policy)#` **service-policy type qos input** *in_policy_name*	Associates the ingress QoS policy map with this identity.
Step 9	`Router(config-identity-policy)#` **service-policy type qos output** *out_policy_name*	Associates the egress QoS policy map with this identity.
Step 10	`Router(config-identity-policy)#` **end**	Exits identity policy configuration submode and returns to privileged EXEC mode.
Step 11	`Router#` **show epm session** [**summary** | **ip** *ip_addr* | **mac** *mac_addr*]	Verifies the configuration when a session is active on the interface.

To remove the identity policy from the switch, use the **no identity policy** *policy_name* command.

After the policy maps have been defined on the switch, configure the Cisco AV pair attributes in each user profile on the RADIUS server using the policy map names:

- cisco-avpair = "ip:sub-policy-In=*in_policy_name*"
- cisco-avpair = "ip:sub-policy-Out=*out_policy_name*"

To set the Cisco AV pair attributes on the RADIUS server, perform the following task:

Command or Action	Purpose
sub-policy-In=*in_policy_name* **sub-policy-Out=***out_policy_name*	Enters the two Cisco AV pairs for service policy on the RADIUS server in the user file. When the switch requests the policy name, this information in the user file is supplied. A RADIUS user file contains an entry for each user that the RADIUS server will authenticate. Each entry, which is also referred to as a *user profile*, establishes an attribute the user can access. In this example, you have configured a service policy that attaches a QoS policy map to the interface and specifies the direction (inbound for data packets traveling into the interface or outbound for data packets leaving the interface). The policy map applied in the inbound direction is example_in_qos and the outbound policy map is example_out_qos.

This example shows the configuration in the user file on the RADIUS server:

```
userid    Password ="cisco"
   Service-Type = Framed,
   Framed-Protocol = PPP,
   cisco-avpair = "sub-policy-In=example_in_qos",
   cisco-avpair = "sub-policy-Out=example_out_qos"
```

This example shows the output of the **show epm session summary** command when a session is active:

```
Router# show epm session summary

EPM Session Information
-----------------------
Total sessions seen so far : 5
Total active sessions      : 1
Session IP Address         : 192.0.2.1
-------------------
```

This example shows the output of the **show epm session ip** *ip_addr* command when a session is active on the interface with IP address 192.0.2.1:

```
Router# show epm session ip 192.0.2.1

Admission feature        : AUTHPROXY
AAA Policies             :
Input Service Policy     : in_policy_name
Output Service Policy    : out_policy_name
```

Configuring Egress DSCP Mutation

These sections describe how to configure egress DSCP mutation:

- Configuring Named DSCP Mutation Maps, page 43-85
- Attaching an Egress DSCP Mutation Map to an Interface, page 43-86

Configuring Named DSCP Mutation Maps

To configure a named DSCP mutation map, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos map dscp-mutation** *map_name dscp1* [*dscp2* [*dscp3* [*dscp4* [*dscp5* [*dscp6* [*dscp7* [*dscp8*]]]]]]] **to** *mutated_dscp*	Configures a named DSCP mutation map.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show mls qos maps**	Verifies the configuration.

When configuring a named DSCP mutation map, note the following information:

- You can enter up to 8 DSCP values that map to a mutated DSCP value.
- You can enter multiple commands to map additional DSCP values to a mutated DSCP value.
- You can enter a separate command for each mutated DSCP value.

This example shows how to map DSCP 30 to mutated DSCP value 8:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls qos map dscp-mutation mutmap1 30 to 8
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos map | begin DSCP mutation
DSCP mutation map mutmap1:                  (dscp= d1d2)
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     -------------------------------------
      0 :    00 01 02 03 04 05 06 07 08 09
      1 :    10 11 12 13 14 15 16 17 18 19
      2 :    20 21 22 23 24 25 26 27 28 29
      3 :    08 31 32 33 34 35 36 37 38 39
      4 :    40 41 42 43 44 45 46 47 48 49
      5 :    50 51 52 53 54 55 56 57 58 59
      6 :    60 61 62 63
```

```
<...Output Truncated...>
Router#
```

Note In the DSCP mutation map displays, the marked-down DSCP values are shown in the body of the matrix; the first digit of the original DSCP value is in the column labeled d1 and the second digit is in the top row. In the example shown, DSCP 30 maps to DSCP 08.

Attaching an Egress DSCP Mutation Map to an Interface

To attach an egress DSCP mutation map to an interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`vlan`** *`vlan_ID`*`} | {`*`type`*[1] *`slot/port`*`[.`*`subinterface`*`]} | {`**`port-channel`** *`number`*`[.`*`subinterface`*`]}}`	Selects the interface to configure.
Step 2	`Router(config-if)#` **`mls qos dscp-mutation`** *`mutation_map_name`*	Attaches an egress DSCP mutation map to the interface.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show running-config interface`** `{{`**`vlan`** *`vlan_ID`*`} | {`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`number`*`}}`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to attach the egress DSCP mutation map named mutmap1 to Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# mls qos dscp-mutation mutmap1
Router(config-if)# end
```

Configuring Ingress CoS Mutation on IEEE 802.1Q Tunnel Ports

IEEE 802.1Q tunnel ports configured to trust received CoS support ingress CoS mutation (see the "Applying Ingress CoS Mutation Maps to IEEE 802.1Q Tunnel Ports" section on page 43-88 for the list of supported modules).

When you configure ingress CoS mutation on an IEEE 802.1Q tunnel port that you have configured to trust received CoS, PFC QoS uses the mutated CoS value instead of the received CoS value in the ingress drop thresholds and for any trust CoS marking and policing.

These sections describe how to configure ingress CoS mutation:

Ingress CoS Mutation Configuration Guidelines and Restrictions

When configuring ingress CoS mutation, follow these guidelines and restrictions:

- Ports that are not configured as IEEE 802.1Q tunnel ports do not support ingress CoS mutation.
- Ports that are not configured to trust received CoS do not support ingress CoS mutation.
- Ingress CoS mutation does not change the CoS value carried by the customer frames. When the customer traffic exits the 802.1Q tunnel, the original CoS is intact.
- The WS-X6704-10GE, WS-X6748-SFP, WS-X6724-SFP, and WS-X6748-GE-TX switching modules support ingress CoS mutation.
- Ingress CoS mutation configuration applies to all ports in a port group. The port groups are:
  - WS-X6704-10GE—4 ports, 4 port groups, 1 port in each group
  - WS-X6748-SFP—48 ports, 4 port groups: ports 1–12, 13–24, 25–36, and 37–48
  - WS-X6724-SFP—24 ports, 2 port groups: ports 1–12 and 13–24
  - WS-X6748-GE-TX—48 ports, 4 port groups: ports 1–12, 13–24, 25–36, and 37–48
- To avoid ingress CoS mutation configuration failures, only create EtherChannels where all member ports support ingress CoS mutation or where no member ports support ingress CoS mutation. Do not create EtherChannels with mixed support for ingress CoS mutation.
- If you configure ingress CoS mutation on a port that is a member of an EtherChannel, the ingress CoS mutation is applied to the port-channel interface.
- You can configure ingress CoS mutation on port-channel interfaces.
- With ingress CoS mutation configured on a port-channel interface, the following occurs:
  - The ingress CoS mutation configuration is applied to the port groups of all member ports of the EtherChannel. If any member port cannot support ingress CoS mutation, the configuration fails.
  - If a port in the port group is a member of a second EtherChannel, the ingress CoS mutation configuration is applied to the second port-channel interface and to the port groups of all member ports of the second EtherChannel. If any member port of the second EtherChannel cannot support ingress CoS mutation, the configuration fails on the first EtherChannel. If the configuration originated on a nonmember port in a port group that has a member port of the first EtherChannel, the configuration fails on the nonmember port.
  - The ingress CoS mutation configuration propagates without limit through port groups, member ports, and port-channel interfaces, regardless of whether or not the ports are configured to trust CoS or are configured as IEEE 802.1Q tunnel ports.
- An EtherChannel where you want to configure ingress CoS mutation must not have member ports that are in port groups containing member ports of other EtherChannels that have member ports that do not support ingress CoS mutation. (This restriction extends without limit through all port-group-linked member ports and port-channel-interface-linked ports.)
- A port where you want to configure ingress CoS mutation must not be in a port group that has a member port of an EtherChannel that has members that do not support ingress CoS mutation. (This restriction extends without limit through all port-group-linked member ports and port-channel-interface-linked ports.)
- There can be only be one ingress CoS mutation configuration applied to all port-group-linked member ports and port-channel-interface-linked ports.

Configuring Ingress CoS Mutation Maps

To configure an ingress CoS mutation map, perform this task:

	Command	Purpose
Step 1	`Router(config)# `**`mls qos map cos-mutation`** *`mutation_map_name mutated_cos1 mutated_cos2 mutated_cos3 mutated_cos4 mutated_cos5 mutated_cos6 mutated_cos7 mutated_cos8`*	Configures an ingress CoS mutation map. You must enter 8 mutated CoS values to which PFC QoS maps ingress CoS values 0 through 7.
Step 2	`Router(config)# `**`end`**	Exits configuration mode.
Step 3	`Router# `**`show mls qos maps cos-mutation`**	Verifies the configuration.

This example shows how to configure a CoS mutation map named testmap:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls qos map cos-mutation testmap 4 5 6 7 0 1 2 3
Router(config)# end
Router#
```

This example shows how to verify the map configuration:

```
Router(config)# show mls qos maps cos-mutation
COS mutation map testmap
cos-in  :   0  1  2  3  4  5  6  7
-----------------------------------
cos-out :   4  5  6  7  0  1  2  3
Router#
```

Applying Ingress CoS Mutation Maps to IEEE 802.1Q Tunnel Ports

To attach an ingress CoS mutation map to an IEEE 802.1Q tunnel port, perform this task:

	Command	Purpose
Step 1	`Router(config)# `**`interface`** `{{`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`number`*`}}`	Selects the interface to configure.
Step 2	`Router(config-if)# `**`mls qos cos-mutation`** *`mutation_map_name`*	Attaches an ingress CoS mutation map to the interface.
Step 3	`Router(config-if)# `**`end`**	Exits configuration mode.
Step 4	`Router# `**`show running-config interface`** `{{`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`number`*`}}` `Router# `**`show mls qos maps cos-mutation`**	Verifies the configuration.

1. *type* = **gigabitethernet** or **tengigabitethernet**

This example shows how to attach the ingress CoS mutation map named testmap to Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# mls qos cos-mutation testmap
Router(config-if)# end
Router# show mls qos maps cos-mutation
COS mutation map testmap
```

```
cos-in  :   0  1  2  3  4  5  6  7
-----------------------------------
cos-out :   4  5  6  7  0  1  2  3

testmap is attached on the following interfaces
Gi1/1
Router#
```

Configuring DSCP Value Maps

These sections describe how DSCP values are mapped to other values:

- Mapping Received CoS Values to Internal DSCP Values, page 43-89
- Mapping Received IP Precedence Values to Internal DSCP Values, page 43-90
- Configuring DSCP Markdown Values, page 43-90
- Mapping Internal DSCP Values to Egress CoS Values, page 43-92

Mapping Received CoS Values to Internal DSCP Values

To configure the mapping of received CoS values to the DSCP value that PFC QoS uses internally on the PFC, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos map cos-dscp** *dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8*	Configures the received CoS to internal DSCP map. You must enter 8 DSCP values to which PFC QoS maps CoS values 0 through 7.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show mls qos maps**	Verifies the configuration.

This example shows how to configure the received CoS to internal DSCP map:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls qos map cos-dscp 0 1 2 3 4 5 6 7
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos maps | begin Cos-dscp map
   Cos-dscp map:
        cos:   0  1  2  3  4  5  6  7
     ----------------------------------
       dscp:   0  1  2  3  4  5  6  7
<...Output Truncated...>
Router#
```

Mapping Received IP Precedence Values to Internal DSCP Values

To configure the mapping of received IP precedence values to the DSCP value that PFC QoS uses internally on the PFC, perform this task:

	Command	Purpose
Step 1	`Router(config)# `**`mls qos map ip-prec-dscp`** *`dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8`*	Configures the received IP precedence to internal DSCP map. You must enter 8 internal DSCP values to which PFC QoS maps received IP precedence values 0 through 7.
Step 2	`Router(config)# `**`end`**	Exits configuration mode.
Step 3	`Router# `**`show mls qos maps`**	Verifies the configuration.

This example shows how to configure the received IP precedence to internal DSCP map:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls qos map ip-prec-dscp 0 1 2 3 4 5 6 7
Router(config)# end
```

This example shows how to verify the configuration:

```
Router# show mls qos maps | begin IpPrecedence-dscp map
   IpPrecedence-dscp map:
     ipprec:   0  1  2  3  4  5  6  7
     ----------------------------------
       dscp:   0  1  2  3  4  5  6  7
<...Output Truncated...>
Router#
```

Configuring DSCP Markdown Values

To configure the mapping of DSCP markdown values used by policers, perform this task:

	Command	Purpose
Step 1	`Router(config)# `**`mls qos map policed-dscp`** {**`normal-burst`** | **`max-burst`**} *`dscp1`* [*`dscp2`* [*`dscp3`* [*`dscp4`* [*`dscp5`* [*`dscp6`* [*`dscp7`* [*`dscp8`*]]]]]]] **`to`** *`markdown_dscp`*	Configures a DSCP markdown map.
Step 2	`Router(config)# `**`end`**	Exits configuration mode.
Step 3	`Router# `**`show mls qos maps`**	Verifies the configuration.

When configuring a DSCP markdown map, note the following information:

- You can enter the **normal-burst** keyword to configure the markdown map used by the **exceed-action policed-dscp-transmit** keywords.
- You can enter the **max-burst** keyword to configure the markdown map used by the **violate-action policed-dscp-transmit** keywords.

Note When you create a policer that does not use the **pir** keyword, and the *maximum_burst_bytes* parameter is equal to the *normal_burst_bytes* parameter (which occurs if you do not enter the *maximum_burst_bytes* parameter), the **exceed-action policed-dscp-transmit** keywords cause PFC QoS to mark traffic down as defined by the **policed-dscp max-burst** markdown map.

- To avoid out-of-sequence packets, configure the markdown maps so that conforming and nonconforming traffic uses the same queue.
- You can enter up to 8 DSCP values that map to a marked-down DSCP value.
- You can enter multiple commands to map additional DSCP values to a marked-down DSCP value.
- You can enter a separate command for each marked-down DSCP value.

Note Configure marked-down DSCP values that map to CoS values consistent with the markdown penalty.

This example shows how to map DSCP 1 to marked-down DSCP value 0:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls qos map policed-dscp normal-burst 1 to 0
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos map
   Normal Burst Policed-dscp map:                                  (dscp= d1d2)
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ------------------------------------
      0 :     00 01 02 03 04 05 06 07 08 09
      1 :     10 11 12 13 14 15 16 17 18 19
      2 :     20 21 22 23 24 25 26 27 28 29
      3 :     30 31 32 33 34 35 36 37 38 39
      4 :     40 41 42 43 44 45 46 47 48 49
      5 :     50 51 52 53 54 55 56 57 58 59
      6 :     60 61 62 63

   Maximum Burst Policed-dscp map:                                  (dscp= d1d2)
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ------------------------------------
      0 :     00 01 02 03 04 05 06 07 08 09
      1 :     10 11 12 13 14 15 16 17 18 19
      2 :     20 21 22 23 24 25 26 27 28 29
      3 :     30 31 32 33 34 35 36 37 38 39
      4 :     40 41 42 43 44 45 46 47 48 49
      5 :     50 51 52 53 54 55 56 57 58 59
      6 :     60 61 62 63
<...Output Truncated...>
Router#
```

Note In the Policed-dscp displays, the marked-down DSCP values are shown in the body of the matrix; the first digit of the original DSCP value is in the column labeled d1 and the second digit is in the top row. In the example shown, DSCP 41 maps to DSCP 41.

Mapping Internal DSCP Values to Egress CoS Values

To configure the mapping of the DSCP value that PFC QoS uses internally on the PFC to the CoS value used for egress LAN port scheduling and congestion avoidance, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos map dscp-cos dscp1 [dscp2 [dscp3 [dscp4 [dscp5 [dscp6 [dscp7 [dscp8]]]]]]] to cos_value`	Configures the internal DSCP to egress CoS map.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos maps`	Verifies the configuration.

When configuring the internal DSCP to egress CoS map, note the following information:

- You can enter up to 8 DSCP values that PFC QoS maps to a CoS value.
- You can enter multiple commands to map additional DSCP values to a CoS value.
- You can enter a separate command for each CoS value.

This example shows how to configure internal DSCP values 0, 8, 16, 24, 32, 40, 48, and 54 to be mapped to egress CoS value 0:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls qos map dscp-cos 0 8 16 24 32 40 48 54 to 0
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos map | begin Dscp-cos map
   Dscp-cos map:                                   (dscp= d1d2)
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     -------------------------------------
      0 :    00 00 00 00 00 00 00 00 00 01
      1 :    01 01 01 01 01 01 00 02 02 02
      2 :    02 02 02 02 00 03 03 03 03 03
      3 :    03 03 00 04 04 04 04 04 04 04
      4 :    00 05 05 05 05 05 05 05 00 06
      5 :    06 06 06 06 00 06 07 07 07 07
      6 :    07 07 07 07
<...Output Truncated...>
Router#
```

Note In the Dscp-cos map display, the CoS values are shown in the body of the matrix; the first digit of the DSCP value is in the column labeled d1 and the second digit is in the top row. In the example shown, DSCP values 41 through 47 all map to CoS 05.

Configuring the Trust State of Ethernet LAN Ports

By default, all ports are untrusted. You can configure the port trust state on all Ethernet LAN ports.

Note On non-Gigabit Ethernet **1q4t/2q2t** ports, you must repeat the trust configuration in a class map.

To configure the trust state of a port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{type[1] slot/port} | {port-channel number}}`	Selects the interface to configure.
Step 2	`Router(config-if)# mls qos trust [dscp | ip-precedence | cos[2]]`	Configures the trust state of the port.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show queueing interface type[1] slot/port | include Trust state`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
2. Not supported for **serial**, **pos** or **atm** interface types.

When configuring the trust state of a port, note the following information:

- To apply a non-default trust configuration only when a Cisco IP phone is attached to the port, see the "Configuring Trusted Boundary with Cisco Device Verification" section on page 43-94.
- To configure QoS on an attached IP phone, see the "Configuring Cisco IP Phone Support" section on page 15-5.
- With no other keywords, the **mls qos trust** command is equivalent to **mls qos trust dscp**.
- You can use the **mls qos trust dscp** command to enable DSCP-based receive-queue drop thresholds on WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports (see the "Configuring DSCP-Based Queue Mapping" section on page 43-101). To avoid dropping traffic because of inconsistent DSCP values when DSCP-based queue mapping is enabled, configure ports with the **mls qos trust dscp** command only when the received traffic carries DSCP values that you know to be consistent with network policy.
- The **mls qos trust cos** command enables CoS-based receive-queue drop thresholds. To avoid dropping traffic because of inconsistent CoS values, configure ports with the **mls qos trust cos** command only when the received traffic is ISL or 802.1Q frames carrying CoS values that you know to be consistent with network policy.
- You can configure IEEE 8021.Q tunnel ports configured with the **mls qos trust cos** command to use a mutated CoS value instead of the received CoS value ("Configuring Ingress CoS Mutation on IEEE 802.1Q Tunnel Ports" section on page 43-86).
- Use the **no mls qos trust** command to set the port state to untrusted.

This example shows how to configure Gigabit Ethernet port 1/1 with the **trust cos** keywords:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# mls qos trust cos
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | include trust
  Trust state: trust COS
Router#
```

Configuring Trusted Boundary with Cisco Device Verification

Release 12.2(33)SXI1 and later releases support the trusted boundary with Cisco device verification feature, which configures Ethernet LAN ports to use CDP to detect whether or not a Cisco IP phone is attached to the port.

- If CDP detects a Cisco IP phone, QoS applies a configured **mls qos trust dscp**, **mls qos trust ip-precedence**, or **mls qos trust cos** interface command.
- If CDP does not detect a Cisco IP phone, QoS ignores any configured nondefault trust state.

To configure trusted boundary with Cisco device verification, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{type1 slot/port}	{port-channel number}}`	Selects the interface to configure.
Step 2	`Router(config-if)# mls qos trust device cisco-phone`	Configures trusted boundary with Cisco device verification.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show queueing interface type1 slot/port	include [Tt]rust`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, **tengigabitethernet**.

When configuring trusted boundary with Cisco device verification, note the following information:

- CDP must be enabled on the port to use trusted boundary with Cisco device verification.
- To configure QoS on an attached IP phone, see the "Configuring Cisco IP Phone Support" section on page 15-5.

This example shows how to configure trusted boundary with Cisco device verification on Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# mls qos trust device cisco-phone
Router(config-if)# end
Router#
```

This example shows how to verify the configuration on a port configured to trust CoS, but that does not have a Cisco IP phone attached:

```
Router# show queueing interface gigabitethernet 1/1 | include [Tt]rust
Trust boundary enabled
  Port is untrusted
  Extend trust state: not trusted [COS = 0]
Router#
```

Configuring the Ingress LAN Port CoS Value

Note Whether or not PFC QoS uses the CoS value applied with the **mls qos cos** command depends on the trust state of the port and the trust state of the traffic received through the port. The **mls qos cos** command does not configure the trust state of the port or the trust state of the traffic received through the port.

To use the CoS value applied with the **mls qos cos** command as the basis of internal DSCP:

- On a port that receives only untagged ingress traffic, configure the ingress port as trusted or configure a trust CoS policy map that matches the ingress traffic.
- On a port that receives tagged ingress traffic, configure a trust CoS policy map that matches the ingress traffic.

You can configure the CoS value that PFC QoS assigns to untagged frames from ingress LAN ports configured as trusted and to all frames from ingress LAN ports configured as untrusted.

To configure the CoS value for an ingress LAN port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`*`type`*[1] *`slot/port`*`}	{`**`port-channel`** *`number`*`}}`	Selects the interface to configure.
Step 2	`Router(config-if)#` **`mls qos cos`** *`port_cos`*	Configures the ingress LAN port CoS value.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show queuing interface`** `{`**`ethernet`** `|` **`fastethernet`** `|` **`gigabitethernet`**`}` *`slot/port`*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure the CoS value 5 on Fast Ethernet port 5/24 and verify the configuration:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/24
Router(config-if)# mls qos cos 5
Router(config-if)# end
Router# show queueing interface fastethernet 5/24 | include Default COS
  Default COS is 5
Router#
```

Configuring Standard-Queue Drop Threshold Percentages

These sections describe configuring standard-queue drop threshold percentages:

- Configuring a Tail-Drop Receive Queue, page 43-96
- Configuring a WRED-Drop Transmit Queue, page 43-97
- Configuring a WRED-Drop and Tail-Drop Receive Queue, page 43-98
- Configuring a WRED-Drop and Tail-Drop Transmit Queue, page 43-98
- Configuring 1q4t/2q2t Tail-Drop Threshold Percentages, page 43-99

Note

- Enter the **show queueing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command to see the queue structure of a port.
- **1p1q0t** ports have no configurable thresholds.

- **1p3q1t** (transmit), **1p2q1t** (transmit), **1p7q2t** (receive), and **1p1q8t** (receive) ports also have nonconfigurable tail-drop thresholds.

When configuring thresholds, note the following information:

- Queue number 1 is the lowest-priority standard queue.
- Higher-numbered queues are higher priority standard queues.
- Receive-queue parameters can be configured only on trusted ports.
- When configuring minimum and maximum threshold values, you cannot configure minimum values to be larger than maximumvalues.

When you configure multiple-threshold standard queues, note the following information:

- The first percentage that you enter sets the lowest-priority threshold.
- The second percentage that you enter sets the next highest-priority threshold.
- The last percentage that you enter sets the highest-priority threshold.
- The percentages range from 1 to 100. A value of 10 indicates a threshold when the buffer is 10-percent full.
- Always set highest-numbered threshold to 100 percent.

When configuring the WRED-drop thresholds, note the following information:

- Each WRED-drop threshold has a low-WRED and a high-WRED value.
- Low-WRED and high-WRED values are a percentage of the queue capacity (the range is from 1 to 100).
- The low-WRED value is the traffic level under which no traffic is dropped. The low-WRED value must be lower than the high-WRED value.
- The high-WRED value is the traffic level above which all traffic is dropped.
- Traffic in the queue between the low- and high-WRED values has an increasing chance of being dropped as the queue fills.

Configuring a Tail-Drop Receive Queue

These port types have only tail-drop thresholds in their receive-queues:

- **1q2t**
- **1p1q4t**
- **2q8t**
- **1q8t**

To configure the drop thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{`**`fastethernet`** `|` **`gigabitethernet`**`}` *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`rcv-queue threshold`** *`queue_id thr1% thr2% thr3% thr4%`* `{`*`thr5% thr6% thr7% thr8%`*`}`	Configures the receive-queue tail-drop threshold percentages.

	Command	Purpose
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show queueing interface`** `{`**`fastethernet`** `|` **`gigabitethernet`**`}` *`slot/port`*	Verifies the configuration.

This example shows how to configure the receive-queue drop thresholds for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# rcv-queue threshold 1 60 75 85 100
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | begin Receive queues
Receive queues [type = 1p1q4t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         Standard            4
       2         Priority            1

  Trust state: trust COS

    queue tail-drop-thresholds
    --------------------------
    1     60[1] 75[2] 85[3] 100[4]
<...Output Truncated...>
Router#
```

Configuring a WRED-Drop Transmit Queue

These port types have only WRED-drop thresholds in their transmit queues:

- **1p2q2t** (transmit)
- **1p2q1t** (transmit)

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `type`[1] *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`wrr-queue random-detect min-threshold`** *`queue_id thr1%`* `[`*`thr2%`*`]`	Configures the low WRED-drop thresholds.
Step 3	`Router(config-if)#` **`wrr-queue random-detect max-threshold`** *`queue_id thr1%`* `[`*`thr2%`*`]`	Configures the high WRED-drop thresholds.
Step 4	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 5	`Router#` **`show queueing interface`** *`type`*[1] *`slot/port`*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Configuring a WRED-Drop and Tail-Drop Receive Queue

These port types have both WRED-drop and tail-drop thresholds in their receive queues:

- **8q4t** (receive)
- **8q8t** (receive)
- **1p7q2t** (receive)
- **1p1q8t** (receive)

To configure the drop thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`rcv-queue threshold`** *`queue_id thr1% thr2% thr3% thr4% thr5% thr6% thr7% thr8%`*	Configures the tail-drop thresholds.
Step 3	`Router(config-if)#` **`rcv-queue random-detect min-threshold`** *`queue_id thr1% thr2% thr3% thr4% thr5% thr6% thr7% thr8%`*	Configures the low WRED-drop thresholds.
Step 4	`Router(config-if)#` **`rcv-queue random-detect max-threshold`** *`queue_id thr1% thr2% thr3% thr4% thr5% thr6% thr7% thr8%`*	Configures the high WRED-drop thresholds.
Step 5	`Router(config-if)#` **`rcv-queue random-detect`** *`queue_id`*	Enables WRED-drop thresholds.
Step 6	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 7	`Router#` **`show queueing interface`** *`type`*[1] *`slot/port`*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Configuring a WRED-Drop and Tail-Drop Transmit Queue

These port types have both WRED-drop and tail-drop thresholds in their transmit queues:

- **1p3q1t** (transmit)
- **1p3q8t** (transmit)
- **1p7q8t** (transmit)

To configure the drop thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`wrr-queue threshold`** *`queue_id thr1% [thr2% thr3% thr4% thr5% thr6% thr7% thr8%]`*	Configures the tail-drop thresholds.
Step 3	`Router(config-if)#` **`wrr-queue random-detect min-threshold`** *`queue_id thr1% [thr2% thr3% thr4% thr5% thr6% thr7% thr8%]`*	Configures the low WRED-drop thresholds.
Step 4	`Router(config-if)#` **`wrr-queue random-detect max-threshold`** *`queue_id thr1% [thr2% thr3% thr4% thr5% thr6% thr7% thr8%]`*	Configures the high WRED-drop thresholds.
Step 5	`Router(config-if)#` **`wrr-queue random-detect`** *`queue_id`*	Enables WRED-drop thresholds.

	Command	Purpose
Step 6	Router(config-if)# **end**	Exits configuration mode.
Step 7	Router# **show queueing interface** *type*[1] *slot/port*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure the low-priority transmit queue high-WRED-drop thresholds for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# wrr-queue random-detect max-threshold 1 70 70
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | begin Transmit queues
  Transmit queues [type = 1p2q2t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         WRR low             2
       2         WRR high            2
       3         Priority            1

    queue random-detect-max-thresholds
    ----------------------------------
      1    40[1] 70[2]
      2    40[1] 70[2]
<...Output Truncated...>
Router#
```

Configuring 1q4t/2q2t Tail-Drop Threshold Percentages

On **1q4t/2q2t** ports, the receive- and transmit-queue drop thresholds have this relationship:

- Receive queue 1 (standard) threshold 1 = transmit queue 1 (standard low priority) threshold 1
- Receive queue 1 (standard) threshold 2 = transmit queue 1 (standard low priority) threshold 2
- Receive queue 1 (standard) threshold 3 = transmit queue 2 (standard high priority) threshold 1
- Receive queue 1 (standard) threshold 4 = transmit queue 2 (standard high priority) threshold 2

To configure tail-drop threshold percentages for the standard receive and transmit queues on **1q4t/2q2t** LAN ports, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {**ethernet** | **fastethernet** | **gigabitethernet**} *slot/port*	Selects the interface to configure.
Step 2	Router(config-if)# **wrr-queue threshold** *queue_id thr1% thr2%*	Configures the receive- and transmit-queue tail-drop thresholds.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# show queueing interface {**ethernet** | **fastethernet** | **gigabitethernet**} *slot/port*	Verifies the configuration.

When configuring the receive- and transmit-queue tail-drop thresholds, note the following information:

- You must use the transmit queue and threshold numbers.
- The *queue_id* is 1 for the standard low-priority queue and 2 for the standard high-priority queue.
- The percentages range from 1 to 100. A value of 10 indicates a threshold when the buffer is 10-percent full.
- Always set threshold 2 to 100 percent.
- Ethernet and Fast Ethernet **1q4t** ports do not support receive-queue tail-drop thresholds.

This example shows how to configure receive queue 1/threshold 1 and transmit queue 1/threshold 1 for Gigabit Ethernet port 2/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 2/1
Router(config-if)# wrr-queue threshold 1 60 100
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 2/1
  Transmit queues [type = 2q2t]:

<...Output Truncated...>

queue tail-drop-thresholds
    --------------------------
    1     60[1] 100[2]
    2     40[1] 100[2]

<...Output Truncated...>

Receive queues [type = 1q4t]:

<...Output Truncated...>

queue tail-drop-thresholds
    --------------------------
    1     60[1] 100[2] 40[3] 100[4]
<...Output Truncated...>
Router#
```

Mapping QoS Labels to Queues and Drop Thresholds

These sections describe how to map QoS labels to queues and drop thresholds:

> **Note** Enter the **show queueing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command to see the queue structure of a port.

These sections describe how to map QoS labels to queues and drop thresholds:

- Queue and Drop Threshold Mapping Guidelines and Restrictions, page 43-101
- Configuring DSCP-Based Queue Mapping, page 43-101
- Configuring CoS-Based Queue Mapping, page 43-106

Queue and Drop Threshold Mapping Guidelines and Restrictions

When mapping QoS labels to queues and thresholds, note the following information:

- When SRR is enabled, you cannot map any CoS values or DSCP values to strict-priority queues.
- Queue number 1 is the lowest-priority standard queue.
- Higher-numbered queues are higher priority standard queues.
- You can map up to 8 CoS values to a threshold.
- You can map up to 64 DSCP values to a threshold.
- Threshold 0 is a nonconfigurable 100-percent tail-drop threshold on these port types:
  - **1p1q0t** (receive)
  - **1p1q8t** (receive)
  - **1p3q1t** (transmit)
  - **1p2q1t** (transmit)
- The standard queue thresholds can be configured as either tail-drop or WRED-drop thresholds on these port types:
  - **1p1q8t** (receive)
  - **1p3q1t** (transmit)
  - **1p3q8t** (transmit)
  - **1p7q1t** (transmit)

Configuring DSCP-Based Queue Mapping

These sections describe how to configure DSCP-based queue mapping:

- Configuring Ingress DSCP-Based Queue Mapping, page 43-102
- Mapping DSCP Values to Standard Transmit-Queue Thresholds, page 43-104
- Mapping DSCP Values to the Transmit Strict-Priority Queue, page 43-105

Note

- DSCP-based queue mapping is supported on WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE ports.
- To configure DSCP-based queue mapping on Supervisor Engine 720-10GE ports, you must enter **shutdown** interface configuration mode commands for the Supervisor Engine 720-10GE Gigabit Ethernet ports, and then enter the **mls qos 10g-only** global configuration command, which disables the Gigabit Ethernet ports on the Supervisor Engine 720-10GE.

Enabling DSCP-Based Queue Mapping

To enable DSCP-based queue mapping, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface tengigabitethernet slot/port`	Selects the interface to configure.
Step 2	`Router(config-if)# mls qos queue-mode mode-dscp`	Enables DSCP-based queue mapping.

	Command	Purpose
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show queueing interface tengigabitethernet slot/port | include Queueing Mode`	Verifies the configuration.

This example shows how to enable DSCP-based queue mapping on 10-Gigabit Ethernet port 6/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 6/1
Router(config-if)# mls qos queue-mode mode-dscp
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show queueing interface tengigabitethernet 6/1 | include Queueing Mode
    Queueing Mode In Tx direction: mode-dscp
    Queueing Mode In Rx direction: mode-dscp
```

Configuring Ingress DSCP-Based Queue Mapping

Ingress DSCP-to-queue mapping is supported only on ports configured to trust DSCP.

These sections describe how to configure ingress DSCP-based queue mapping:

- Enabling DSCP-Based Queue Mapping, page 43-101
- Mapping DSCP Values to Standard Receive-Queue Thresholds, page 43-103

Configuring the Port to Trust DSCP

To configure the port to trust DSCP perform this task:

	Command	Purpose
Step 1	`Router(config)# interface tengigabitethernet slot/port`	Selects the interface to configure.
Step 2	`Router(config-if)# mls qos trust dscp`	Configures the port to trust received DSCP values.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show queueing interface tengigabitethernet slot/port | include Trust state`	Verifies the configuration.

This example shows how to configure 10-Gigabit Ethernet port 6/1 to trust received DSCP values:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 6/1
Router(config-if)# mls qos trust dscp
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 6/1 | include Trust state
  Trust state: trust DSCP
```

Mapping DSCP Values to Standard Receive-Queue Thresholds

To map DSCP values to the standard receive-queue thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface tengigabitethernet`** *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`rcv-queue dscp-map`** *`queue_# threshold_# dscp1`* `[`*`dscp2`* `[`*`dscp3`* `[`*`dscp4`* `[`*`dscp5`* `[`*`dscp6`* `[`*`dscp7`* `[`*`dscp8`*`]]]]]]]`	Maps DSCP values to the standard receive queue thresholds.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show queueing interface tengigabitethernet`** *`slot/port`*	Verifies the configuration.

When mapping DSCP values, note the following information:

- You can enter up to 8 DSCP values that map to a queue and threshold.
- You can enter multiple commands to map additional DSCP values to the queue and threshold.
- You must enter a separate command for each queue and threshold.

This example shows how to map the DSCP values 0 and 1 to threshold 1 in the standard receive queue for 10-Gigabit Ethernet port 6/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 6/1
Router(config-if)# rcv-queue dscp-map 1 1 0 1
Router(config-if)# end
Router#
```

Note The receive queue mapping is shown in the second "queue thresh dscp-map" displayed by the **show queueing interface** command.

This example shows how to verify the configuration:

```
Router# show queueing interface tengigabitethernet 1/1 | begin queue thresh dscp-map
<...Output Truncated...>
queue thresh dscp-map
---------------------------------------
    1     1      0-9 11 13-17 19 21-25 27 29-39 48-63
    1     2      12 20 28
    1     3      10 18 26
    1     4
    2     1
    2     2
    2     3
    2     4
    3     1
    3     2
    3     3
    3     4
    4     1
    4     2
    4     3
    4     4
    5     1
    5     2
```

```
    5     3
    5     4
    6     1
    6     2
    6     3
    6     4
    7     1
    7     2
    7     3
    7     4
    8     1      40-47
    8     2
    8     3
    8     4
<...Output Truncated...>
Router#
```

Mapping DSCP Values to Standard Transmit-Queue Thresholds

To map DSCP values to standard transmit-queue thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface tengigabitethernet`** *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`wrr-queue dscp-map`** *`transmit_queue_# threshold_# dscp1`* `[`*`dscp2`* `[`*`dscp3`* `[`*`dscp4`* `[`*`dscp5`* `[`*`dscp6`* `[`*`dscp7`* `[`*`dscp8`*`]]]]]]]`	Maps DSCP values to a standard transmit-queue threshold.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show queueing interface tengigabitethernet`** *`slot/port`*	Verifies the configuration.

When mapping DSCP values, note the following information:

- You can enter up to 8 DSCP values that map to a queue and threshold.
- You can enter multiple commands to map additional DSCP values to the queue and threshold.
- You must enter a separate command for each queue and threshold.

This example shows how to map the DSCP values 0 and 1 to standard transmit queue 1/threshold 1 for 10-Gigabit Ethernet port 6/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 6/1
Router(config-if)# wrr-queue dscp-map 1 1 0 1
Router(config-if)# end
Router#
```

Note The eighth queue is the strict priority queue in the output of the **show queueing interface** command.

This example shows how to verify the configuration:

```
Router# show queueing interface tengigabitethernet 6/1 | begin queue thresh dscp-map
queue thresh dscp-map
---------------------------------------
1     1      0 1 2 3 4 5 6 7 8 9 11 13 15 16 17 19 21 23 25 27 29 31 33 39 41 42 43 44 45
47
```

```
1      2
1      3
1      4
2      1       14
2      2       12
2      3       10
2      4
3      1       22
3      2       20
3      3       18
3      4
4      1       24 30
4      2       28
4      3       26
4      4
5      1       32 34 35 36 37 38
5      2
5      3
5      4
6      1       48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
6      2
6      3
6      4
7      1
7      2
7      3
7      4
8      1       40 46
<...Output Truncated...>
Router#
```

Mapping DSCP Values to the Transmit Strict-Priority Queue

To map DSCP values to the transmit strict-priority queue, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface tengigabitethernet** *slot/port*	Selects the interface to configure.
Step 2	Router(config-if)# **priority-queue dscp-map** *queue_# dscp1* [*dscp2* [*dscp3* [*dscp4* [*dscp5* [*dscp6* [*dscp7* [*dscp8*]]]]]]]	Maps DSCP values to the transmit strict-priority queue. You can enter multiple **priority-queue dscp-map** commands to map more than 8 DSCP values to the strict-priority queue.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show queueing interface tengigabitethernet** *slot/port*	Verifies the configuration.

When mapping DSCP values to the strict-priority queue, note the following information:

- The queue number is always 1.
- You can enter up to 8 DSCP values to map to the queue.
- You can enter multiple commands to map additional DSCP values to the queue.

This example shows how to map DSCP value 7 to the strict-priority queue on 10 Gigabit Ethernet port 6/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router(config)# interface tengigabitethernet 6/1
Router(config-if)# priority-queue dscp-map 1 7
Router(config-if)# end
Router#
```

Note The strict priority queue is queue 8 in the output of the **show queueing interface** command.

This example shows how to verify the configuration:

```
 Router# show queueing interface tengigabitethernet 6/1 | begin queue thresh dscp-map
    queue thresh dscp-map
    ---------------------------------------
<...Output Truncated...>
    8      1       7 40 46
<...Output Truncated...>
Router#
```

Configuring CoS-Based Queue Mapping

These sections describe how to configure CoS-based queue mapping:

- Mapping CoS Values to Standard Receive-Queue Thresholds, page 43-106
- Mapping CoS Values to Standard Transmit-Queue Thresholds, page 43-107
- Mapping CoS Values to Strict-Priority Queues, page 43-107
- Mapping CoS Values to Tail-Drop Thresholds on 1q4t/2q2t LAN Ports, page 43-108

Mapping CoS Values to Standard Receive-Queue Thresholds

To map CoS values to the standard receive-queue thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **interface** *type*[1] *slot/port*	Selects the interface to configure.
Step 2	`Router(config-if)#` **rcv-queue cos-map** *queue_# threshold_# cos1* [*cos2* [*cos3* [*cos4* [*cos5* [*cos6* [*cos7* [*cos8*]]]]]]]	Maps CoS values to the standard receive queue thresholds.
Step 3	`Router(config-if)#` **end**	Exits configuration mode.
Step 4	`Router#` **show queueing interface** *type*[1] *slot/port*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to map the CoS values 0 and 1 to threshold 1 in the standard receive queue for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# rcv-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1
<...Output Truncated...>
```

```
    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    1     3      4 5
    1     4      6 7
<...Output Truncated...>
Router#
```

Mapping CoS Values to Standard Transmit-Queue Thresholds

To map CoS values to standard transmit-queue thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **interface** *type*[1] *slot/port*	Selects the interface to configure.
Step 2	`Router(config-if)#` **wrr-queue cos-map** *transmit_queue_# threshold_# cos1* [*cos2* [*cos3* [*cos4* [*cos5* [*cos6* [*cos7* [*cos8*]]]]]]]	Maps CoS values to a standard transmit-queue threshold.
Step 3	`Router(config-if)#` **end**	Exits configuration mode.
Step 4	`Router#` **show queueing interface** *type*[1] *slot/port*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to map the CoS values 0 and 1 to standard transmit queue 1/threshold 1 for Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# wrr-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 5/36 | begin queue thresh cos-map
    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    2     1      4 5
    2     2      6 7
<...Output Truncated...>
Router#
```

Mapping CoS Values to Strict-Priority Queues

To map CoS values to the receive and transmit strict-priority queues, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **interface** *type*[1] *slot/port*	Selects the interface to configure.
Step 2	`Router(config-if)#` **priority-queue cos-map** *queue_# cos1* [*cos2* [*cos3* [*cos4* [*cos5* [*cos6* [*cos7* [*cos8*]]]]]]]	Maps CoS values to the receive and transmit strict-priority queues.

	Command	Purpose
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show queueing interface type[1] slot/port`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When mapping CoS values to the strict-priority queues, note the following information:

- The queue number is always 1.
- You can enter up to 8 CoS values to map to the queue.
- When used, the **priority-queue cos-map** command changes both ingress and egress priority queue CoS mapping.

This example shows how to map CoS value 7 to the strict-priority queues on Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# priority-queue cos-map 1 7
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1
<...Output Truncated...>
Transmit queues [type = 1p2q2t]:
<...Output Truncated...>
    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    2     1      4
    2     2      6
    3     1      5 7

  Receive queues [type = 1p1q4t]:
<...Output Truncated...>
    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    1     3      4 6
    1     4      7
    2     1      5
<...Output Truncated...>
Router#
```

Mapping CoS Values to Tail-Drop Thresholds on 1q4t/2q2t LAN Ports

Note Enter the **show queueing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command to see the queue structure of a port.

On **1q4t/2q2t** LAN ports, the receive- and transmit-queue tail-drop thresholds have this relationship:

- Receive queue 1 (standard) threshold 1 = transmit queue 1 (standard low priority) threshold 1
- Receive queue 1 (standard) threshold 2 = transmit queue 1 (standard low priority) threshold 2

- Receive queue 1 (standard) threshold 3 = transmit queue 2 (standard high priority) threshold 1
- Receive queue 1 (standard) threshold 4 = transmit queue 2 (standard high priority) threshold 2

To map CoS values to tail-drop thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`wrr-queue cos-map transmit_queue_# threshold_# cos1`** `[`*`cos2`* `[`*`cos3`* `[`*`cos4`* `[`*`cos5`* `[`*`cos6`* `[`*`cos7`* `[`*`cos8`*`]]]]]]]`	Maps CoS values to a tail-drop threshold.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show queueing interface`** *`type`*[1] *`slot/port`*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When mapping CoS values to a tail-drop threshold, note the following information:

- Use the transmit queue and threshold numbers.
- Queue 1 is the low-priority standard transmit queue.
- Queue 2 is the high-priority standard transmit queue.
- There are two thresholds in each queue.
- Enter up to 8 CoS values to map to the threshold.

This example shows how to map the CoS values 0 and 1 to standard transmit queue 1/threshold 1 for Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# wrr-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 5/36 | begin queue thresh cos-map
    queue thresh cos-map
    ---------------------------------------
    1     1      0 1
    1     2      2 3
    2     1      4 5
    2     2      6 7
<...Output Truncated...>
Router#
```

Allocating Bandwidth Between Standard Transmit Queues

The switch transmits frames from one standard queue at a time using one of these dequeuing algorithms, which use percentages or weights to allocate relative bandwidth to the queues as they are serviced sequentially:

- Shaped round robin (SRR)—SRR allows a queue to use only the allocated bandwidth. Supported as an option on **1p3q8t** ports and on **1p7q4t** ports.
- Deficit weighted round robin (DWRR)—DWRR keeps track of any lower-priority queue under-transmission caused by traffic in a higher-priority queue and compensates in the next round. DWRR is the dequeuing algorithm on **1p3q1t**, **1p2q1t**, **1p3q8t**, **1p7q4t**, and **1p7q8t** ports.

  Note You configure DWRR ports with the same commands that you use on WRR ports.

- Weighted round robin (WRR)—WRR allows a queue to use more than the allocated bandwidth if the other queues are not using any, up to the total bandwidth of the port. WRR is the dequeuing algorithm on all other ports.
- See the "Module to Queue Type Mappings" section on page 43-24 for information about the modules that support these algorithms.

You can enter percentages or weights to allocate bandwidth. The higher the percentage or weight that is assigned to a queue, the more transmit bandwidth is allocated to it. If you enter weights, the ratio of the weights divides the total bandwidth of the queue. For example, for three queues on a Gigabit Ethernet port, weights of 25:25:50 provide this division:

- Queue 1—250 Mbps
- Queue 2—250 Mbps
- Queue 3—500 Mbps

Note The actual bandwidth allocation depends on the granularity that the port hardware applies to the configured percentages or weights.

To allocate bandwidth between standard transmit queues, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`wrr-queue`** `[`**`bandwidth`** `|` **`shape`**`]` **`percent`** *`low_priority_queue_percentage`* `[`*`intermediate_priority_queue_percentages`*`]` *`high_priority_queue_percentage`* Or: `Router(config-if)#` **`wrr-queue`** `[`**`bandwidth`** `|` **`shape`**`]` *`low_priority_queue_weight`* `[`*`intermediate_priority_queue_weights`*`]` *`high_priority_queue_weight`*	Allocates bandwidth between standard transmit queues: • Enter the **bandwidth** keyword to configure DWRR or WRR. • Enter the **shape** keyword to configure SRR. Use of SRR prevents use of the strict priority queue. To configure SRR, any CoS or DSCP values mapped to a strict-priority queue must be remapped to a standard queue (see the "Mapping QoS Labels to Queues and Drop Thresholds" section on page 43-100). • Percentages should add up to 100. You must enter percentages for all the standard transmit queues on the port. • The valid values for weight range from 1 to 255. You must enter weights for all the standard transmit queues on the port.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show queueing interface`** *`type`*[1] *`slot/port`*	Verifies the configuration.

1. *`type`* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to allocate a 3-to-1 bandwidth ratio for Gigabit Ethernet port 1/2:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/2
Router(config-if)# wrr-queue bandwidth 3 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/2 | include bandwidth
WRR bandwidth ratios:    3[queue 1]    1[queue 2]
Router#
```

Setting the Receive-Queue Size Ratio

You can set the size ratio between the standard receive queues on **2q8t**, **8q4t**, and **8q8t** ports and between the strict-priority and standard receive queues on **1p1q0t** or **1p1q8t** ports.

To set the size ratio between the receive queues, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {**fastethernet** | **tengigabitethernet**} *slot/port*	Selects the interface to configure.
Step 2	Router(config-if)# **rcv-queue queue-limit** *low_priority_queue_weight* [*intermediate_priority_queue_weights*] *high_priority_queue_weight* Or: Router(config-if)# **rcv-queue queue-limit** *standard_queue_weight* *strict_priority_queue_weight*	Sets the size ratio between the receive queues.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show queueing interface** {**fastethernet** | **tengigabitethernet**} *slot/port*	Verifies the configuration.

When setting the receive-queue size ratio, note the following information:

- The **rcv-queue queue-limit** command configures ports on a per-ASIC basis.
- Estimate the mix of differing priority traffic on your network (for example, 80 percent standard traffic and 20 percent strict-priority traffic).
- Use the estimated percentages as queue weights.
- Valid values are from 1 to 100 percent, except on **1p1q8t** ports, where valid values for the strict priority queue are from 3 to 100 percent.

This example shows how to set the receive-queue size ratio for Fast Ethernet port 2/2:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 2/2
Router(config-if)# rcv-queue queue-limit 75 15
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 2/2 | include queue-limit
    queue-limit ratios:     75[queue 1]  15[queue 2]
Router#
```

Configuring the Transmit-Queue Size Ratio

To configure the transmit-queue size ratio, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`wrr-queue queue-limit`** *`low_priority_queue_weight`* [*`intermediate_priority_queue_weights`*] *`high_priority_queue_weight`*	Configures the queue size ratio between transmit queues.
Step 3	`Router(config-if)#` **`priority-queue queue-limit`** *`strict_priority_queue_weight`*	Configures the strict priority queue size. **Note** Not supported on all switching modules.
Step 4	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 5	`Router#` **`show queueing interface`** *`type`*[1] *`slot/port`*	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring the transmit-queue size ratio between transmit queues, note the following information:

- The **wrr-queue queue-limit** command is not supported on **1p3q1t** ports.
- For ports that have an egress strict priority queue:
  - You can enter the **priority-queue queue-limit** interface command to set the size of the egress strict priority queue on these switching modules:

    —WS-X6502-10GE (**1p2q1t**)
    —WS-X6148A-GE-TX (**1p3q8t**)
    —WS-X6148-RJ-45 (**1p3q8t**)
    —WS-X6148-FE-SFP (**1p3q8t**)
    —WS-X6748-SFP (**1p3q8t**)
    —WS-X6724-SFP (**1p3q8t**)
    —WS-X6748-GE-TX (**1p3q8t**)
    —WS-X6704-10GE (**1p7q8t**)
    —WS-SUP32-10GE-3B (**1p3q8t**)
    —WS-SUP32-GE-3B (**1p3q8t**)
    —WS-X6708-10GE, WS-X6716-10GE, WS-X6716-10T, and Supervisor Engine 720-10GE (**1p7q4t**)
- Estimate the mix of low priority-to-high priority traffic on your network (for example, 80 percent low-priority traffic and 20 percent high-priority traffic).
- Use the estimated percentages as queue weights.
- You must enter weights for all the standard transmit queues on the interface (2, 3, or 7 weights).
- Valid values are from 1 to 100 percent, except on **1p2q1t** egress LAN ports, where valid values for the high priority queue are from 5 to 100 percent.

This example shows how to set the transmit-queue size ratio for Gigabit Ethernet port 1/2:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/2
Router(config-if)# wrr-queue queue-limit 75 15
Router(config-if)# end
```

```
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/2 | include queue-limit
    queue-limit ratios:     75[queue 1]  25[queue 2]
Router#
```

Common QoS Scenarios

This section provides sample configurations for some common QoS scenarios. If you already know how to configure PFC QoS for your network or if you need specific configuration information, see the other sections of this chapter.

The scenarios in this section are based on a sample network that is described in the "Sample Network Design Overview" section on page 43-114. This section uses this sample network to describe some regularly used QoS configurations.

These sections describe some common QoS scenarios:

Sample Network Design Overview

This sample network is based on a traditional campus network architecture that uses Catalyst 6500 series switches in the access, distribution, and core layers. The access layer provides 10/100 Ethernet service to desktop users. The network has Gigabit Ethernet links from the access layer to the distribution layer and Gigabit or 10-Gigabit Ethernet links from the distribution layer to the core layer.

This is the basic port configuration:

Access Layer

```
switchport mode access
switchport access vlan 10
switchport voice vlan 110
```

Distribution and Core Interswitch Links

```
switchport mode trunk
```

These are the three traffic classes in the sample network:

- Voice
- High-priority application traffic
- Best-effort traffic

The QoS configuration described in this section identifies and prioritizes each of these traffic classes.

Note If your network requires more service levels, PFC QoS supports up to 64 traffic classes.

These QoS scenarios describe the following three fundamental QoS configurations, which are often a general part of QoS deployment:

- Classifying traffic from PCs and IP phones in the access layer
- Accepting the traffic priority value on interswitch links between layers
- Prioritizing traffic on interswitch links between layers

These QoS scenarios assume that the network carries only IP traffic and use the IP DSCP values to assign traffic priority. These QoS scenarios do not directly use IP type of service (ToS) or Ethernet 802.1p class of service (CoS).

IP packets can carry a priority value, which can be set at various points within the network topology. Best-practice design recommendations are to classify and mark traffic as close to the source of the traffic as possible. If traffic priorities are set correctly at the edge, then intermediate hops do not have to perform detailed traffic identification. Instead, they can administer QoS policies based on these previously set priority values. This approach simplifies policy administration.

Note

- You should develop a QoS deployment strategy for assigning packet priorities to your particular network traffic types and applications. For more information on QoS guidelines, see RFC 2597 and RFC 2598 as well as the various QoS design guides published by Cisco Systems, Inc.
- Do not enable PFC QoS globally and leave all other PFC QoS configuration at default values. When you enable PFC QoS globally, it uses its default values. These are two problems that exist with the PFC QoS default configuration:
  - With PFC QoS globally enabled, the default trust state of the Ethernet ports in the system is untrusted. The untrusted port state sets the QoS priority of all traffic flowing through the switch to the port CoS value (zero by default): all traffic will be zero-priority traffic.
  - With PFC QoS globally enabled, the port buffers are allocated into CoS-based queues and only part of the buffer is available for zero-priority traffic: zero-priority traffic has less buffer available than when PFC QoS is disabled.

  These problems with the PFC QoS default configuration can have a negative effect on network performance.

Classifying Traffic from PCs and IP Phones in the Access Layer

The access layer switches have a PC daisy-chained to an IP phone on a 100 Mbps link. This section describes how to classify voice traffic from the phone and data traffic from the PC so that they have different priorities.

This is the QoS classification scheme for the traffic arriving on an access layer port:

- Voice traffic: DSCP 46 (highest priority)
- Voice signaling traffic: DSCP 24 (medium priority)
- PC SAP traffic: DSCP 25 (medium priority)
- All other PC traffic: DSCP 0 (best effort)

This classification strategy provides a way to support three different classes of service on the network:

- High priority for voice traffic
- Medium priority for voice signaling and important application traffic
- Low priority for the remaining traffic

You can alter this model to fit other network environments.

PFC QoS can trust received priorities or assign new priorities by applying a QoS policy to the traffic. You configure a QoS policy using the Modular QoS CLI (MQC). In the access switches, the traffic is identified using ACLs, which differentiate the various traffic types entering the port. Once identified, a QoS policy marks the traffic with the appropriate DSCP value. These assigned DSCP values will be trusted when the traffic enters the distribution and core switches.

The port on the access switch where the phone and PC are attached has been configured for a voice VLAN (VLAN 110), which is used to separate the phone traffic (subnet 10.1.110.0/24) from the PC traffic (10.1.10.0/24). The voice VLAN subnet uniquely identifies the voice traffic. The UDP and TCP port numbers identify the different applications.

This is the access port access control list (ACL) configuration:

Identify the Voice Traffic from an IP Phone (VVLAN)

```
ip access-list extended CLASSIFY-VOICE
    permit udp 10.1.110.0 0.0.0.255 any range 16384 32767
```

Identify the Voice Signaling Traffic from an IP Phone (VVLAN)

```
ip access-list extended CLASSIFY-VOICE-SIGNAL
    permit udp 10.1.110.0 0.0.0.255 any range 2000 2002
```

Identify the SAP Traffic from the PC (DVLAN)

```
ip access-list extended CLASSIFY-PC-SAP
    permit tcp 10.1.10.0 0.0.0.255 any range 3200 3203
    permit tcp 10.1.10.0 0.0.0.255 any eq 3600 any

ip access-list extended CLASSIFY-OTHER
    permit ip any any
```

The next step in configuring the QoS policy is to define the class maps. These class maps associate the identifying ACLs with the QoS actions that you want to perform (marking, in this case). This is the syntax for the class maps:

```
class-map match-all CLASSIFY-VOICE
    match access-group name CLASSIFY-VOICE
class-map match-all CLASSIFY-VOICE-SIGNAL
    match access-group name CLASSIFY-VOICE-SIGNAL
class-map match-all CLASSIFY-PC-SAP
    match access-group name CLASSIFY-PC-SAP
class-map match-all CLASSIFY-OTHER
    match access-group name CLASSIFY-OTHER
```

After you create the class maps, create a policy map that defines the action of the QoS policy so that it sets a particular DSCP value for each traffic type or traffic class. This example creates one policy map (called IPPHONE-PC), and all the class maps are included in that single policy map, with an action defined in each class map. This is the syntax for the policy map and class maps:

```
policy-map IPPHONE-PC
   class CLASSIFY-VOICE
    set dscp ef
   class CLASSIFY-VOICE-SIGNAL
```

```
  set dscp cs3
 class CLASSIFY-PC-SAP
  set dscp 25
 class CLASSIFY-OTHER
  set dscp 0
```

At this point, the QoS policy defined in the policy map still has not taken effect. After you configure a policy map, you must apply it to an interface for it to affect traffic. You use the **service-policy** command to apply the policy map. Remember that an input service policy can be applied to either a port or to VLAN interfaces, but that an output service policy can only be applied to VLAN interfaces (only the PFC3 supports output policies). In this example, you apply the policy as an input service-policy to each interface that has a PC and IP phone attached. This example uses port-based QoS, which is the default for Ethernet ports.

```
interface FastEthernet5/1
 service-policy input IPPHONE-PC
```

A QoS policy now has been successfully configured to classify the traffic coming in from both an IP phone and a PC.

To ensure that the policy maps are configured properly, enter this command:

```
Router# show policy-map interface fastethernet 5/1
 FastEthernet5/1

  Service-policy input:IPPHONE-PC

    class-map:CLASSIFY-VOICE (match-all)
      Match:access-group name CLASSIFY-VOICE
      set dscp 46:

    class-map:CLASSIFY-PC-SAP (match-all)
      Match:access-group name CLASSIFY-PC-SAP
      set dscp 25:

    class-map:CLASSIFY-OTHER (match-all)
      Match:access-group name CLASSIFY-OTHER
      set dscp 0:

    class-map:CLASSIFY-VOICE-SIGNAL (match-all)
      Match:access-group name CLASSIFY-VOICE-SIGNAL
      set dscp 24:
```

To ensure that the port is using the correct QoS mode, enter this command:

```
Router# show queueing interface gigabitethernet 5/1 | include Port QoS
Port QoS is enabled
```

To ensure that the class map configuration is correct, enter this command:

```
Router# show class-map
 Class Map match-all CLASSIFY-OTHER (id 1)
   Match access-group name CLASSIFY-OTHER

 Class Map match-any class-default (id 0)
   Match any

 Class Map match-all CLASSIFY-PC-SAP (id 2)
   Match access-group name CLASSIFY-PC-SAP

 Class Map match-all CLASSIFY-VOICE-SIGNAL (id 4)
   Match access-group name CLASSIFY-VOICE-SIGNAL

 Class Map match-all CLASSIFY-VOICE (id 5)
   Match access-group name CLASSIFY-VOICE
```

To monitor the byte statistics for each traffic class, enter this command:

```
Router# show mls qos ip gig 5/1
   [In] Policy map is IPPHONE-PC   [Out] Default.
 QoS Summary [IP]:     (* - shared aggregates, Mod - switch module)

           Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                         Id         Id

-------------------------------------------------------------------------------------
          Gi5/1  5  In CLASSIFY-V   46    1     No  0              0  0
          Gi5/1  5  In CLASSIFY-V   24    2     No  0              0  0
          Gi5/1  5  In CLASSIFY-O    0    3     No  0              0  0
          Gi5/1  5  In CLASSIFY-P   25    4     No  0              0  0
Router#
```

Accepting the Traffic Priority Value on Interswitch Links

The previous section described how to configure the marking operation. This section describes how the upstream devices will use the packet marking.

You must decide whether the incoming traffic priority should be honored or not. To implement the decision, you configure the trust state of the port. When traffic arrives on a port that is set not to trust incoming traffic priority settings, the priority setting of the incoming traffic is rewritten to the lowest priority (zero). Traffic that arrives on an interface that is set to trust incoming traffic priority settings retains its priority setting.

Examples of ports on which it might be valid to trust incoming priority settings are ports that are connected to IP phones and other IP voice devices, video devices, or any device that you trust to send frames with a valid predetermined priority. If you know that appropriate marking is completed when traffic first enters the network, you may also want to set uplink interfaces to trust the incoming priority settings.

Configure ports that are connected to workstations or any devices that do not send all traffic with a predetermined valid priority as untrusted (the default).

In the previous example, you configured QoS to properly mark the voice, SAP, and other best effort traffic at the access layer. This example configures QoS to honor those values as the traffic passes through other network devices by configuring the interswitch links to trust the packet DSCP values.

The previous example had several different traffic classes entering a port and selectively applied different QoS policies to the different traffic types. The configuration was done with the MQC QoS policy syntax, which allows you to apply different marking or trust actions to the different traffic classes arriving on a port.

If you know that all traffic entering a particular port can be trusted (as is the case on access-distribution or distribution-core uplink ports), you can use the port trust configuration. Using port trust does not provide any support for different traffic types entering a port, but it is a much simpler configuration option. This is the command syntax for port trust:

```
interface gigabitethernet 5/1
 mls qos trust dscp
```

With ports configured to trust received DSCP, the DSCP value for the traffic leaving the switch will be the same as the DSCP value for the traffic entering the trusted ports. After you have configured the trust state, you can use the following commands to verify that the setting has taken effect:

```
Router# show queueing interface gigabitethernet 5/1 | include Trust
Trust state:trust DSCP
```

Prioritizing Traffic on Interswitch Links

This section describes how the switches operate using trusted values.

One of the most fundamental principles of QoS is to protect high-priority traffic in the case of oversubscription. The marking and trusting actions described in the "Classifying Traffic from PCs and IP Phones in the Access Layer" section on page 43-115 and the "Accepting the Traffic Priority Value on Interswitch Links" section on page 43-118 prepare the traffic to handle oversubscription, but they do not provide different levels of service. To achieve differing levels of service, the networking device must have an advanced scheduling algorithm to prioritize traffic as it sends traffic from a particular interface. This scheduling function is responsible for transmitting the high-priority traffic with greater frequency than the low-priority traffic. The net effect is a differentiated service for the various traffic classes.

These two concepts are fundamental to the provision of differentiated service for various traffic classes:

- Assigning the traffic to a particular queue
- Setting the queue scheduling algorithm

Once QoS has been enabled, default values are applied for both of these features. For many networks, these default values are sufficient to differentiate the network traffic. For other networks, theses values might need to be adjusted to produce the desired result. Only in rare cases should there be a need for significant changes from the default settings for these features.

The Ethernet ports support a variety of queue structures, ranging from a single queue up to an eight-queue architecture. You can compare the queue structure to a group of traffic lanes used to service different traffic types. For example, the police get prioritized treatment when driving down the freeway so that they can get to accidents or crime scenes quickly. In an analogous way, the voice traffic on an IP network requires the same prioritized treatment. The switch uses the queue structure to provide these lanes of differentiated service.

The exact queue type is specific to the Ethernet module that you are working with. This example uses a module that has four transmit queues, described as 1p3q8t, which indicates:

- One strict priority queue (1p)
- Three regular queues supporting Weighted-Round Robin scheduling (3q), each with eight WRED thresholds (8t, not discussed here)

The Ethernet ports also have input queue structures, but these are used less often, and because there probably will not be congestion within the switch fabric, this example does not include them.

To assign traffic to these queues, you need to configure a mapping of priority values to queues. QoS uses the DSCP-to-CoS map to map the 64 possible outgoing DSCP values to the eight possible 802.1p values, and then uses a CoS-to-queue map to map the CoS values to queues.

When the packet enters the switch, QoS is either configured to classify and mark the packet with a configured DSCP value (as in the "Classifying Traffic from PCs and IP Phones in the Access Layer" section on page 43-115) or to trust the packet's incoming DSCP value (as in the "Accepting the Traffic Priority Value on Interswitch Links" section on page 43-118). These options determine the packet's priority as it leaves the switch.

This example shows how to display the DSCP-to-CoS mapping:

```
Router# show mls qos maps dscp-cos
   Dscp-cos map:                                   (dscp= d1d2)
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     -------------------------------------
      0 :    00 00 00 00 00 00 00 00 01 01
      1 :    01 01 01 01 01 01 02 02 02 02
      2 :    02 02 02 02 03 03 03 03 03 03
      3 :    03 03 04 04 04 04 04 04 04 04
```

```
      4 :     05 05 05 05 05 05 05 05 06 06
      5 :     06 06 06 06 06 06 07 07 07 07
      6 :     07 07 07 07
Router#
```

The example marked the voice traffic with a DSCP value of 46. You can use the command output to translate DSCP 46 to CoS 5. You can use the command output to translate the other marked DSCP values to CoS values.

You can make changes to this mapping table to suit the needs of your particular network. Only minor changes are typically necessary; this example does not make any changes.

For queueing purposes, the configuration derives a CoS value from the outgoing DSCP value. This CoS value is used for queue assignment even if the outgoing port is an access port (that is, not a trunk port). However, there will be no 802.1q VLAN tag transmitted on the network if the outgoing port is an access port.

Map each derived CoS value to the queue structure. This example shows how to display the default CoS-to-queue mapping, which shows the queue to which each of the eight CoS values is mapped:

```
Router# show queueing interface gigabitethernet 5/1 | begin cos-map
    queue thresh cos-map
    ---------------------------------------
    1     1      0
    1     2      1
    1     3
    1     4
    1     5
    1     6
    1     7
    1     8
    2     1      2
    2     2      3 4
    2     3
    2     4
    2     5
    2     6
    2     7
    2     8
    3     1      6 7
    3     2
    3     3
    3     4
    3     5
    3     6
    3     7
    3     8
    4     1      5

<output truncated>
```

You want voice traffic mapped to the strict priority queue, which is queue 4 on 1p3q8t ports. The example maps the DSCP 46 voice traffic to CoS 5, which means that you want the CoS 5 traffic to be mapped to the strict priority queue, and you can use the output of the **show queueing interface** command to verify that CoS 5 traffic is mapped to the strict priority queue.

This is a list of the queue mappings for all of the traffic types in this example:

Traffic Type	DSCP	CoS (from DSCP-to-CoS map)	Output Queue
Voice	46	5	Strict Priority
Voice signaling	24	3	Queue 2, Threshold 2
PC SAP	25	3	Queue 2, Threshold 2
Other traffic	0	0	Queue 1, Threshold 1

Traffic that is transmitted through the switch is directed to these different queues (or “traffic lanes”) based on priority. Because there are more CoS values (zero through seven) than egress queues (three per interface in this example), there are drop thresholds in each standard (that is, nonstrict priority) queue. When more than one CoS value is assigned to a given queue, different drop thresholds can be assigned to these CoS values to distinguish between the different priorities. The thresholds specify the maximum percentage of the queue that traffic with a given CoS value can use before additional traffic with that CoS value is dropped. The example only uses three QoS values (high, medium, and low), so you can assign each CoS value to a separate queue and use the default 100-percent drop thresholds.

You can change the DSCP-to-CoS and CoS-to-queue mapping to suit the needs of your particular network. Only minor changes are typically necessary, and this example includes no changes. If your network requires different mapping, see the “Mapping CoS Values to Standard Transmit-Queue Thresholds” section on page 43-107.

Now you understand how traffic is assigned to the available queues on the output ports of the switch. The next concept to understand is how the queue weights operate, which is called the queue scheduling algorithm.

The scheduling algorithms used on the Ethernet ports are strict priority (SP) queueing and weighted round robin (WRR) queueing. These algorithms determine the order, or the priority, that the various queues on a port are serviced.

The strict priority queueing algorithm is simple. One queue has absolute priority over all of the other queues. Whenever there is a packet in the SP queue, the scheduler will service that queue, which ensures the highest possibility of transmitting the packet and the lowest possible latency in transmission even in periods of congestion. The strict priority queue is ideal for voice traffic because voice traffic requires the highest priority and lowest latency on a network, and it also is a relatively low-bandwidth traffic type, which means that voice traffic is not likely to consume all available bandwidth on a port. You would not want to assign a high-bandwidth application (for example, FTP) to the strict priority queue because the FTP traffic could consume all of the bandwidth available to the port, starving the other traffic classes.

The WRR algorithm uses relative weights that are assigned to the WRR queues. If there are three queues and their weights are 100:150:200 (which are the default settings), then queue 1 gets only 22 percent of the available bandwidth, queue 2 gets 33 percent, and queue 3 gets 45 percent. With WRR, none of the queues are restricted to these percentages. If queue 2 and queue 3 do not have any traffic, queue 1 can use all available bandwidth.

In this example, queue 1 has a lower priority than queue 2, and queue 2 has a lower priority than queue 3. The low-priority traffic (phone-other and PC-other) maps to queue 1, and the medium-priority traffic (voice-signaling and PC-SAP) maps to queue 2.

The strict-priority queue does not require any configuration after traffic has been mapped to it. The WRR queues have a default bandwidth allocation that might be sufficient for your network; if it is not, then you can change the relative weights to suit your traffic types (see the “Allocating Bandwidth Between Standard Transmit Queues” section on page 43-110).

The best way to verify that the switch is handling oversubscription is to ensure that there is minimal packet drop. Use the **show queueing interface** command to determine where that packet loss is happening. This command displays the number of dropped packets for each queue.

Using Policers to Limit the Amount of Traffic from a PC

Rate limiting is a useful way of ensuring that a particular device or traffic class does not consume more bandwidth than expected. On the Ethernet ports, the supported rate-limiting method is called policing. Policing is implemented in the PFC hardware with no performance impact. A policer operates by allowing the traffic to flow freely as long as the traffic rate remains below the configured transmission rate. Traffic bursts are allowed, provided that they are within the configured burst size. Any traffic that exceeds the configured rate and burst can be either dropped or marked down to a lower priority. The benefit of policing is that it can constrain the amount of bandwidth that a particular application consumes, which helps ensure quality of service on the network, especially during abnormal network conditions such as a virus or worm attack.

This example focuses on a basic per-interface aggregate policer applied to a single interface in the inbound direction, but you can use other policing options to achieve this same result.

The configuration of a policer is similar to the marking example provided in the "Classifying Traffic from PCs and IP Phones in the Access Layer" section on page 43-115 because policing uses the same ACL and MQC syntax. The syntax in that example created a class-map to identify the traffic and then created a policy-map to specify how to mark the traffic.

The policing syntax is similar enough that we can use the marking example ACL and modify the marking example class map by replacing the **set dscp** command with a **police** command. This example reuses the CLASSIFY-OTHER class-map to identify the traffic with a modified IPPHONE-PC policy map to police the matched traffic to a maximum of 50 Mbps, while continuing to mark the traffic that conforms to this rate.

The class maps and the ACL and **class-map** commands that are used to identify the "other" traffic are included below for reference; no changes have been made.

- ACL commands:

```
ip access-list extended CLASSIFY-OTHER
permit ip any any
```

- Class map commands:

```
class-map match-all CLASSIFY-OTHER
match access-group name CLASSIFY-OTHER
```

The difference between this policer configuration and the marking configuration is the policy-map action statements. The marking example uses the **set dscp** command to mark the traffic with a particular DSCP value. This policing example marks the CLASSIFY-OTHER traffic to a DSCP value of zero and polices that traffic to 50 Mbps. To do this, replace the **set dscp** command with a **police** command. The **police** command allows a marking action to take place: it marks all traffic below the 50 Mbps limit to DSCP 0 and drops any traffic above the 50 Mbps threshold.

This is the modified IPPHONE-PC policy map, which includes the **police** command:

```
policy-map IPPHONE-PC
class CLASSIFY-OTHER
police 50000000 1562500 conform-action set-dscp-transmit default exceed-action drop
```

These are the **police** command parameters:

- The 50000000 parameter defines the committed information rate (CIR) for traffic allowed in this traffic class. This example configures the CIR to be 50 Mbps.
- The 1562500 parameter defines the CIR burst size for traffic in this traffic class; this example uses a default maximum burst size. Set the CIR burst size to the maximum TCP window size used on the network.
- The **conform action** keywords define what the policer does with CLASSIFY-OTHER packets transmitted when the traffic level is below the 50-Mbps rate. In this example, **set-dscp-transmit default** applies DSCP 0 to those packets.
- The **exceed action** defines what the policer does with CLASSIFY-OTHER packets transmitted when the traffic level is above the 50 Mbps CIR. In this example, **exceed action drop** drops those packets.

This is a basic example of a single rate per-interface aggregate policer. The PFC3 also support a dual-rate policer for providing both CIR and peak information rate (PIR) granularity.

Attach the policy map to the appropriate interface using the **service-policy input** command:

```
interface FastEthernet5/1
service-policy input IPPHONE-PC
```

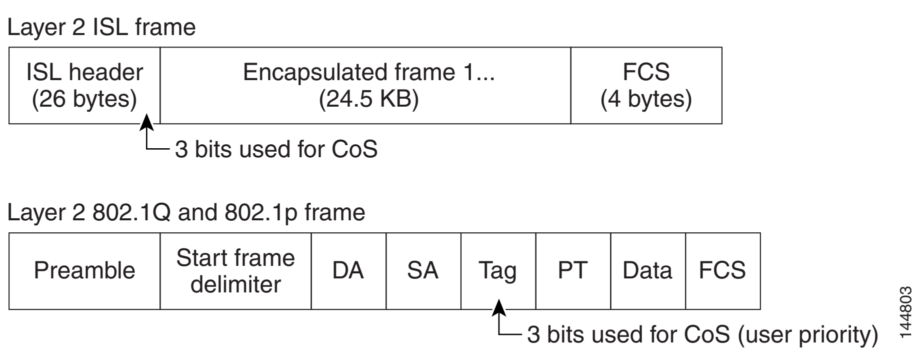
To monitor the policing operation, use these commands:

```
show policy-map interface fastethernet 5/1
show class-map
show mls qos ip fastethernet 5/1
```

PFC QoS Glossary

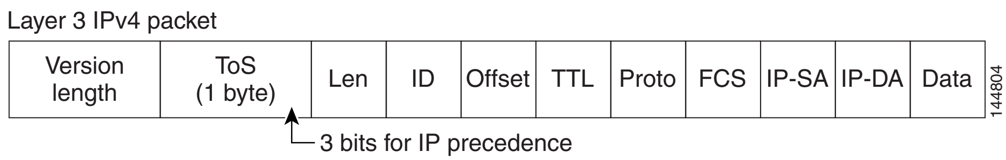
This section defines some of the QoS terminology used in this chapter:

- *Buffers*—A storage area used for handling data in transit. Buffers are used in internetworking to compensate for differences in processing speed between network devices. Bursts of data can be stored in buffers until they can be handled by slower processing devices. Sometimes referred to as a packet buffer.
- *Class of service (CoS)* is a Layer 2 QoS label carried in three bits of either an ISL, 802.1Q, or 802.1p header. CoS values range between zero and seven.

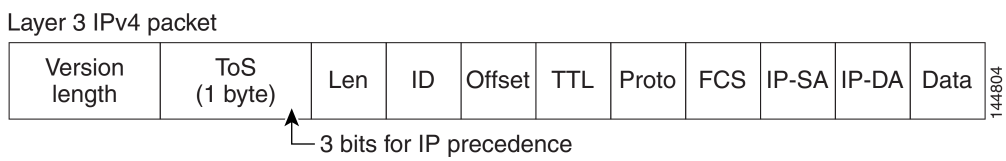


- *Classification* is the process used for selecting traffic to be marked for QoS.

- *Congestion avoidance* is the process by which PFC QoS reserves ingress and egress LAN port capacity for Layer 2 frames with high-priority Layer 2 CoS values. PFC QoS implements congestion avoidance with Layer 2 CoS value-based drop thresholds. A drop threshold is the percentage of queue buffer utilization above which frames with a specified Layer 2 CoS value is dropped, leaving the buffer available for frames with higher-priority Layer 2 CoS values.
- *Differentiated Services Code Point* (DSCP) is a Layer 3 QoS label carried in the six most-significant bits of the ToS byte in the IP header. DSCP ranges between 0 and 63.



- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *IP Precedence* is a Layer 3 QoS label carried in the three most-significant bits of the ToS byte in the IP header. IP precedence ranges between zero and seven.



- *Labels*—See QoS labels.
- *Marking* is the process of setting a Layer 3 DSCP value in a packet; in this publication, the definition of marking is extended to include setting Layer 2 CoS values. Marking changes the value of a label.
- *Packets* carry traffic at Layer 3.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing is done on the PFC and Distributed Forwarding Cards (DFCs). Policing can mark or drop traffic.
- *Queues*—Queues are allocations of buffer space used to temporarily store data on a port.
- *QoS labels*—PFC QoS uses CoS, DSCP, and IP Precedence as QoS labels. QoS labels are prioritization values carried in Layer 3 packets and Layer 2 frames.
- *Scheduling* is the assignment of Layer 2 frames to a queue. PFC QoS assigns frames to a queue based on Layer 2 CoS values.
- *Shaped round robin* (SRR) is a dequeuing algorithm.
- *Threshold*—Percentage of queue capacity above which traffic is dropped.
- *Type of service* (ToS) is a one-byte field that exists in an IP version 4 header that is used to specify the priority value applied to the packet. The ToS field consists of eight bits. The first three bits specify the IP precedence value, which can range from zero to seven, with zero being the lowest priority and seven being the highest priority. The ToS field can also be used to specify a DSCP value. DSCP is defined by the six most significant bits of the ToS. DSCP values can range from 0 to 63.
- *Weight*—Ratio of bandwidth allocated to a queue.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Using AutoQoS

This chapter describes how to use the automatic quality of service (QoS) configuration feature. Release 12.2(33)SXH and later releases support the automatic quality of service (QoS) configuration feature.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

- Understanding AutoQoS, page 44-1
- Using AutoQoS, page 44-3

Understanding AutoQoS

AutoQoS is a macro that applies the recommended Architecture for Voice, Video, and Integrated Data (AVVID) QoS settings to a port. These sections describe how autoQoS works:

- AutoQoS Support for a Cisco IP Phone, page 44-2
- AutoQoS Support for Cisco IP Communicator, page 44-2
- AutoQoS Support for Marked Traffic, page 44-2

AutoQoS Support for a Cisco IP Phone

Cisco IP phones are usually connected directly to ports. Optionally, you can attach a PC to the phone and use the phone as a hop to the switch.

The traffic that comes from the phone can be marked with an 802.1Q or 802.1p tag. The tag contains a VLAN ID and CoS value. When you configure the port to trust the CoS value that comes from the phone, the switch uses the CoS value to prioritize the phone traffic.

There is a three-port switch built into Cisco IP phones that forwards the traffic that comes from the PC, the phone, and the switch port. Cisco IP phones have trust and classification capabilities that you need to configure (see the "Configuring Cisco IP Phone Support" section on page 15-5).

AutoQoS supports Cisco IP phones with the **auto qos voip cisco-phone** interface configuration command. When you enter the **auto qos voip cisco-phone** interface configuration command on a port that is configured to support an IP phone and to which an IP phone is connected, the autoQoS feature does the following:

- If QoS was not already enabled, enables QoS globally.
- If VLAN-based QoS was configured for the port, reverts to the default port-based QoS (done for all ports on switching modules with **1p1q0t/1p3q1t** ports).
- Sets the port trust state to trust CoS.
- Creates and applies a trust-CoS QoS policy to ports on switching modules with non-Gigabit Ethernet **1q4t/2q2t** ports, which do not support port trust.

AutoQoS Support for Cisco IP Communicator

The Cisco IP Communicator program runs on a PC and emulates a Cisco IP phone. The Cisco IP Communicator marks its voice traffic with a DSCP value instead of a CoS value. When you configure the port to trust the DSCP value that comes from the Cisco IP Communicator, the switch uses the DSCP value to prioritize the Cisco IP Communicator traffic.

AutoQoS supports the Cisco IP Communicator program with the **auto qos voip cisco-softphone** interface configuration command. When you enter the **auto qos voip cisco-softphone** interface configuration command on a port that is connected to a device running the Cisco IP Communicator program, the autoQoS feature does the following:

- If QoS was not already enabled, enables QoS globally.
- If VLAN-based QoS was configured for the port, reverts to the default port-based QoS (done for all ports on switching modules with **1p1q0t/1p3q1t** ports).
- If a trust state was configured for the port, reverts to the default untrusted state.
- Creates and applies ingress policers to trust DSCP 46 and remark DSCP 26 packets to DSCP 24. Packets with other DSCP values or out-of-profile packets are remarked with DSCP 0.

AutoQoS Support for Marked Traffic

Ports that connect to the interior of your network might receive traffic that has already been marked with QoS labels that are consistent with your network QoS policies, and which do not need to be changed. You can use the QoS trust feature to process the marked traffic using the received QoS values.

AutoQoS supports marked traffic with the **auto qos voip trust** interface configuration command. When you enter the **auto qos voip trust** interface configuration command, the autoQoS feature does the following:

- If QoS was not already enabled, enables QoS globally.
- If VLAN-based QoS was configured for the port, reverts to the default port-based QoS (done for all ports on switching modules with **1p1q0t/1p3q1t** ports).
- If the port is configured with the **switchport** command, sets the port trust state to trust CoS.
- If the port is not configured with the **switchport** command, sets the port trust state to trust DSCP.
- Creates and applies a trust-CoS or trust-DSCP QoS policy to ports on switching modules with non-Gigabit Ethernet **1q4t/2q2t** ports, which do not support port trust.

Using AutoQoS

These sections describe how to use autoQoS:

AutoQoS Configuration Guidelines and Restrictions

These sections provide the configuration guidelines and restrictions for autoQoS:

- AutoQoS generates commands for the port and adds the generated commands to the running configuration.
- The generated QoS commands are applied as if you were entering them from the CLI. An existing configuration might cause the application of the generated commands to fail or an existing configuration might be overridden by the generated commands. These actions occur without warning. If the generated commands are successfully applied, any configuration that was not overridden remains in the running configuration. Any commands that were overridden might still exist in the startup-config file.
- Some of the generated commands are the type of PFC QoS commands that are applied to all ports controlled by a port ASIC. When one of these commands is applied, PFC QoS displays the messages caused by application of the command to all the ports controlled by the port ASIC. Depending on the module, these commands are applied to as many as 48 ports. See the “Number of port groups” and “Port ranges per port group” listed for each module in the *Release Notes for Cisco IOS Release 12.2SX*.
- You might not be able to configure support for Cisco IP phones and the other autoQoS options on ports that are controlled by the same port ASIC because of conflicting port trust state requirements.
- If application of the generated commands fails, the previous running configuration is restored.
- Enable autoQoS before you configure other QoS commands. If necessary, you can modify the QoS configuration after the autoQoS configuration completes.
- AutoQoS cannot attach a policy map to an interface if there is already a policy map attached.
- Do not modify a policy map or class map that includes AUTOQOS in its name.
- You cannot configure autoQoS on the following:
  - Port-channel interfaces

- VLAN interfaces (also known as switch virtual interfaces or SVIs)
- Tunnel interfaces
- Loopback interfaces
- Subinterfaces on any type of interface

Configuring AutoQoS

AutoQoS generates commands that are appropriate for the QoS port architecture of the port on which you enter an **auto qos voip** command. For each of the different **auto qos voip** commands, autoQoS generates different QoS commands for each of these QoS port architectures:

- 1p1q0t/1p3q1t
- 1p1q4t/1p2q2t
- 1p1q4t/1p3q8t
- 1p1q8t/1p2q1t
- 1q2t/1p2q2t
- 1q2t/1p3q8t
- 1q4t/2q2t
- 1q8t/1p3q8t
- 1q8t/1p7q8t
- 2q8t/1p3q8t
- 8q4t/1p7q4t
- 8q8t/1p7q8t

The procedures in the following sections include the commands that you need to enter to display the generated commands, but the specific commands that autoQoS generates are not listed in this document. These sections describe how to configure autoQoS:

- Configuring AutoQoS Support for a Cisco IP Phone, page 44-4
- Configuring AutoQoS Support for Cisco IP Communicator, page 44-5
- Configuring AutoQoS Support for Marked Traffic, page 44-6

Configuring AutoQoS Support for a Cisco IP Phone

Note Complete the configuration procedures in the "Configuring Cisco IP Phone Support" section on page 15-5 before you configure autoQoS for a Cisco IP phone.

To configure autoQoS for a Cisco IP phone, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface type[1] slot/port`	Selects the interface to configure.

	Command	Purpose
Step 3	`Router(config-if)# auto qos voip cisco-phone`	Configures autoQoS for a Cisco IP phone.
Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Router# show auto qos interface` *type*[1] *slot/port*	Displays the configured autoQoS commands.
Step 6	`Router# show running-config | include mls qos map cos-dscp`	Displays the generated received CoS to internal DSCP map.
Step 7	`Router# show running-config interface` *type*[1] *slot/port*	Displays all of the commands configured on the interface.

1. *type* = **fastethernet** or **gigabitethernet**

When configuring autoQoS for a Cisco IP phone, note the following information:

- To disable autoQoS on an interface, use the **no auto qos voip** interface configuration command.

Note The **no auto qos voip** interface configuration command does not disable QoS globally or delete the received CoS to internal DSCP map created by autoQoS.

- You might see messages that instruct you to configure other ports to trust CoS. You must do so to enable the autoQoS generated commands.

This example shows how to enable autoQoS on Fast Ethernet interface 1/1:

```
Router(config)# interface fastethernet 1/1
Router(config-if)# auto qos voip cisco-phone
```

Configuring AutoQoS Support for Cisco IP Communicator

To configure autoQoS for Cisco IP Communicator, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface` *type*[1] *slot/port*	Selects the interface to configure.
Step 3	`Router(config-if)# auto qos voip cisco-softphone`	Configures autoQoS for Cisco IP Communicator.
Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Router# show auto qos interface` *type*[1] *slot/port*	Displays the configured autoQoS commands.
Step 6	`Router# show policy-map AUTOQOS-CISCO-SOFT-PHONE`	Displays the policy map and policers created by autoQoS.
Step 7	`Router# show class-map AUTOQOS-CISCO-SOFTPHONE-SIGNAL` `Router# show class-map AUTOQOS-CISCO-SOFTPHONE-DATA`	Displays the class maps created by autoQoS.
Step 8	`Router# show running-config | include mls qos map policed-dscp`	Displays the DSCP markdown maps created by autoQoS.
Step 9	`Router# show running-config interface` *type*[1] *slot/port*	Displays all of the commands configured on the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring autoQoS for Cisco IP Communicator, note the following information:

- To disable autoQoS on an interface, use the **no auto qos voip** interface configuration command.

Note The **no auto qos voip** interface configuration command does not disable QoS globally or delete the policy, class, and DSCP markdown maps created by autoQoS.

- You cannot configure support for Cisco IP Communicator on ports that are configured with the **switchport** keyword.
- PFC QoS supports 1023 aggregate policers and each use of the **auto qos voip cisco-softphone** command on a port uses two aggregate policers.

This example shows how to enable autoQoS on Fast Ethernet interface 1/1:

```
Router(config)# interface fastethernet 1/1
Router(config-if)# auto qos voip cisco-softphone
```

Configuring AutoQoS Support for Marked Traffic

To configure autoQoS for marked traffic, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface type[1] slot/port`	Selects the interface to configure.
Step 3	`Router(config-if)# auto qos voip trust`	Configures autoQoS for marked traffic.
Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Router# show auto qos interface type[1] slot/port`	Displays the configured autoQoS commands.
Step 6	`Router# show running-config | include mls qos map cos-dscp`	For ports configured with the **switchport** command, displays the generated received CoS to internal DSCP map.
Step 7	`Router# show running-config interface type[1] slot/port`	Displays all of the commands configured on the interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring autoQoS to trust marked traffic, note the following information:

- To disable autoQoS on an interface, use the **no auto qos voip** interface configuration command.

Note The **no auto qos voip** interface configuration command does not disable QoS globally or delete the received CoS to internal DSCP map created by autoQoS.

- For ports configured with the **switchport** command, you might see messages that instruct you to configure other ports to trust CoS. You must do so to enable the autoQoS generated commands.

This example shows how to enable autoQoS on Fast Ethernet interface 1/1:

```
Router(config)# interface fastethernet 1/1
Router(config-if)# auto qos voip trust
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring MPLS QoS

This chapter describes how to configure Multiprotocol Label Switching (MPLS) quality of service (QoS) in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- MPLS QoS extends to MPLS traffic the PFC QoS features described in Chapter 43, "Configuring PFC QoS."
- This chapter provides supplemental information on MPLS QoS features. Be sure that you understand the PFC QoS features before you read this chapter.
- All policing and marking available for MPLS QoS are managed from the modular QoS command-line interface (CLI). The modular QoS CLI (MQC) is a command-line interface that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach those traffic policies to interfaces. A detailed description of the modular QoS CLI can be found in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/fqos_c.html

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains these sections:

Terminology

This section defines some MPLS QoS terminology:

- *Class of Service* (CoS) refers to three bits in either an Inter-Switch Link (ISL) header or an 802.1Q header that are used to indicate the priority of the Ethernet frame as it passes through a switched network. The CoS bits in the 802.1Q header are commonly referred to as the 802.1p bits. To maintain QoS when a packet traverses both Layer 2 and Layer 3 domains, the type of service (ToS) and CoS values can be mapped to each other.
- *Classification* is the process used for selecting traffic to be marked for QoS.
- *Differentiated Services Code Point* (DSCP) is the first six bits of the ToS byte in the IP header. DSCP is only present in an IP packet.
- *E-LSP* is a label switched path (LSP) on which nodes infer the QoS treatment for MPLS packets exclusively from the experimental (EXP) bits in the MPLS header. Because the QoS treatment is inferred from the EXP (both class and drop precedence), several classes of traffic can be multiplexed onto a single LSP (use the same label). A single LSP can support up to eight classes of traffic because the EXP field is a 3-bit field. The maximum number of classes would be less after reserving some values for control plane traffic or if some of the classes have a drop precedence associated with them.
- *EXP bits* define the QoS treatment (per-hop behavior) that a node should give to a packet. It is the equivalent of the DiffServ Code Point (DSCP) in the IP network. A DSCP defines a class and drop precedence. The EXP bits are generally used to carry all the information encoded in the IP DSCP. In some cases, however, the EXP bits are used exclusively to encode the dropping precedence.
- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *IP precedence* is the three most significant bits of the ToS byte in the IP header.
- *QoS tags* are prioritization values carried in Layer 3 packets and Layer 2 frames. A Layer 2 CoS label can have a value ranging between zero for low priority and seven for high priority. A Layer 3 IP precedence label can have a value ranging between zero for low priority and seven for high priority. IP precedence values are defined by the three most significant bits of the 1-byte ToS byte. A Layer 3 DSCP label can have a value between 0 and 63. DSCP values are defined by the six most significant bits of the 1-byte IP ToS field.
- *LERs* (label edge routers) are devices that impose and dispose of labels upon packets; also referred to as Provider Edge (PE) routers.
- *LSRs* (label switching routers) are devices that forward traffic based upon labels present in a packet; also referred to as Provider (P) routers.
- *Marking* is the process of setting a Layer 3 DSCP value in a packet. Marking is also the process of choosing different values for the MPLS EXP field to mark packets so that they have the priority that they require during periods of congestion.

- *Packets* carry traffic at Layer 3.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.

MPLS QoS Features

QoS enables a network to provide improved service to selected network traffic. This section explains the following MPLS QoS features, which are supported in an MPLS network:

- MPLS Experimental Field, page 45-3
- Trust, page 45-3
- Classification, page 45-3
- Policing and Marking, page 45-4
- Preserving IP ToS, page 45-4
- EXP Mutation, page 45-4
- MPLS DiffServ Tunneling Modes, page 45-4

MPLS Experimental Field

Setting the MPLS experimental (EXP) field value satisfies the requirement of service providers who do not want the value of the IP precedence field modified within IP packets transported through their networks.

By choosing different values for the MPLS EXP field, you can mark packets so that packets have the priority that they require during periods of congestion.

By default, the IP precedence value is copied into the MPLS EXP field during imposition.You can mark the MPLS EXP bits with an MPLS QoS policy.

Trust

For received Layer 3 MPLS packets, the PFC usually trusts the EXP value in the received topmost label. None of the following have any effect on MPLS packets:

- Interface trust state
- Port CoS value
- Policy-map **trust** command

For received Layer 2 MPLS packets, the PFC can either trust the EXP value in the received topmost label or apply port trust or policy trust to the MPLS packets for CoS and egress queueing purposes.

Classification

Classification is the process that selects the traffic to be marked. Classification accomplishes this by partitioning traffic into multiple priority levels, or classes of service. Traffic classification is the primary component of class-based QoS provisioning. The PFC makes classification decisions based on the EXP bits in the received topmost label of received MPLS packets (after a policy is installed). See the "Configuring a Class Map to Classify MPLS Packets" section on page 45-20 for information.

Policing and Marking

Policing causes traffic that exceeds the configured rate to be discarded or marked down to a higher drop precedence. Marking is a way to identify packet flows to differentiate them. Packet marking allows you to partition your network into multiple priority levels or classes of service.

The MPLS QoS policing and marking features that you can implement depend on the received traffic type and the forwarding operation applied to the traffic. See "Configuring a Policy Map" section on page 45-22 for information.

Preserving IP ToS

The PFC automatically preserves the IP ToS during all MPLS operations including imposition, swapping, and disposition.You do not need to enter a command to save the IP ToS.

EXP Mutation

You can configure a named egress EXP mutation map to mutate the internal DSCP-derived EXP value before it is used as the egress EXP value. You can attach egress EXP mutation maps to these interface types:

- LAN port subinterfaces
- Layer 3 VLAN interfaces
- Layer 3 LAN ports

You cannot attach EXP mutation maps to these interface types:

- Layer 2 LAN ports (ports configured with the **switchport** command)
- FlexWAN ports or subinterfaces

For configuration information, see the"Configuring MPLS QoS Egress EXP Mutation" section on page 45-28.

MPLS DiffServ Tunneling Modes

The PFC uses MPLS DiffServ tunneling modes. Tunneling provides QoS transparency from one edge of a network to the other edge of the network. See the "MPLS DiffServ Tunneling Modes" section on page 45-30 for information.

MPLS QoS Overview

MPLS QoS enables network administrators to provide differentiated types of service across an MPLS network. Differentiated service satisfies a range of requirements by supplying for each transmitted packet the service specified for that packet by its QoS. Service can be specified in different ways, for example, using the IP precedence bit settings in IP packets.

Specifying the QoS in the IP Precedence Field

When you send IP packets from one site to another, the IP precedence field (the first three bits of the DSCP field in the header of an IP packet) specifies the QoS. Based on the IP precedence marking, the packet is given the treatment configured for that quality of service. If the service provider network is an MPLS network, then the IP precedence bits are copied into the MPLS EXP field at the edge of the network. However, the service provider might want to set QoS for an MPLS packet to a different value determined by the service offering.

In that case, the service provider can set the MPLS EXP field. The IP header remains available for the customer's use; the QoS of an IP packet is not changed as the packet travels through the MPLS network.

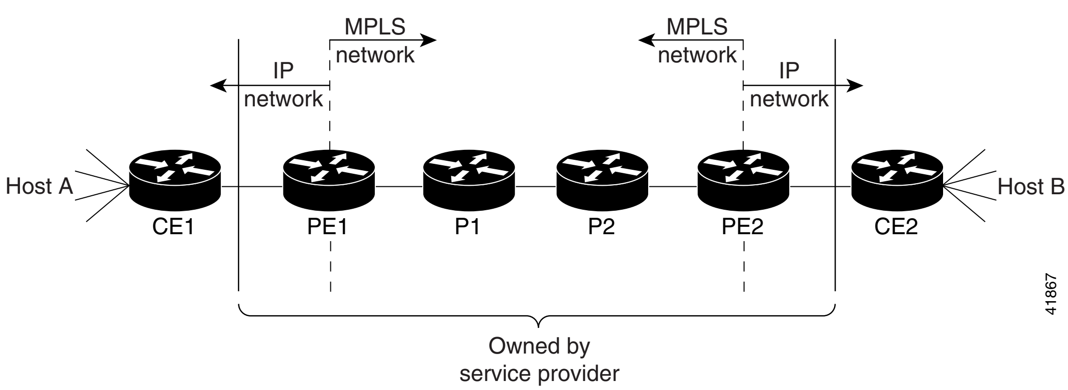
For more information, see the "MPLS DiffServ Tunneling Modes" section on page 45-30.

MPLS QoS

This section describes how MPLS QoS works.

Figure 45-1 shows an MPLS network of a service provider that connects two sites of a customer network.

Figure 45-1 MPLS Network Connecting Two Sites of a Customer's IP Network



The network is bidirectional, but for the purpose of this document the packets move left to right.

In Figure 45-1, the symbols have the following meanings:

- CE1—Customer equipment 1
- PE1—Service provider ingress label edge router (LER)
- P1—Label switch router (LSR) within the core of the network of the service provider
- P2—LSR within the core of the network of the service provider
- PE2—service provider egress LER
- CE2—Customer equipment 2

Note PE1 and PE2 are at the boundaries between the MPLS network and the IP network.

These sections describe LER and LSR operation in an MPLS network.

- LERs at the Input Edge of an MPLS Network, page 45-6
- LSRs in the Core of an MPLS Network, page 45-6
- LERs at the Output Edge of an MPLS Network, page 45-7

Note The QoS capabilities at the input interface differ depending on whether the input interface is a LAN port or a port adapter on a FlexWAN or Enhanced FlexWAN module. This section is for LAN ports. For information on a FlexWAN or Enhanced FlexWAN module, see the *FlexWAN and Enhanced FlexWAN Installation and Configuration Note*.

LERs at the Input Edge of an MPLS Network

Note Incoming labels are aggregate or nonaggregate. The aggregate label indicates that the arriving MPLS or MPLS VPN packet must be switched through an IP lookup to find the next hop and the outgoing interface. The nonaggregate label indicates that the packet contains the IP next hop information.

This section describes how edge LERs can operate at either the ingress or the egress side of an MPLS network.

At the ingress side of an MPLS network, LERs process packets as follows:

1. Layer 2 or Layer 3 traffic enters the edge of the MPLS network at the edge LER (PE1).
2. The PFC receives the traffic from the input interface and uses the 802.1p bits or the IP ToS bits to determine the EXP bits and to perform any classification, marking, and policing. For classification of incoming IP packets, the input service policy can also use access control lists (ACLs).
3. For each incoming packet, the PFC performs a lookup on the IP address to determine the next-hop router.
4. The appropriate label is pushed (imposition) into the packet, and the EXP value resulting from the QoS decision is copied into the MPLS EXP field in the label header.
5. The PFC forwards the labeled packets to the appropriate output interface for processing.
6. The PFC also forwards the 802.1p bits or the IP ToS bits to the output interface.
7. At the output interface, the labeled packets are differentiated by class for marking or policing. For LAN interfaces, egress classification is still based on IP, not on MPLS.
8. The labeled packets (marked by EXP) are sent to the core MPLS network.

LSRs in the Core of an MPLS Network

This section describes how LSRs used at the core of an MPLS network process packets:

1. Incoming MPLS-labeled packets (and 802.1p bits or IP ToS bits) from an edge LER (or other core device) arrive at the core LSR.
2. The PFC receives the traffic from the input interface and uses the EXP bits to perform classification, marking, and policing.
3. The PFC3 performs a table lookup to determine the next-hop LSR.

4. An appropriate label is placed (swapped) into the packet and the MPLS EXP bits are copied into the label header.
5. The PFC forwards the labeled packets to the appropriate output interface for processing.
6. The PFC also forwards the 802.1p bits or the IP ToS bits to the output interface.
7. The outbound packet is differentiated by the MPLS EXP field for marking or policing.
8. The labeled packets (marked with EXP) are sent to another LSR in the core MPLS network or to an LER at the output edge.

Note Within the service provider network, there is no IP precedence field for the queueing algorithm to use because the packets are MPLS packets. The packets remain MPLS packets until they arrive at PE2, the provider edge router.

LERs at the Output Edge of an MPLS Network

At the egress side of an MPLS network, LERs process packets as follows:

1. MPLS-labeled packets (and 802.1p bits or IP ToS bits) from a core LSR arrive at the egress LER (PE2) from the MPLS network backbone.
2. The PFC pops the MPLS labels (disposition) from the packets. Aggregate labels are classified using the original 802.1p bits or the IP ToS bits. Nonaggregate labels are classified with the EXP value by default.
3. For aggregate labels, the PFC performs a lookup on the IP address to determine the packet's destination; the PFC then forwards the packet to the appropriate output interface for processing. For non-aggregate labels, forwarding is based on the label. By default, non-aggregate labels are popped at the penultimate-hop router (next to last), not the egress PE router.
4. The PFC also forwards the 802.1p bits or the IP ToS bits to the output interface.
5. The packets are differentiated according to the 802.1p bits or the IP ToS bits and treated accordingly.

Note The MPLS EXP bits allow you to specify the QoS for an MPLS packet. The IP precedence and DSCP bits allow you to specify the QoS for an IP packet.

Understanding MPLS QoS

MPLS QoS supports IP QoS. For MPLS packets, the EXP value is mapped into an internal DSCP so that the PFC can apply non-MPLS QoS marking and policing.

For both the ingress and egress policies, MPLS QoS marking and policing decisions are made on a per-interface basis at an ingress PFC. The ingress interfaces are physical ports, subinterfaces, or VLANs.

The QoS policy ACLs are programmed in QoS TCAM separately for ingress and egress lookup. The ternary content addressable memory (TCAM) egress lookup takes place after the IP forwarding table (FIB) and NetFlow lookups are completed.

The results of each QoS TCAM lookup yield an index into RAM that contains policer configuration and policing counters. Additional RAM contains the microflow policer configuration; the microflow policing counters are maintained in the respective NetFlow entries that match the QoS ACL.

The results of ingress and egress aggregate and microflow policing are combined into a final policing decision. The out-of-profile packets can be either dropped or marked down in the DSCP.

This section describes MPLS QoS for the following:

- LERs at the EoMPLS Edge, page 45-8
- LERs at the IP Edge (MPLS, MPLS VPN), page 45-9
- LSRs at the MPLS Core, page 45-13

Note The following sections see QoS features for LAN ports and FlexWAN ports. For details about how the different features work, see the appropriate documentation.

LERs at the EoMPLS Edge

This section summarizes the Ethernet over MPLS (EoMPLS) QoS features that function on the LERs. EoMPLS QoS support is similar to IP-to-MPLS QoS:

- For EoMPLS, if the port is untrusted, the CoS trust state is automatically configured for VC type 4 (VLAN mode), not for VC type 5 (port mode). 802.1q CoS preservation across the tunnel is similar.
- Packets received on tunnel ingress are treated as untrusted for EoMPLS interfaces, except for VC Type 4 where trust CoS is automatically configured on the ingress port and policy marking is not applied.
- If the ingress port is configured as trusted, packets received on an EoMPLS interface are never marked by QoS policy in the original IP packet header (marking by IP policy works on untrusted ports).
- 802.1p CoS is preserved from entrance to exit, if available through the 802.1q header.
- After exiting the tunnel egress, queueing is based on preserved 802.1p CoS if 1p tag has been tunnelled in the EoMPLS header (VC type 4); otherwise, queuing is based on the CoS derived from the QoS decision.

Ethernet to MPLS

For Ethernet to MPLS, the ingress interface, MPLS QoS, and egress interface features are similar to corresponding features for IP to MPLS. For more information, see these sections:

- Classification for IP-to-MPLS, page 45-9
- Classification for IP-to-MPLS Mode MPLS QoS, page 45-10
- Classification at IP-to-MPLS Ingress Port, page 45-10
- Classification at IP-to-MPLS Egress Port, page 45-10

MPLS to Ethernet

For MPLS to Ethernet, the ingress interface, MPLS QoS, and egress interface features are similar to corresponding features for MPLS to IP except for the case of EoMPLS decapsulation where egress IP policy cannot be applied (packets can be classified as MPLS only).

For more information, see these sections:

- Classification for MPLS-to-IP, page 45-11
- Classification for MPLS-to-IP MPLS QoS, page 45-11
- Classification at MPLS-to-IP Ingress Port, page 45-11
- Classification at MPLS-to-IP Egress Port, page 45-12.

LERs at the IP Edge (MPLS, MPLS VPN)

This section provides information about QoS features for LERs at the ingress (CE-to-PE) and egress (PE-to-CE) edges for MPLS and MPLS VPN networks. Both MPLS and MPLS VPN support general MPLS QoS features. See the "MPLS VPN" section on page 45-12 for additional MPLS VPN-specific QoS information.

IP to MPLS

The PFC provides the following MPLS QoS capabilities at the IP-to-MPLS edge:

- Assigning an EXP value based on the **mls qos trust** or **policy-map** command
- Marking an EXP value using a policy
- Policing traffic using a policy

This section provides information about the MPLS QoS classification that the PFC supports at the IP-to-MPLS edge. Additionally, this section provides information about the capabilities provided by the ingress and egress interface modules.

Classification for IP-to-MPLS

The PFC ingress and egress policies for IP traffic classify traffic on the original received IP using **match** commands for IP precedence, IP DSCP, and IP ACLs. Egress policies do not classify traffic on the imposed EXP value nor on a marking done by an ingress policy.

After the PFC applies the port trust and QoS policies, it assigns the internal DSCP. The PFC then assigns the EXP value based on the internal DSCP-to-EXP global map for the labels that it imposes. If more than one label is imposed, the EXP value is the same in each label. The PFC preserves the original IP ToS when the MPLS labels are imposed.

The PFC assigns the egress CoS based on the internal DSCP-to-CoS global map. If the default internal DSCP-to-EXP and the internal DSCP-to-CoS maps are consistent, then the egress CoS has the same value as the imposed EXP.

If the ingress port receives both IP-to-IP and IP-to-MPLS traffic, classification should be used to separate the two types of traffic. For example, if the IP-to-IP and IP-to-MPLS traffic have different destination address ranges, you can classify traffic on the destination address, and then apply IP ToS policies to the IP-to-IP traffic and apply a policy (that marks or sets the EXP value in the imposed MPLS header) to the IP-to-MPLS traffic. See the following two examples:

- A PFC policy to mark IP ToS sets the internal DSCP—If it is applied to all traffic, then for IP-to-IP traffic, the egress port will rewrite the CoS (derived from the internal DSCP) to the IP ToS byte in the egress packet. For IP-to-MPLS traffic, the PFC will map the internal DSCP to the imposed EXP value.
- A PFC policy to mark MPLS EXP sets the internal DSCP—If it is applied to all traffic, then for IP-to-IP traffic, the egress port rewrites the IP ToS according to the ingress IP policy (or trust). The CoS is mapped from the ToS. For IP-to-MPLS traffic, the PFC will map the internal DSCP to the imposed EXP value.

Classification for IP-to-MPLS Mode MPLS QoS

MPLS QoS at the ingress to PE1supports:

- Matching on IP precedence or DSCP values or filtering with an access group
- The **set mpls experimental imposition** and **police** commands

MPLS QoS at the egress of PE1 supports the **match mpls experimental topmost** command.

Classification at IP-to-MPLS Ingress Port

Classification for IP-to-MPLS is the same as for IP-to-IP. LAN port classification is based on the received Layer 2 802.1Q CoS value. FlexWAN interfaces classify based on information in the received Layer 3 IP header.

Classification at IP-to-MPLS Egress Port

LAN port classification is based on the received EXP value and the egress CoS values is mapped from that value.

FlexWAN interfaces classify traffic when you use the **match mpls experimental** command to match on the egress CoS as a proxy for the EXP value. The **match mpls experimental** command does not match on the EXP value in the topmost label.

If the egress port is a trunk, the LAN ports copy the egress CoS into the egress 802.1Q field.

MPLS to IP

MPLS QoS supports these capabilities at the MPLS-to-IP edge:

- Option to propagate EXP value into IP DSCP on exit from an MPLS domain per egress interface
- Option to use IP service policy on the MPLS-to-IP egress interface

This section provides information about the MPLS-to-IP MPLS QoS classification. Additionally, this section provides information about the capabilities provided by the ingress and egress modules.

Classification for MPLS-to-IP

The PFC assigns the internal DSCP (internal priority that the PFC assigns to each frame) based on the QoS result. The QoS result is affected by the following:

- Default trust EXP value
- Label type (per-prefix or aggregate)
- Number of VPNs
- Explicit NULL use
- QoS policy

There are three different classification modes:

- Regular MPLS classification—For nonaggregate labels, in the absence of MPLS recirculation, the PFC classifies the packet based on MPLS EXP ingress or egress policy. The PFC queues the packet based on COS derived from EXP-to-DSCP-to-CoS mapping. The underlying IP DSCP is either preserved after egress decapsulation, or overwritten from the EXP (through the EXP-to-DSCP map).
- IP classification for aggregate label hits in VPN CAM—The PFC does one of the following:
  - Preserves the underlying IP ToS
  - Rewrites the IP ToS by a value derived from the EXP-to-DSCP global map
  - Changes the IP ToS to any value derived from the egress IP policy

  In all cases, egress queueing is based on the final IP ToS from the DSCP-to-CoS map.
- IP classification with aggregate labels not in VPN CAM—After recirculation, the PFC differentiates the MPLS-to-IP packets from the regular IP-to-IP packets based on the ingress reserved VLAN specified in the MPLS decapsulation adjacency. The reserved VLAN is allocated per VRF both for VPN and non-VPN cases. The ingress ToS after recirculation can be either the original IP ToS value, or derived from the original EXP value. The egress IP policy can overwrite this ingress ToS to an arbitrary value.

Note For information about recirculation, see the "Recirculation" section on page 32-4.

For incoming MPLS packets on the PE-to-CE ingress, the PFC supports MPLS classification only. Ingress IP policies are not supported. PE-to-CE traffic from the MPLS core is classified or policed on egress as IP.

Classification for MPLS-to-IP MPLS QoS

MPLS QoS at the ingress to PE2 supports matching on the EXP value and the **police** command.

MPLS QoS at the egress of PE2 supports matching on IP precedence or DSCP values or filtering with an access group and the **police** command.

Classification at MPLS-to-IP Ingress Port

LAN port classification is based on the EXP value. FlexWAN interfaces classify traffic using the **match mpls experimental** command. The **match mpls experimental** command matches on the EXP value in the received topmost label.

Classification at MPLS-to-IP Egress Port

Note The egress classification queuing is different for LAN and WAN ports.

Classification for MPLS-to-IP is the same as it is for IP-to-IP.

The LAN interface classification is based on the egress CoS. The FlexWAN interfaces classify traffic on information in the transmitted IP header.

If the egress port is a trunk, the LAN ports copy the egress CoS into the egress 802.1Q field.

Note For MPLS to IP, egress IP ACL or QoS is not effective on the egress interface if the egress interface has MPLS IP (or tag IP) enabled. The exception is a VPN CAM hit, in which case the packet is classified on egress as IP.

MPLS VPN

The information in this section also applies to an MPLS VPN network.

The following PE MPLS QoS features are supported for MPLS VPN:

- Classification, policing, or marking of CE-to-PE IP traffic through the VPN subinterface
- Per-VPN QoS (per-port, per-VLAN, or per-subinterface)

For customer edge (CE)-to-PE traffic, or for CE-to-PE-to-CE traffic, the subinterface support allows you to apply IP QoS ingress or egress policies to subinterfaces and to physical interfaces. Per-VPN policing is also provided for a specific interface or subinterface associated with a given VPN on the CE side.

In situations when there are multiple interfaces belonging to the same VPN, you can perform per-VPN policing aggregation using the same shared policer in the ingress or egress service policies for all similar interfaces associated with the same PFC.

For aggregate VPN labels, the EXP propagation in recirculation case may not be supported because MPLS adjacency does not know which egress interface the final packet will use.

Note For information on recirculation, see the "Recirculation" section on page 32-4.

The PFC propagates the EXP value if all interfaces in the VPN have EXP propagation enabled.

The following PE MPLS QoS features are supported:

- General MPLS QoS features for IP packets
- Classification, policing, or marking of CE-to-PE IP traffic through the VPN subinterface
- Per-VPN QoS (per-port, per-VLAN, or per-subinterface)

LSRs at the MPLS Core

This section provides information about MPLS QoS features for LSRs at the core (MPLS-to-MPLS) for MPLS and MPLS VPN networks. Ingress features, egress interface, and PFC features for Carrier Supporting Carrier (CsC) QoS features are similar to those used with MPLS to MPLS described in the next section. A difference between CsC and MPLS to MPLS is that with CsC labels can be imposed inside the MPLS domain.

MPLS to MPLS

MPLS QoS at the MPLS core supports the following:

- Per-EXP policing based on a service policy
- Copying the input topmost EXP value into the newly imposed EXP value
- Optional EXP mutation (changing of EXP values on an interface edge between two neighboring MPLS domains) on the egress boundary between MPLS domains
- Microflow policing based on individual label flows for a particular EXP value
- Optional propagation of topmost EXP value into the underlying EXP value when popping the topmost label from a multi-label stack.

The following section provides information about MPLS-to-MPLS MPLS QoS classification. Additionally, the section provides information about the capabilities provided by the ingress and egress modules.

Classification for MPLS-to-MPLS

For received MPLS packets, the PFC ignores the port trust state, the ingress CoS, and any policy-map **trust** commands. Instead, the PFC trusts the EXP value in the topmost label.

Note The MPLS QoS ingress and egress policies for MPLS traffic classify traffic on the EXP value in the received topmost label when you enter the **match mpls experimental** command.

MPLS QoS maps the EXP value to the internal DSCP using the EXP-to-DSCP global map. What the PFC does next depends on whether it is swapping labels, imposing a new label, or popping a label:

- Swapping labels—When swapping labels, the PFC preserves the EXP value in the received topmost label and copies it to the EXP value in the outgoing topmost label. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP global maps are consistent, then the egress CoS is based on the EXP in the outgoing topmost label.

  The PFC can mark down out-of-profile traffic using the **police** command's **exceed** and **violate** actions. It does not mark in-profile traffic, so the **conform** action must be transmitted and the **set** command cannot be used. If the PFC is performing a markdown, it uses the internal DSCP as an index into the internal DSCP markdown map. The PFC maps the result of the internal DSCP markdown to an EXP value using the internal DSCP-to-EXP global map. The PFC rewrites the new EXP value to the topmost outgoing label and does not copy the new EXP value to the other labels in the stack. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, then the egress CoS is based on the EXP value in the topmost outgoing label.

- Imposing an additional label—When imposing a new label onto an existing label stack, the PFC maps the internal DSCP to the EXP value in the imposed label using the internal DSCP-to-EXP map. It then copies the EXP value in the imposed label to the underlying swapped label. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, the egress CoS is based on the EXP value in the imposed label.

  The PFC can mark in-profile and mark down out-of-profile traffic. After it marks the internal DSCP, the PFC uses the internal DSCP-to-EXP global map to map the internal DSCP to the EXP value in the newly imposed label. The PFC then copies the EXP in the imposed label to the underlying swapped label. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. Therefore, the egress CoS is based on the EXP in the imposed label.
- Popping a label—When popping a label from a multi-label stack, the PFC preserves the EXP value in the exposed label. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, then the egress CoS is based on the EXP value in the popped label.
- If EXP propagation is configured for the egress interface, the PFC maps the internal DSCP to the EXP value in the exposed label using the DSCP-to-EXP global map. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, the egress CoS is based on the EXP value in the exposed label.

Classification for MPLS-to-MPLS MPLS QoS

MPLS QoS at the ingress to P1 or P2 supports the following:

- Matching with the **match mpls experimental topmost** command
- The **set mpls experimental imposition**, **police**, and **police** with **set imposition** commands

MPLS QoS at the egress of P1 or P2 supports matching with the **match mpls experimental topmost** command.

Classification at MPLS-to-MPLS Ingress Port

LAN port classification is based on the egress CoS from the PFC. FlexWAN interfaces classify traffic using the **match mpls experimental** command. The **match mpls experimental** command matches on the EXP value in the received topmost label.

Classification at MPLS-to-MPLS Egress Port

LAN port classification is based on the egress CoS value from the PFC. FlexWAN interfaces classify traffic using the **match mpls experimental** command. The **match mpls experimental** command matches on the egress CoS; it does not match on the EXP in the topmost label.

If the egress port is a trunk, the LAN ports copy the egress CoS into the egress 802.1Q field.

MPLS QoS Default Configuration

This section describes the MPLS QoS default configuration. The following global MPLS QoS settings apply:

Feature	Default Value
PFC QoS global enable state	**Note** With PFC QoS disabled and all other PFC QoS parameters at default values, default EXP is mapped from IP precedence. **Note** With PFC QoS enabled and all other PFC QoS parameters at default values, PFC QoS sets Layer 3 DSCP to zero (**untrusted** ports only), Layer 2 CoS to zero, the imposed EXP to zero in all traffic transmitted from LAN ports (default is untrusted). For trust CoS, the default EXP value is mapped from COS; for trust DSCP, the default EXP value is mapped from IP precedence.
PFC QoS port enable state	Enabled when PFC QoS is globally enabled
Port CoS value	0
Microflow policing	Enabled
IntraVLAN microflow policing	Disabled
Port-based or VLAN-based PFC QoS	Port-based
EXP to DSCP map (DSCP set from EXP values)	EXP 0 = DSCP 0 EXP 1 = DSCP 8 EXP 2 = DSCP 16 EXP 3 = DSCP 24 EXP 4 = DSCP 32 EXP 5 = DSCP 40 EXP 6 = DSCP 48 EXP 7 = DSCP 56
IP precedence to DSCP map (DSCP set from IP precedence values)	IP precedence 0 = DSCP 0 IP precedence 1 = DSCP 8 IP precedence 2 = DSCP 16 IP precedence 3 = DSCP 24 IP precedence 4 = DSCP 32 IP precedence 5 = DSCP 40 IP precedence 6 = DSCP 48 IP precedence 7 = DSCP 56
DSCP to EXP map (EXP set from DSCP values)	DSCP 0–7 = EXP 0 DSCP 8–15 = EXP 1 DSCP 16–23 = EXP 2 DSCP 24–31 = EXP 3 DSCP 32–39 = EXP 4 DSCP 40–47 = EXP 5 DSCP 48–55 = EXP 6 DSCP 56–63 = EXP 7

Feature	Default Value
Marked-down DSCP from DSCP map	Marked-down DSCP value equals original DSCP value (no mark down)
EXP mutation map	No mutation map by default
Policers	None
Policy maps	None
MPLS flow mask in NetFlow table	Label + EXP value
MPLS core QoS	There are four possibilities at the MPLS core QoS: • Swapping—Incoming EXP field is copied to outgoing EXP field. • Swapping + imposition—Incoming EXP field is copied to both the swapped EXP field and the imposed EXP field. **Note** If there is a service policy with a set for EXP field, its EXP field will be placed into the imposed label and also into the swapped label. • Disposition of topmost label—Exposed EXP field is preserved. • Disposition of only label—Exposed IP DSCP is preserved.
MPLS to IP edge QoS	Preserve the exposed IP DSCP

MPLS QoS Commands

MPLS QoS supports the following MPLS QoS commands:

- **match mpls experimental topmost**
- **set mpls experimental imposition**
- **police**
- **mls qos map exp-dscp**
- **mls qos map dscp-exp**
- **mls qos map exp-mutation**
- **mls qos exp-mutation**
- **show mls qos mpls**
- **no mls qos mpls trust exp**

Note For information about supported non-MPLS QoS commands, see "Configuring PFC QoS" section on page 43-58.

The following commands are not supported:

- **set qos-group**
- **set discard-class**

MPLS QoS Restrictions and Guidelines

When configuring MPLS QoS, follow these guidelines and restrictions:

- For IP-to-MPLS or EoMPLS imposition when the received packet is an IP packet:
  - When QoS is disabled, the EXP value is based on the received IP ToS.
  - When QoS is queuing only, the EXP value is based on the received IP ToS.
- For EoMPLS imposition when the received packet is a non-IP packet:
  - When QoS is disabled, the EXP value is based on the ingress CoS.
  - When QoS is queuing only, the EXP value is based on the received IP ToS.
- For MPLS-to-MPLS operations:
  - Swapping when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Swapping when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Imposing additional label when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Imposing an additional label when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Popping one label when QoS is disabled, the EXP value is based on the underlying EXP value.
  - Popping one label when QoS is queuing only, the EXP value is based on the underlying EXP value.
- EXP value is irrelevant to MPLS-to-IP disposition.
- The **no mls qos rewrite ip dscp** command is incompatible with MPLS. The default **mls qos rewrite ip dscp** command must remain enabled in order for the PFC to assign the correct EXP value for the labels that it imposes.
- The **no mls qos mpls trust exp** command allows you to treat MPLS packets simiarly to Layer 2 packets for CoS and egress queueing purposes by applying port trust or policy trust instead of the default EXP value.

Configuring MPLS QoS

These sections describe how to configure MPLS QoS:

Enabling QoS Globally

Before you can configure QoS on the PFC, you must enable the QoS functionality globally using the **mls qos** command. This command enables default QoS conditioning of traffic.

When the **mls qos** command is enabled, the PFC assigns a priority value to each frame. This value is the internal DSCP. The internal DSCP is assigned based on the contents of the received frame and the QoS configuration. This value is rewritten to the egress frame's CoS and ToS fields.

To enable QoS globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos`	Enables PFC QoS globally on the switch.
	`Router(config)# no mls qos`	Disables PFC QoS globally on the switch.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos`	Verifies the configuration.

This example shows how to enable QoS globally:

```
Router(config)# mls qos
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos
QoS is enabled globally
  Microflow policing is enabled globally
  QoS ip packet dscp rewrite enabled globally

  Qos trust state is DSCP on the following interfaces:
    Gi4/1 Gi4/1.12

  Qos trust state is IP Precedence on the following interfaces:
    Gi4/2 Gi4/2.42
  Vlan or Portchannel(Multi-Earl) policies supported: Yes
  Egress policies supported: Yes


 ----- Module [5] -----
  QoS global counters:
    Total packets: 5957870
    IP shortcut packets: 0
    Packets dropped by policing: 0
    IP packets with TOS changed by policing: 6
    IP packets with COS changed by policing: 0
    Non-IP packets with COS changed by policing: 3
    MPLS packets with EXP changed by policing: 0
```

Enabling Queueing-Only Mode

To enable queueing-only mode, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos queueing-only`	Enables queueing-only mode.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos`	Verifies the configuration.

When you enable queueing-only mode, the router does the following:

- Disables marking and policing globally
- Configures all ports to trust Layer 2 CoS

Note The switch applies the port CoS value to untagged ingress traffic and to traffic that is received through ports that cannot be configured to trust CoS.

This example shows how to enable queueing-only mode:

```
Router# configure terminal
Router(config)# mls qos queueing-only
Router(config)# end
Router#
```

Restrictions and Usage Guidelines

If QoS is disabled (**no mls qos**) for the PFC, the EXP value is determined as follows:

- For IP-to-MPLS or EoMPLS imposition when the received packet is an IP packet:
  - When QoS is disabled (**no mls qos**), the EXP value is based on the received IP ToS.
  - When QoS is queuing only (**mls qos queueing-only**), the EXP value is based on the received IP ToS.
- For EoMPLS imposition when the received packet is a non-IP packet:
  - When QoS is disabled, the EXP value is based on the ingress CoS.
  - When QoS is queuing only, the EXP value is based on the received IP ToS.
- For MPLS-to-MPLS operations:
  - Swapping when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Swapping when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Imposing an additional label when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Imposing additional label when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Popping one label when QoS is disabled, the EXP value is based on the underlying EXP value.

- Popping one label when QoS is queuing only, the EXP value is based on the underlying EXP value.

- EXP value is irrelevant to MPLS-to-IP disposition.

Configuring a Class Map to Classify MPLS Packets

You can use the **match mpls experimental topmost** command to define traffic classes inside the MPLS domain by packet EXP values. This allows you to define service policies to police the EXP traffic on a per-interface basis by using the **police** command.

To configure a class map, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`class-map`** *`class_name`*	Specifies the class map to which packets will be matched.
Step 2	`Router(config-cmap)#` **`match mpls experimental topmost`** *`value`*	Specifies the packet characteristics that will be matched to the class.
Step 3	`Router(config-cmap)#` **`exit`**	Exits class-map configuration mode.

This example shows that all packets that contain MPLS experimental value 3 are matched by the traffic class named exp3:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# class-map exp3
Router(config-cmap)# match mpls experimental topmost 3
Router(config-cmap)# exit
Router(config)# policy-map exp3
Router(config-pmap)# class exp3
Router(config-pmap-c)# police 1000000 8000000 conform-action transmit exceed-action drop
Router(config-pmap-c)# exit
Router(config-pmap)# end
Router# show class exp3
 Class Map match-all exp3 (id 61)
   Match mpls experimental topmost 3
Router# show policy-map exp3
  Policy Map exp3
    Class exp3
     police cir 1000000 bc 8000000 be 8000000 conform-action transmit exceed-action drop
Router# show running-config interface fastethernet 3/27
Building configuration...

Current configuration : 173 bytes
!
interface FastEthernet3/27
  ip address 47.0.0.1 255.0.0.0
  tag-switching ip
 end

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/27
Router(config-if)# service-policy input exp3
Router(config-if)#
Router#
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router# show running-config interface fastethernet 3/27
Building configuration...

Current configuration : 173 bytes
!
interface FastEthernet3/27
  ip address 47.0.0.1 255.0.0.0
  tag-switching ip
  service-policy input exp3
 end

Router#
1w4d: %SYS-5-CONFIG_I: Configured from console by console
Router# show mls qos mpls
 QoS Summary [MPLS]:      (* - shared aggregates, Mod - switch module)

      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
-------------------------------------------------------------------------------
    Fa3/27  5  In       exp3    0    2   dscp  0             0              0

       All  5   -    Default    0    0*    No  0    3466140423              0
Router# show policy-map interface fastethernet 3/27
 FastEthernet3/27

  Service-policy input: exp3

    class-map: exp3 (match-all)
      Match: mpls experimental topmost 3
      police :
        1000000 bps 8000000 limit 8000000 extended limit
      Earl in slot 5 :
        0 bytes
        5 minute offered rate 0 bps
        aggregate-forwarded 0 bytes action: transmit
        exceeded 0 bytes action: drop
        aggregate-forward 0 bps exceed 0 bps

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/27
Router(config-if)# service-policy output ip2tag
Router(config-if)# end
Router# show mls qos ip
 QoS Summary [IPv4]:      (* - shared aggregates, Mod - switch module)

      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
-------------------------------------------------------------------------------
     Vl300  5  In          x   44    1     No  0             0              0
    Fa3/27  5 Out      iptcp   24    2     --  0             0              0

       All  5   -    Default    0    0*    No  0    3466610741              0
```

Restrictions and Usage Guidelines

The following restrictions and guidelines apply when classifying MPLS packets:

- The **match mpls experimental** command specifies the name of an EXP field value to be used as the match criterion against which packets are checked to determine if they belong to the class specified by the class map.
- To use the **match mpls experimental** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. After you identify the class, you can use the **match mpls experimental** command to configure its match criteria.
- If you specify more than one command in a class map, only the last command entered applies. The last command overrides the previously entered commands.

Configuring the MPLS Packet Trust State on Ingress Ports

You can use the **no mls qos mpls trust exp** command to apply port or policy trust to MPLS packets in the same way that you apply them to Layer 2 packets.

To configure the MPLS packet trust state of an ingress port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`*`type slot/port`*`}	{`**`port-channel`** *`number`*`}}`	Selects the interface to configure.
Step 2	`Router(config-if)#` **`no mls qos mpls trust exp`**	Sets the trust state of an MPLS packet so that all trusted cases (trust cos, trust dscp, trust ip-precedence) are treated as trust-cos.
Step 3	`Router(config-if)#` **`end`**	Exits interface configuration mode.
Step 4	`Router#` **`show mls qos`**	Verifies the configuration.

This example shows how to set the trusted state of MPLS packets to untrusted so that the incoming MPLS packets operate like incoming Layer 2 packets.

```
Router(config)# interface fastethernet 3/27
Router(config-if)# no mls qos mpls trust exp
Router(config-if)#
```

Restrictions and Usage Guidelines

The following restrictions and guidelines apply when using the **no mls qos mpls trust exp** command to configure the MPLS packet trust state on input ports:

- This command affects both Layer 2 and Layer 3 packets; use this command only on interfaces with Layer 2 switched packets.
- The **no mls qos mpls trust exp** command affects ingress marking; it does not affect classification.

Configuring a Policy Map

You can attach only one policy map to an interface. Policy maps can contain one or more policy map classes, each with different policy map commands.

Configure a separate policy map class in the policy map for each type of traffic that an interface receives. Put all commands for each type of traffic in the same policy map class. MPLS QoS does not attempt to apply commands from more than one policy map class to matched traffic.

Configuring a Policy Map to Set the EXP Value on All Imposed Labels

To set the value of the MPLS EXP field on all imposed label entries, use the **set mpls experimental imposition** command in QoS policy-map class configuration mode. To disable the setting, use the **no** form of this command.

Note The **set mpls experimental imposition** command replaces the **set mpls experimental** command.

	Command	Purpose
Step 1	Router(config)# **policy-map** *policy_name*	Creates a policy map.
Step 2	Router(config-pmap)# **class-map** *name* [**match-all** | **match-any**]	Accesses the QoS class-map configuration mode to configure QoS class maps.
Step 3	Router(config-pmap-c)# **set mpls experimental imposition** {*mpls-exp-value* | *from-field* [**table** *table-map-name*]}	Sets the value of the MPLS experimental (EXP) field on all imposed label entries.
Step 4	Router(config-pmap-c)# **exit**	Exits class-map configuration mode.

The following example sets the MPLS EXP imposition value according to the DSCP value defined in the MPLS EXP value 3:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# access-l 101 p tcp any any
Router(config)# class-map iptcp
Router(config-cmap)# match acc 101
Router(config-cmap)# exit
Router(config)#
Router(config-cmap)# policy-map ip2tag
Router(config-pmap)# class iptcp
Router(config-pmap-c)# set mpls exp imposition 3
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#
Router#
1w4d: %SYS-5-CONFIG_I: Configured from console by console
Router#
Router# show policy-map ip2tag
  Policy Map ip2tag
    Class iptcp
      set mpls experimental imposition 3
Router# show class iptcp
 Class Map match-all iptcp (id 62)
   Match access-group101

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/27
Router(config-if)# ser in ip2tag
Router(config-if)#
Routers
1w4d: %SYS-5-CONFIG_I: Configured from console by console
Router# show pol ip2tag
  Policy Map ip2tag
    Class iptcp
      set mpls experimental imposition 3
Router# show class-map iptcp
```

```
 Class Map match-all iptcp (id 62)
   Match access-group  101

Router# show access-l 101
Extended IP access list 101
    10 permit tcp any any
Router# show mls qos ip
 QoS Summary [IPv4]:      (* - shared aggregates, Mod - switch module)

      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
-------------------------------------------------------------------------------
    Fa3/27  5  In      iptcp   24    2     No  0             0             0
     Vl300  5  In          x   44    1     No  0             0             0

       All  5   -    Default    0    0*    No  0     3466448105             0
Router#
Router# show policy-map interface fastethernet 3/27
 FastEthernet3/27

  Service-policy input: ip2tag

    class-map: iptcp (match-all)
      Match: access-group 101
      set mpls experimental 3:
      Earl in slot 5 :
        0 bytes
        5 minute offered rate 0 bps
        aggregate-forwarded 0 bytes

    class-map: class-default (match-any)
      Match: any

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

This example shows how to verify the configuration:

```
Router# show policy map ip2tag
  Policy Map ip2tag
    Class iptcp
      set mpls experimental imposition 3
```

EXP Value Imposition Guidelines and Restrictions

When setting the EXP value on all imposed labels, follow these guidelines and restrictions:

- Use the **set mpls experimental imposition** command during label imposition. This command sets the MPLS EXP field on all imposed label entries.
- The **set mpls experimental imposition** command is supported only on input interfaces (imposition).
- The **set mpls experimental imposition** command does not mark the EXP value directly; instead, it marks the internal DSCP that is mapped to EXP through the internal DSCP-to-EXP global map.
- It is important to note that classification (based on the original received IP header) and marking (done to the internal DSCP) do not distinguish between IP-to-IP traffic and IP-to-MPLS traffic. The commands that you use to mark IP ToS and mark EXP have the same result as when you mark the internal DSCP.

- To set the pushed label entry value to a value different from the default value during label imposition, use the **set mpls experimental imposition** command.
- You optionally can use the **set mpls experimental imposition** command with the IP precedence, DSCP field, or QoS IP ACL to set the value of the MPLS EXP field on all imposed label entries.
- When imposing labels onto the received IP traffic with the PFC, you can mark the EXP field using the **set mpls experimental imposition** command.

For more information on this command, see the *Cisco IOS Switching Services Command Reference, Release 12.3* located at this URL:

http://www.cisco.com/en/US/docs/ios/mpls/command/reference/mp_s1.html#set_mpls_experimental_imposition

Configuring a Policy Map Using the Police Command

Policing is a function in the PFC hardware that provides the ability to rate limit a particular traffic class to a specific rate. The PFC supports aggregate policing and microflow policing.

Aggregate policing meters all traffic that ingresses into a port, regardless of different source, destination, protocol, source port, or destination port. Microflow policing meters all traffic that ingresses into a port, on a per flow (per source, destination, protocol, source port, and destination port). For additional information on aggregate and microflow policing, see the "Policers" section on page 43-17.

To configure traffic policing, use the **police** command. For information on this command, see the Cisco IOS Master Command List.

	Command	Purpose
Step 1	`Router(config)#` **`policy-map`** *`policy_name`*	Creates a policy map.
Step 2	`Router(config-pmap)#` **`class-map`** *`name`* `[`**`match-all`** `|` **`match-any`**`]`	Accesses the QoS class map configuration mode to configure QoS class maps.
Step 3	`Router(config-pmap-c)#` **`police`** `{`**`aggregate`** *`name`*`}`	Adds the class to a shared aggregate policer.
Step 4	`Router(config-pmap-c)#` **`police`** *`bps burst_normal burst_max`* **`conform-action`** *`action`* **`exceed-action`** *`action`* **`violate-action`** *`action`*	Creates a per-class-per-interface policer.
Step 5	`Router(config-pmap-c)#` **`police flow`** `{`*`bps`* `[`*`burst_normal`*`]` `|` `[`**`conform-action`** *`action`*`]` `[`**`exceed-action`** *`action`*`]`	Creates an ingress flow policer. (Not supported in egress policy.)
Step 6	`Router(config-pmap-c)#` **`exit`**	Exits class-map configuration mode.

This is an example of creating a policy map with a policer:

```
Router(config)# policy-map ip2tag
Router(config-pmap)# class iptcp
Router(config-pmap-c)# no set mpls exp topmost 3
Router(config-pmap-c)# police 1000000 1000000 c set-mpls-exp?
set-mpls-exp-imposition-transmit

Router(config-pmap-c)# police 1000000 1000000 c set-mpls-exp-imposit 3 e d
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fastethernet 3/27
Router(config-if)# ser in ip2tag
Router(config-if)#
```

This is an example of verifying the configuration:

```
Router# show pol ip2tag
  Policy Map ip2tag
    Class iptcp
     police cir 1000000 bc 1000000 be 1000000 conform-action
set-mpls-exp-imposition-transmit 3 exceed-action drop
Router# show running-config interface fastethernet 3/27
Building configuration...

Current configuration : 202 bytes
!
interface FastEthernet3/27
  logging event link-status
  service-policy input ip2tag
end

Router# show mls qos ip
 QoS Summary [IPv4]:      (* - shared aggregates, Mod - switch module)

      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
-------------------------------------------------------------------------------
    Fa3/27  5  In      iptcp   24    2     No  0              0              0
     Vl300  5  In          x   44    1     No  0              0              0

       All  5   -    Default    0    0*    No  0     3468105262              0
Router# show policy interface fastethernet 3/27
 FastEthernet3/27

  Service-policy input: ip2tag

    class-map: iptcp (match-all)
      Match: access-group 101
      police :
        1000000 bps 1000000 limit 1000000 extended limit
      Earl in slot 5 :
        0 bytes
        5 minute offered rate 0 bps
        aggregate-forwarded 0 bytes action: set-mpls-exp-imposition-transmit
        exceeded 0 bytes action: drop
        aggregate-forward 0 bps exceed 0 bps

    class-map: class-default (match-any)
      Match: any

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
R7# show mls qos ip
 QoS Summary [IPv4]:      (* - shared aggregates, Mod - switch module)

      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
-------------------------------------------------------------------------------
    Fa3/27  5  In      iptcp   24    2     No  0              0              0
     Vl300  5  In          x   44    1     No  0              0              0

       All  5   -    Default    0    0*    No  0     3468161522              0
```

Restrictions and Usage Guidelines

The following restrictions and guidelines apply when using the **police** command to configure a policy map:

- With MPLS, the **exceed-action** *action* command and the **violate-action** *action* command work similarly to IP usage. The packet may get dropped or the EXP value is marked down. For information on how these actions affect IP-to-IP traffic, see the "Configuring a Policy Map" section on page 43-75.
- With MPLS, the **set-dscp transmit** *action* command and the **set-prec-transmit** *action* command set the internal DSCP that is mapped into the CoS bits, which affects queueing, however, they do not change the EXP value, except for imposition.
- When swapping labels for received MPLS traffic with the PFC, you can mark down out-of-profile traffic using the **police** command **exceed-action policed-dscp-transmit** and **violate-action policed-dscp-transmit** keywords. The PFC does not mark in-profile traffic; when marking down out-of-profile traffic, the PFC marks the outgoing topmost label. The PFC does not propagate the marking down through the label stack.
- With MPLS, the flow key is based on the label and EXP value; there is no flowmask option. Otherwise, flow key operation is similar to IP-to-IP. See the "Configuring a Policy Map" section on page 43-75.
- You can use the **police** command to set the pushed label entry value to a value different from the default value during label imposition.
- When imposing labels onto the received IP traffic with the PFC, you can mark the EXP field using the **conform-action set-mpls-exp-imposition-transmit** keywords.
- During IP-to-MPLS imposition, IP ToS marking is not supported. If you configure a policy to mark IP ToS, the PFC marks the EXP value.

Displaying a Policy Map

You can display a policy map with an interface summary for MPLS QoS classes or with the configuration of all classes configured for all service policies on the specified interface.

Displaying an MPLS QoS Policy Map Class Summary

To display an MPLS QoS policy map class summary, perform this task:

Command	Purpose
`Router# show mls qos mpls [{interface interface_type interface_number}	{module slot}`	Displays an MPLS QoS policy map class summary.

This example shows how to display an MPLS QoS policy map class summary:

```
Router# show mls qos mpls
 QoS Summary [MPLS]:      (* - shared aggregates, Mod - switch module)
      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
-------------------------------------------------------------------------------
    Fa3/27  5  In       exp3    0    2   dscp  0             0              0
       All  5   -    Default    0    0*    No  0     3466140423             0
```

Displaying the Configuration of All Classes

To display the configuration of all classes configured for all service policies on the specified interface, perform this task:

Command	Purpose
Router# **show policy interface** *interface_type interface_number*	Displays the configuration of all classes configured for all policy maps on the specified interface.

This example shows the configurations for all classes on Fast Ethernet interface 3/27:

```
Router# show policy interface fastethernet 3/27
 FastEthernet3/27

  Service-policy input: ip2tag

    class-map: iptcp (match-all)
      Match: access-group 101
      police :
        1000000 bps 1000000 limit 1000000 extended limit
      Earl in slot 5 :
        0 bytes
        5 minute offered rate 0 bps
        aggregate-forwarded 0 bytes action: set-mpls-exp-imposition-transmit
        exceeded 0 bytes action: drop
        aggregate-forward 0 bps exceed 0 bps

    class-map: class-default (match-any)
      Match: any

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Configuring MPLS QoS Egress EXP Mutation

You can configure a named egress EXP mutation map to mutate the internal DSCP-derived EXP value before it is used as the egress EXP value. You can attach a named egress EXP mutation map to any interface on which PFC QoS supports egress QoS.

These sections describe how to configure MPLS QoS EXP mutation:

- Configuring Named EXP Mutation Maps, page 45-29
- Attaching a Named EXP Mutation Map to an Egress Interface, page 45-29

Configuring Named EXP Mutation Maps

To configure a named EXP mutation map, perform this task:

	Command	Purpose
Step 1	`Router(config)# ` **`mls qos map exp-mutation`** *`map_name mutated_exp1 mutated_exp2 mutated_exp3 mutated_exp4 mutated_exp5 mutated_exp6 mutated_exp7 mutated_exp8`*	Configures a named EXP mutation map.
Step 2	`Router(config)# ` **`end`**	Exits configuration mode.
Step 3	`Router# ` **`show mls qos maps`**	Verifies the configuration.

When configuring a named EXP mutation map, note the following information:

- You can configure 7 named EXP mutation maps.
- You can enter up to eight EXP values that map to a mutated EXP value.
- You can enter multiple commands to map additional EXP values to a mutated EXP value.
- You can enter a separate command for each mutated EXP value.
- You can attach named EXP mutation maps to any interface on which PFC QoS supports egress QoS.

Attaching a Named EXP Mutation Map to an Egress Interface

To attach an EXP mutation map to an egress interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# ` **`interface`** `{{`**`vlan`** *`vlan_ID`*`}	{`*`type`*[1] *`slot/port`*`[.`*`subinterface`*`]}	{`**`port-channel`** *`number`*`[.`*`subinterface`*`]}}`	Selects the interface to configure.
Step 2	`Router(config-if)# ` **`mls qos exp-mutation`** *`exp-mutation-table-name`*	Attaches an egress EXP mutation map to the interface.
Step 3	`Router(config-if)# ` **`end`**	Exits configuration mode.
Step 4	`Router# ` **`show running-config interface`** `{{`**`vlan`** *`vlan_ID`*`}	{`*`type slot/port`*`}	{`**`port-channel`** *`number`*`}}`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to attach the egress EXP mutation map named mutemap2:

```
Router(config)# interface fastethernet 3/26
Router(config-if)# mls qos exp-mutation mutemap2
Router(config-if)# end
```

Configuring EXP Value Maps

These sections describe how EXP values are mapped to other values:

- Configuring the Ingress-EXP to Internal-DSCP Map, page 45-30
- Configuring the Egress-DSCP to Egress-EXP Map, page 45-30

Configuring the Ingress-EXP to Internal-DSCP Map

To configure the ingress-EXP to internal-DSCP map, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos map exp-dscp dscp_values`	Configures the ingress-EXP value to internal-DSCP map. You must enter eight DSCP values corresponding to the EXP values. Valid values are 0 through 63.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos maps`	Verifies the configuration.

This example shows how to configure the ingress-EXP to internal-DSCP map:

```
Router(config)# mls qos map exp-dscp 43 43 43 43 43 43 43 43
Router(config)#
```

This example shows how to verify the configuration:

```
Router(config)# show mls qos map exp-dscp
   Exp-dscp map:
         exp:    0   1   2   3   4   5   6   7
     -----------------------------------
        dscp:  43 43 43 43 43 43 43 43
```

Configuring the Egress-DSCP to Egress-EXP Map

To configure the egress-DSCP to egress-EXP map, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos map dscp-exp dscp_values to exp_values`	Configures the egress-DSCP to egress-EXP map. You can enter up to eight DSCP values at one time to a single EXP value. Valid values are 0 through 7.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos maps`	Verifies the configuration.

This example shows how to configure the egress-DSCP to egress-EXP map:

```
Router(config)# mls qos map dscp-exp 20 25 to 3
Router(config)#
```

MPLS DiffServ Tunneling Modes

Tunneling provides QoS the ability to be transparent from one edge of a network to the other edge of the network. A tunnel starts where there is label imposition. A tunnel ends where there is label disposition; that is, where the label is removed from the stack, and the packet goes out as an MPLS packet with a different per-hop behavior (PHB) layer underneath or as an IP packet with the IP PHB layer.

For the PFC, there are two ways to forward packets through a network:

- Short Pipe mode—In Short Pipe mode, the egress PE router uses the original packet marking instead of the marking used by the intermediate provider (P) routers. EXP marking does not propagate to the packet ToS byte.

  For a description of this mode, see the "Short Pipe Mode" section on page 45-31.

  For the configuration information, see the "Configuring Short Pipe Mode" section on page 45-34.

- Uniform mode—In Uniform mode, the marking in the IP packet may be manipulated to reflect the service provider's QoS marking in the core. This mode provides consistent QoS classification and marking throughout the network including CE and core routers. EXP marking is propagated to the underlying ToS byte.

  For a description, see the "Uniform Mode" section on page 45-32.

  For the configuration procedure, see the "Configuring Uniform Mode" section on page 45-39.

Both tunneling modes affect the behavior of edge and penultimate label switching routers (LSRs) where labels are put onto packets and removed from packets. They do not affect label swapping at intermediate routers. A service provider can choose different types of tunneling modes for each customer.

For additional information, see "MPLS DiffServ Tunneling Modes" at this URL:
http://www.cisco.com/en/US/docs/ios-xml/ios/mp_te_diffserv/configuration/15-mt/mp-diffserv-tun-mode.html.

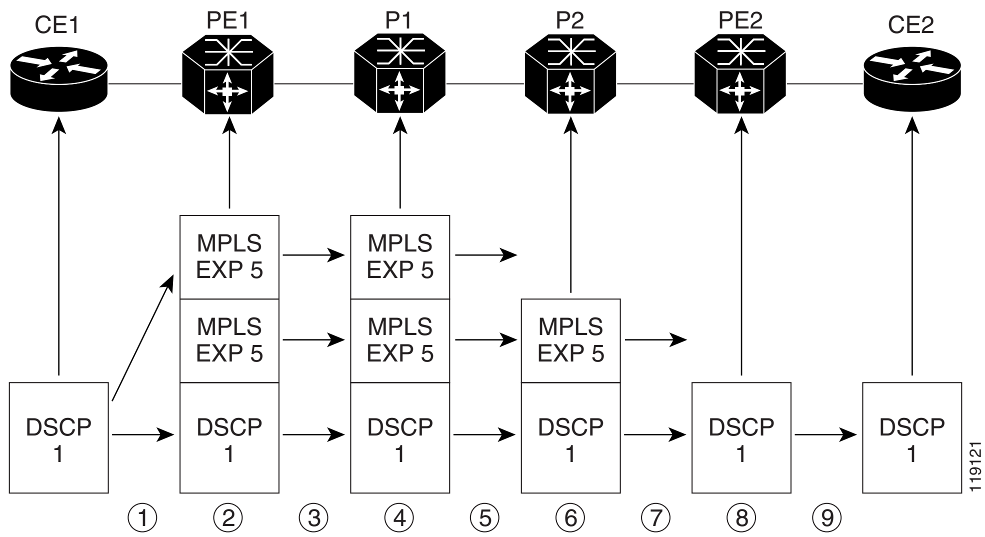
Short Pipe Mode

Short pipe mode is used when the customer and service provider are in different DiffServ domains. It allows the service provider to enforce its own DiffServ policy while preserving customer DiffServ information, which provides a DiffServ transparency through the service provider network.

QoS policies implemented in the core do not propagate to the packet ToS byte. The classification based on MPLS EXP value ends at the customer-facing egress PE interface; classification at the customer-facing egress PE interface is based on the original IP packet header and not the MPLS header.

Note The presence of an egress IP policy (based on the customer's PHB marking and not on the provider's PHB marking) automatically implies the Short Pipe mode.

Figure 45-2 Short Pipe Mode Operation with VPNs



Short Pipe mode functions as follows:

1. CE1 transmits an IP packet to PE1 with an IP DSCP value of 1.
2. PE1 sets the MPLS EXP field to 5 in the imposed label entries.
3. PE1 transmits the packet to P1.
4. P1 sets the MPLS EXP field value to 5 in the swapped label entry.
5. P1 transmits the packet to P2.
6. P2 pops the IGP label entry.
7. P2 transmits the packet to PE2.
8. PE2 pops the BGP label.
9. PE2 transmits the packet to CE2, but does QoS based on the IP DSCP value.

For additional information, see “MPLS DiffServ Tunneling Modes” at this URL:

http://www.cisco.com/en/US/docs/ios-xml/ios/mp_te_diffserv/configuration/15-mt/mp-diffserv-tun-mode.html

Short Pipe Mode Restrictions and Guidelines

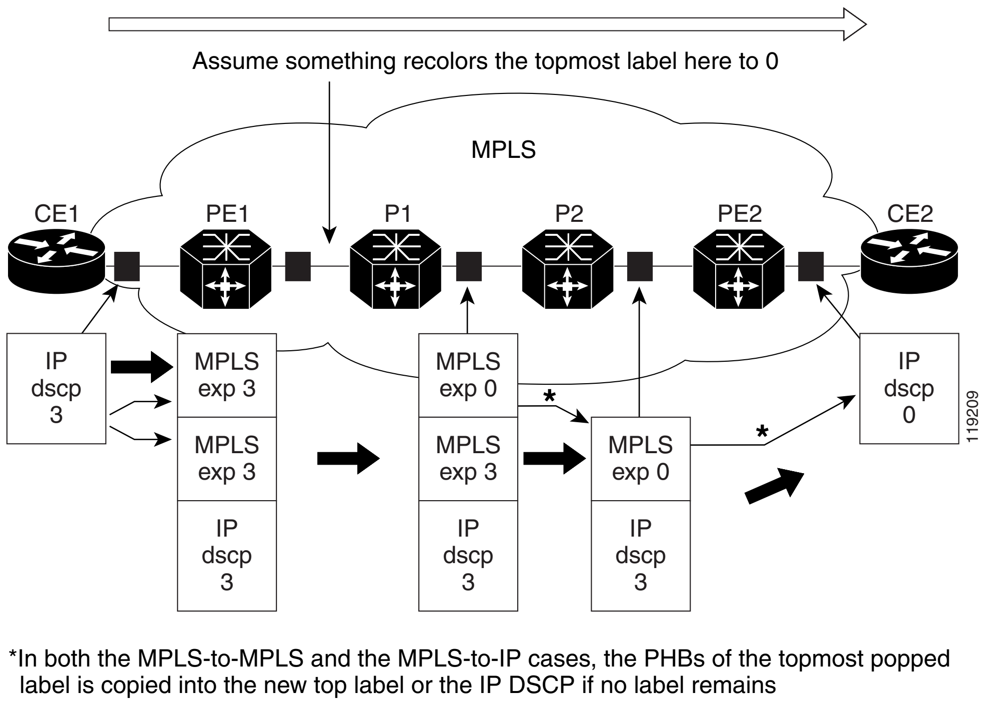
The following restriction applies to Short Pipe mode:

Short Pipe mode is not supported if the MPLS-to-IP egress interface is EoMPLS (the adjacency has the end of marker (EOM) bit set).

Uniform Mode

In Uniform mode, packets are treated uniformly in the IP and MPLS networks; that is, the IP precedence value and the MPLS EXP bits always correspond to the same PHB. Whenever a router changes or recolors the PHB of a packet, that change must be propagated to all encapsulation markings. The propagation is performed by a router only when a PHB is added or exposed due to label imposition or disposition on any router in the packet's path. The color must be reflected everywhere at all levels. For example, if a packet's QoS marking is changed in the MPLS network, the IP QoS marking reflects that change.

Figure 45-3 Uniform Mode Operation



The procedure varies according to whether IP precedence bit markings or DSCP markings are present.

The following actions occur if there are IP precedence bit markings:

1. IP packets arrive in the MPLS network at PE1, the service provider edge router.
2. A label is copied onto the packet.
3. If the MPLS EXP field value is recolored (for example, if the packet becomes out-of-rate because too many packets are being transmitted), that value is copied to the IGP label. The value of the BGP label is not changed.
4. At the penultimate hop, the IGP label is removed. That value is copied into the next lower level label.
5. When all MPLS labels have been removed from the packet that is sent out as an IP packet, the IP precedence or DSCP value is set to the last changed EXP value in the core.

The following is an example when there are IP precedence bit markings:

1. At CE1 (customer equipment 1), the IP packet has an IP precedence value of 3.
2. When the packet arrives in the MPLS network at PE1 (the service provider edge router), the IP precedence value of 3 is copied to the imposed label entries of the packet.
3. The MPLS EXP field in the IGP label header might be changed within the MPLS core (for example, at P1) by a mark down.

Note Because the IP precedence bits are 3, the BGP label and the IGP label also contain 3 because in Uniform mode, the labels always are identical. The packet is treated uniformly in the IP and MPLS networks.

Uniform Mode Restrictions and Guidelines

The following restriction applies to the Uniform mode:

If the egress IP ACLs or service policies are configured on the MPLS-to-IP exit point, the Uniform mode is always enforced because of recirculation.

MPLS DiffServ Tunneling Restrictions and Usage Guidelines

The MPLS DiffServ tunneling restrictions and usage guidelines are as follows:

- One label-switched path (LSP) can support up to eight classes of traffic (that is, eight PHBs) because the MPLS EXP field is a 3-bit field.
- MPLS DiffServ tunneling modes support E-LSPs. An E-LSP is an LSP on which nodes determine the QoS treatment for MPLS packet exclusively from the EXP bits in the MPLS header.

The following features are supported with the MPLS differentiated service (DiffServ) tunneling modes:

- MPLS per-hop behavior (PHB) layer management. (Layer management is the ability to provide an additional layer of PHB marking to a packet.)
- Improved scalability of the MPLS layer management by control on managed customer edge (CE) routers.
- MPLS can tunnel a packet's QoS (that is, the QoS is transparent from edge to edge). With QoS transparency, the IP marking in the IP packet is preserved across the MPLS network.
- The MPLS EXP field can be marked differently and separately from the PHB marked in the IP precedence or DSCP field.

Configuring Short Pipe Mode

The following sections describe how to configure the Short Pipe mode:

- Ingress PE Router—Customer Facing Interface, page 45-35
- Configuring Ingress PE Router—P Facing Interface, page 45-36
- Configuring the P Router—Output Interface, page 45-37
- Configuring the Egress PE Router—Customer Facing Interface, page 45-38

Note

- The steps that follow show one way, but not the only way, to configure Short Pipe mode.
- The Short Pipe mode on the egress PE (or PHP) is automatically configured when you attach to the interface an egress service policy that includes an IP class.

Ingress PE Router—Customer Facing Interface

This procedure configures a policy map to set the MPLS EXP field in the imposed label entries.

To set the EXP value, the ingress LAN port must be untrusted. FlexWAN ports do not have the trust concept, but, as with traditional Cisco IOS routers, the ingress ToS is not changed (unless a marking policy is configured).

For MPLS and VPN, the ingress PE supports all ingress PFC IP policies. For information about the classification for PFC IP policies based on IP ACL/DSCP/precedence, see Chapter 43, "Configuring PFC QoS."

To configure a policy map to set the MPLS EXP field in the imposed label entries, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos**	Enables PFC QoS globally.
Step 2	Router(config)# **access-list** *ipv4_acl_number_or_name* **permit any**	Creates an IPv4 access list.
Step 3	Router(config)# **class-map** *class_name*	Creates a class map.
Step 4	Router(config-cmap)# **match access-group** *ipv4_acl_number_or_name*	Configures the class map to filter with the ACL created in step 2.
Step 5	Router(config)# **policy-map** *policy_map_name*	Creates a named QoS policy.
Step 6	Router(config-pmap)# **class** *class_name*	Configures the policy to use the class map created in step 3.
Step 7	Router(config-pmap-c)# **police** *bits_per_second* [*normal_burst_bytes*] **conform-action set-mpls-exp-transmit** *exp_value* **exceed-action drop**	Configures policing, including the following: • Action to take on packets that conform to the rate limit specified in the service level agreement (SLA). • Action to take on packets that exceed the rate limit specified in the SLA. The *exp_value* sets the MPLS EXP field.
Step 8	Router(config)# **interface** *type slot/port*	Selects an interface to configure.
Step 9	Router(config-if)# **no mls qos trust**	Configures the interface as untrusted.
Step 10	Router(config-if)# **service-policy input** *policy_map_name*	Attaches the policy map created in step 5 to the interface as an input service policy.

Configuration Example

This example shows how to configure a policy map to set the MPLS EXP field in the imposed label entries:

```
Router(config)# mls qos
Router(config)# access-list 1 permit any
Router(config)# class-map CUSTOMER-A
Router(config-cmap)# match access-group 1
Router(config)# policy-map set-MPLS-PHB
Router(config-pmap)# class CUSTOMER-A
Router(config-pmap-c)# police 50000000 conform-action set-mpls-exp-transmit 4
exceed-action drop
Router(config)# interface GE-WAN 3/1
Router(config-if)# no mls qos trust
Router(config)# interface GE-WAN 3/1.31
Router(config-if)# service-policy input set-MPLS-PHB
```

Configuring Ingress PE Router—P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

Note QoS features shown here are available only with FlexWAN and Enhanced FlexWAN modules.

To classify packets based on their MPLS EXP field and provide appropriate discard and scheduling treatments, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`mls qos`**	Enables PFC QoS globally.
Step 2	`Router(config)#` **`class-map`** *`class_name`*	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	`Router(config-c-map)#` **`match mpls experimental`** *`exp_list`*	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 4	`Router(config)#` **`policy-map`** *`name`*	Configures the QoS policy for packets that match the class or classes.
Step 5	`Router(config-p-map)#` **`class`** *`class_name`*	Associates the traffic class with the service policy.
Step 6	`Router(config-p-map-c)#` **`bandwidth`** {*`bandwidth_kbps`* | **`percent`** *`percent`*}	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	`Router(config-p-map)#` **`class class-default`**	Specifies the default class so that you can configure or modify its policy.
Step 8	`Router(config-p-map-c)#` **`random-detect`**	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 9	`Router(config)#` **`interface`** *`type slot/port`*	Selects an interface to configure.
Step 10	`Router(config-if)#` **`service-policy output`** *`name`*	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Note The **bandwidth** command and **random-detect** command are not supported on LAN ports.

Configuration Example

This example shows how to classify packets based on their MPLS EXP field and provide appropriate discard and scheduling treatments:

```
Router(config)# mls qos
Router(config)# class-map MPLS-EXP-4
Router(config-c-map)# match mpls experimental 4
Router(config)# policy-map output-qos
Router(config-p-map)# class MPLS-EXP-4
Router(config-p-map-c)# bandwidth percent 40
Router(config-p-map)# class class-default
Router(config-p-map-c)# random-detect
```

```
Router(config)# interface pos 4/1
Router(config-if)# service-policy output output-qos
```

Configuring the P Router—Output Interface

Note QoS features shown here are available only with FlexWAN and Enhanced FlexWAN modules.

To classify packets based on their MPLS EXP field and provide appropriate discard and scheduling treatments, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos`	Enables PFC QoS globally.
Step 2	`Router(config)# class-map class_name`	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	`Router(config-c-map)# match mpls experimental exp_list`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 4	`Router(config)# policy-map name`	Configures the QoS policy for packets that match the class or classes.
Step 5	`Router(config-p-map)# class class_name`	Associates the traffic class with the service policy.
Step 6	`Router(config-p-map-c)# bandwidth {bandwidth_kbps | percent percent}`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	`Router(config-p-map)# class class-default`	Specifies the default class so that you can configure or modify its policy.
Step 8	`Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP precedence or the MPLS EXP field value.
Step 9	`Router(config)# interface type slot/port`	Selects an interface to configure.
Step 10	`Router(config-if)# service-policy output name`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Note The **bandwidth** command and **random-detect** command are not supported on LAN ports.

Configuration Example

This example shows how to classify packets based on their MPLS EXP field and provide appropriate discard and scheduling treatments:

```
Router(config)# mls qos
Router(config)# class-map MPLS-EXP-4
Router(config-c-map)# match mpls experimental 4
Router(config)# policy-map output-qos
Router(config-p-map)# class MPLS-EXP-4
Router(config-p-map-c)# bandwidth percent 40
Router(config-p-map)# class class-default
```

```
Router(config-p-map-c)# random-detect
Router(config)# interface pos 2/1
Router(config-if)# service-policy output output-qos
```

Configuring the Egress PE Router—Customer Facing Interface

Note QoS features shown here are available only with FlexWAN and Enhanced FlexWAN modules.

To classify a packet based on its IP DSCP value and provide appropriate discard and scheduling treatments, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos**	Enables PFC QoS globally.
Step 2	Router(config)# **class-map** *class_name*	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	Router(config-c-map)# **match ip dscp** *dscp_values*	Uses the DSCP values as the match criteria.
Step 4	Router(config)# **policy-map** *name*	Configures the QoS policy for packets that match the class or classes.
Step 5	Router(config-p-map)# **class** *class_name*	Associates the traffic class with the service policy.
Step 6	Router(config-p-map-c)# **bandwidth** {*bandwidth_kbps* | **percent** *percent*}	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	Router(config-p-map)# **class class-default**	Specifies the default class so that you can configure or modify its policy.
Step 8	Router(config-p-map-c)# **random-detect dscp-based**	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 9	Router(config)# **interface** *type slot/port*	Selects an interface to configure.
Step 10	Router(config-if)# **service-policy output** *name*	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Note The **bandwidth** command and **random-detect** command are not supported on LAN ports.

Configuration Example

This example shows how to classify a packet based on its IP DSCP value and provide appropriate discard and scheduling treatments:

```
Router(config)# mls qos
Router(config)# class-map IP-PREC-4
Router(config-c-map)# match ip precedence 4
Router(config)# policy-map output-qos
Router(config-p-map)# class IP-PREC-4
Router(config-p-map-c)# bandwidth percent 40
Router(config-p-map)# class class-default
Router(config-p-map-c)# random-detect
```

```
Router(config)# interface GE-WAN 3/2.32
Router(config-if)# service-policy output output-qos
```

Configuring Uniform Mode

This section describes how to configure the following:

Note The steps that follow show one way, but not the only way, to configure the Uniform mode.

Configuring the Ingress PE Router—Customer Facing Interface

For Uniform mode, setting the trust state to IP precedence or IP DSCP allows the PFC to copy the IP PHB into the MPLS PHB.

Note This description applies to PFC QoS for LAN ports. For information about FlexWAN and Enhanced FlexWAN QoS, see the *FlexWAN and Enhanced FlexWAN Modules Installation and Configuration Guide* at this URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexwan-config-guide.html.

To configure a policy map to set the MPLS EXP field in imposed label entries, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos**	Enables PFC QoS globally.
Step 2	Router(config)# **access-list** *ipv4_acl_number_or_name* **permit any**	Creates an IPv4 access list.
Step 3	Router(config)# **class-map** *class_name*	Creates a class map.
Step 4	Router(config-cmap)# **match access-group** *ipv4_acl_number_or_name*	Configures the class map to filter with the ACL created in Step 2.
Step 5	Router(config)# **policy-map** *policy_map_name*	Creates a named QoS policy.
Step 6	Router(config-pmap)# **class** *class_name*	Configures the policy to use the class map created in step 3.
Step 7	Router(config-pmap-c)# **police** *bits_per_second* [*normal_burst_bytes*] **conform-action transmit exceed-action drop**	Configures policing, including the following: • Action to take on packets that conform to the rate limit specified in the SLA. • Action to take on packets that exceed the rate limit specified in the SLA.
Step 8	Router(config)# **interface** *type slot/port*	Selects an interface to configure.

	Command	Purpose
Step 9	Router(config-if)# **mls qos trust dscp**	Configures received DSCP as the basis of the internal DSCP for all the port's ingress traffic.
Step 10	Router(config-if)# **service-policy input** *policy_map_name*	Attaches the policy map created in step 5 to the interface as an input service policy.

Configuration Example

This example shows how to configure a policy map to set the MPLS EXP field in imposed label entries:

```
Router(config)# mls qos
Router(config)# access-list 1 permit any
Router(config)# class-map CUSTOMER-A
Router(config-cmap)# match access-group 1
Router(config)# policy-map SLA-A
Router(config-pmap)# class CUSTOMER-A
Router(config-pmap-c)# police 50000000 conform-action transmit exceed-action drop
Router(config)# interface GE-WAN 3/1
Router(config-if)# mls qos trust dscp
Router(config)# interface GE-WAN 3/1.31
Router(config-if)# service-policy input SLA-A
```

Configuring the Ingress PE Router—P Facing Interface

To classify packets based on their MPLS EXP field and provide appropriate discard and scheduling treatments, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos**	Enables PFC QoS globally.
Step 2	Router(config)# **class-map** *class_name*	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	Router(config-c-map)# **match mpls experimental** *exp_list*	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 4	Router(config)# **policy-map** *name*	Configures the QoS policy for packets that match the class or classes.
Step 5	Router(config-p-map)# **class** *class_name*	Associates the traffic class with the service policy.
Step 6	Router(config-p-map-c)# **bandwidth** {*bandwidth_kbps* | **percent** *percent*}	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	Router(config-p-map)# **class class-default**	Specifies the default class so that you can configure or modify its policy.
Step 8	Router(config-p-map-c)# **random-detect**	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 9	Router(config)# **interface** *type slot/port*	Selects an interface to configure.
Step 10	Router(config-if)# **service-policy output** *name*	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Note The **bandwidth** command and **random-detect** command are not supported on LAN ports.

Configuration Example

This example shows how to classify packets based on their MPLS EXP field and provide appropriate discard and scheduling treatments:

```
Router(config)# mls qos
Router(config)# class-map MPLS-EXP-3
Router(config-c-map)# match mpls experimental 3
Router(config)# policy-map output-qos
Router(config-p-map)# class MPLS-EXP-3
Router(config-p-map-c)# bandwidth percent 40
Router(config-p-map)# class class-default
Router(config-p-map-c)# random-detect
Router(config)# interface pos 4/1
Router(config-if)# service-policy output output-qos
```

Configuring the Egress PE Router—Customer Facing Interface

To configure the egress PE router at the customer-facing interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`mls qos`**	Enables PFC QoS globally.
Step 2	`Router(config)#` **`class-map`** *`class_name`*	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	`Router(config-c-map)#` **`match ip precedence`** ***`precedence-value`***	Identifies IP precedence values as match criteria.
Step 4	`Router(config)#` **`policy-map`** *`name`*	Configures the QoS policy for packets that match the class or classes.
Step 5	`Router(config-p-map)#` **`class`** *`class_name`*	Associates the traffic class with the service policy.
Step 6	`Router(config-p-map-c)#` **`bandwidth`** {*`bandwidth_kbps`* | **`percent`** *`percent`*}	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	`Router(config-p-map)#` **`class class-default`**	Specifies the default class so that you can configure or modify its policy.
Step 8	`Router(config-p-map-c)#` **`random-detect`**	Applies WRED to the policy based on the IP precedence or the MPLS EXP field value.
Step 9	`Router(config)#` **`interface`** *`type slot/port`*	Selects an interface to configure.
Step 10	`Router(config-if)` **`mpls propagate-cos`**	Enables propagation of EXP value into the underlying IP DSCP at the MPLS domain exit LER egress port.
Step 11	`Router(config-if)#` **`service-policy output`** *`name`*	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Note The **bandwidth** command and **random-detect** command are not supported on LAN ports.

Configuration Example

This example shows how to configure the egress PE router at the customer-facing interface:

```
Router(config)# mls qos
Router(config)# class-map IP-PREC-4
Router(config-c-map)# match ip precedence 4
Router(config)# policy-map output-qos
Router(config-p-map)# class IP-PREC-4
Router(config-p-map-c)# bandwidth percent 40
Router(config-p-map)# class class-default
Router(config-p-map-c)# random-detect
Router(config)# interface GE-WAN 3/2.32
Router(config-if) mpls propagate-cos
Router(config-if)# service-policy output output-qos
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring PFC QoS Statistics Data Export

This chapter describes how to configure PFC QoS statistics data export in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains these sections:

Understanding PFC QoS Statistics Data Export

The PFC QoS statistics data export feature generates per-LAN-port and per-aggregate policer utilization information and forwards this information in UDP packets to traffic monitoring, planning, or accounting applications. You can enable PFC QoS statistics data export on a per-LAN-port or on a per-aggregate policer basis. The statistics data generated per port consists of counts of the input and output packets and bytes. The aggregate policer statistics consist of counts of allowed packets and counts of packets exceeding the policed rate.

The PFC QoS statistics data collection occurs periodically at a fixed interval, but you can configure the interval at which the data is exported. PFC QoS statistics collection is enabled by default, and the data export feature is disabled by default for all ports and all configured aggregate policers.

Note The PFC QoS statistics data export feature is completely separate from NetFlow Data Export and does not interact with it.

PFC QoS Statistics Data Export Default Configuration

Table 46-1 shows the PFC QoS statistics data export default configuration.

Table 46-1 PFC QoS Default Configuration

Feature	Default Value
PFC QoS Data Export	
Global PFC QoS data export	Disabled
Per port PFC QoS data export	Disabled
Per named aggregate policer PFC QoS data export	Disabled
Per class map policer PFC QoS data export	Disabled
PFC QoS data export time interval	300 seconds
Export destination	Not configured
PFC QoS data export field delimiter	Pipe character (|)

Configuring PFC QoS Statistics Data Export

These sections describe how to configure PFC QoS statistics data export:

Enabling PFC QoS Statistics Data Export Globally

To enable PFC QoS statistics data export globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos statistics-export`	Enables PFC QoS statistics data export globally.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos statistics-export info`	Verifies the configuration.

This example shows how to enable PFC QoS statistics data export globally and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export
Router(config)# end
% Warning: Export destination not set.
% Use 'mls qos statistics-export destination' command to configure the export destination
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
---------------------------------------------------------------
Export Status : enabled
Export Interval : 300 seconds
Export Delimiter : |
Export Destination : Not configured
Router#
```

Note You must enable PFC QoS statistics data export globally for other PFC QoS statistics data export configuration to take effect.

Enabling PFC QoS Statistics Data Export for a Port

To enable PFC QoS statistics data export for a port, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** *type*[1] *slot/port*	Selects the interface to configure.
Step 2	Router(config-if)# **mls qos statistics-export**	Enables PFC QoS statistics data export for the port.
Step 3	Router(config)# **end**	Exits configuration mode.
Step 4	Router# **show mls qos statistics-export info**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable PFC QoS statistics data export on FastEthernet port 5/24 and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 5/24
Router(config-if)# mls qos statistics-export
Router(config-if)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
---------------------------------------------------------------
Export Status : enabled
Export Interval : 300 seconds
Export Delimiter : |
Export Destination : Not configured

QoS Statistics Data Export is enabled on following ports:
---------------------------------------------------------
FastEthernet5/24
Router#
```

When enabled on a port, PFC QoS statistics data export contains the following fields, separated by the delimiter character:

- Export type (“1” for a port)
- Slot/port

- Number of ingress packets
- Number of ingress bytes
- Number of egress packets
- Number of egress bytes
- Time stamp

Enabling PFC QoS Statistics Data Export for a Named Aggregate Policer

To enable PFC QoS statistics data export for a named aggregate policer, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos statistics-export aggregate-policer** *aggregate_policer_name*	Enables PFC QoS statistics data export for a named aggregate policer.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show mls qos statistics-export info**	Verifies the configuration.

This example shows how to enable PFC QoS statistics data export for an aggregate policer named aggr1M and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export aggregate-policer aggr1M
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
---------------------------------------------------------------
Export Status : enabled
Export Interval : 300 seconds
Export Delimiter : |
Export Destination : Not configured

QoS Statistics Data Export is enabled on following ports:
---------------------------------------------------------
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----------------------------------------------------------------------------
aggr1M
Router#
```

When enabled for a named aggregate policer, PFC QoS statistics data export contains the following fields, separated by the delimiter character:

- Export type (“3” for an aggregate policer)
- Aggregate policer name
- Direction (“in”)
- PFC or DFC slot number
- Number of in-profile bytes
- Number of bytes that exceed the CIR
- Number of bytes that exceed the PIR
- Time stamp

Enabling PFC QoS Statistics Data Export for a Class Map

To enable PFC QoS statistics data export for a class map, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos statistics-export class-map** *classmap_name*	Enables PFC QoS statistics data export for a class map.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show mls qos statistics-export info**	Verifies the configuration.

This example shows how to enable PFC QoS statistics data export for a class map named class3 and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export class-map class3
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
---------------------------------------------------------------
Export Status : enabled
Export Interval : 300 seconds
Export Delimiter : |
Export Destination : Not configured

QoS Statistics Data Export is enabled on following ports:
---------------------------------------------------------
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----------------------------------------------------------------------------
aggr1M

QoS Statistics Data Export is enabled on following class-maps:
---------------------------------------------------------------
class3
Router#
```

When enabled for a class map, PFC QoS statistics data export contains the following fields, separated by the delimiter character:

- For data from a physical port:
  - Export type (“4” for a classmap and port)
  - Class map name
  - Direction (“in”)
  - Slot/port
  - Number of in-profile bytes
  - Number of bytes that exceed the CIR
  - Number of bytes that exceed the PIR
  - Time stamp
- For data from a VLAN interface:
  - Export type (“5” for a class map and VLAN)
  - Classmap name

- Direction (“in”)
- PFC or DFC slot number
- VLAN ID
- Number of in-profile bytes
- Number of bytes that exceed the CIR
- Number of bytes that exceed the PIR
- Time stamp

- For data from a port channel interface:
  - Export type (“6” for a class map and port channel)
  - Class map name
  - Direction (“in”)
  - PFC or DFC slot number
  - Port channel ID
  - Number of in-profile bytes
  - Number of bytes that exceed the CIR
  - Number of bytes that exceed the PIR
  - Time stamp

Setting the PFC QoS Statistics Data Export Time Interval

To set the time interval for the PFC QoS statistics data export, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos statistics-export interval interval_in_seconds`	Sets the time interval for the PFC QoS statistics data export. **Note** The interval needs to be short enough to avoid counter wraparound with the activity in your configuration, but because exporting PFC QoS statistic creates a significant load on the switch, be careful when decreasing the interval.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos statistics-export info`	Verifies the configuration.

This example shows how to set the PFC QoS statistics data export interval and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export interval 250
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
---------------------------------------------------------------
Export Status : enabled
Export Interval : 250 seconds
Export Delimiter : |
Export Destination : Not configured
```

```
QoS Statistics Data Export is enabled on following ports:
---------------------------------------------------------
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----------------------------------------------------------------------------
aggr1M

QoS Statistics Data Export is enabled on following class-maps:
---------------------------------------------------------------
class3
Router#
```

Configuring PFC QoS Statistics Data Export Destination Host and UDP Port

To configure the PFC QoS statistics data export destination host and UDP port number, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`mls qos statistics-export destination`** `{`*`host_name`* `|` *`host_ip_address`*`} {`**`port`** *`port_number`* `|` **`syslog`** `[`**`facility`** *`facility_name`*`] [`**`severity`** *`severity_value`*`]}`	Configures the PFC QoS statistics data export destination host and UDP port number.
	`Router(config)#` **`no mls qos statistics-export destination`**	Clears configured values.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.
Step 3	`Router#` **`show mls qos statistics-export info`**	Verifies the configuration.

Note When the PFC QoS data export destination is a syslog server, the exported data is prefaced with a syslog header.

Table 46-2 lists the supported PFC QoS data export facility and severity parameter values.

Table 46-2 Supported PFC QoS Data Export Facility Parameter Values

Name	Definition	Name	Definition
kern	kernel messages	cron	cron/at subsystem
user	random user-level messages	local0	reserved for local use
mail	mail system	local1	reserved for local use
daemon	system daemons	local2	reserved for local use
auth	security/authentication messages	local3	reserved for local use
syslog	internal syslogd messages	local4	reserved for local use
lpr	line printer subsytem	local5	reserved for local use
news	netnews subsytem	local6	reserved for local use
uucp	uucp subsystem	local7	reserved for local use

Table 46-3 lists the supported PFC QoS data export severity parameter values.

Table 46-3 Supported PFC QoS Data Export Severity Parameter Values

Severity Parameter		
Name	**Number**	**Definition**
emerg	0	system is unusable
alert	1	action must be taken immediately
crit	2	critical conditions
err	3	error conditions
warning	4	warning conditions
notice	5	normal but significant condition
info	6	informational
debug	7	debug-level messages

This example shows how to configure 172.20.52.3 as the destination host and syslog as the UDP port number and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export destination 172.20.52.3 syslog
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
---------------------------------------------------------------
Export Status : enabled
Export Interval : 250 seconds
Export Delimiter : |
Export Destination : 172.20.52.3, UDP port 514 Facility local6, Severity debug

QoS Statistics Data Export is enabled on following ports:
---------------------------------------------------------
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----------------------------------------------------------------------------
aggr1M

QoS Statistics Data Export is enabled on following class-maps:
---------------------------------------------------------------
class3
```

Setting the PFC QoS Statistics Data Export Field Delimiter

To set the PFC QoS statistics data export field delimiter, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls qos statistics-export delimiter** *delimiter_character*	Sets the PFC QoS statistics data export field delimiter.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show mls qos statistics-export info**	Verifies the configuration.

This example shows how to set the PFC QoS statistics data export field delimiter and verify the configuration:

```
Router# configure terminal
Router(config)# mls qos statistics-export delimiter ,
Router(config)# end
Router# show mls qos statistics-export info
QoS Statistics Data Export Status and Configuration information
---------------------------------------------------------------
Export Status : enabled
Export Interval : 250 seconds
Export Delimiter : ,
Export Destination : 172.20.52.3, UDP port 514 Facility local6, Severity debug

QoS Statistics Data Export is enabled on following ports:
---------------------------------------------------------
FastEthernet5/24

QoS Statistics Data export is enabled on following shared aggregate policers:
-----------------------------------------------------------------------------
aggr1M

QoS Statistics Data Export is enabled on following class-maps:
---------------------------------------------------------------
class3
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 9

Security

Configuring Network Security

This chapter contains network security information unique to Cisco IOS Release 12.2SX, which supplements the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.2, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

- *Cisco IOS Security Command Reference*, Release 12.2, at this URL

  http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Configuring MAC Address-Based Traffic Blocking

To block all traffic to or from a MAC address in a specified VLAN, perform this task:

Command	Purpose
`Router(config)#` **`mac-address-table static`** *`mac_address`* **`vlan`** *`vlan_ID`* **`drop`**	Blocks all traffic to or from the configured MAC address in the specified VLAN.
`Router(config)#` **`no mac-address-table static`** *`mac_address`* **`vlan`** *`vlan_ID`*	Clears MAC address-based blocking.

This example shows how to block all traffic to or from MAC address 0050.3e8d.6400 in VLAN 12:

```
Router# configure terminal
Router(config)# mac-address-table static 0050.3e8d.6400 vlan 12 drop
```

Configuring TCP Intercept

TCP intercept flows are processed in hardware.

For configuration procedures, see the *Cisco IOS Security Configuration Guide*, Release 12.2, "Traffic Filtering and Firewalls," "Configuring TCP Intercept (Preventing Denial-of-Service Attacks)," at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfdenl.html

Configuring Unicast Reverse Path Forwarding Check

These sections describe configuring unicast reverse path forwarding check (unicast RPF check):

- Understanding PFC3 Unicast RPF Check Support, page 47-2
- Unicast RPF Check Guidelines and Restrictions, page 47-3
- Configuring Unicast RPF Check, page 47-4

Understanding PFC3 Unicast RPF Check Support

The unicast RPF check verifies that the source address of received IP packets is reachable. The unicast RPF check discards IP packets that lack a verifiable IP source prefix (route), which helps mitigate problems that are caused by traffic with malformed or forged (spoofed) IP source addresses.

For unicast RPF check without ACL filtering, the PFC3 provides hardware support for the RPF check of traffic from multiple interfaces.

For unicast RPF check with ACL filtering, the PFC determines whether or not traffic matches the ACL. The PFC sends the traffic denied by the RPF ACL to the route processor (RP) for the unicast RPF check. Packets permitted by the ACL are forwarded in hardware without a unicast RPF check.

Unicast RPF Check Guidelines and Restrictions

These sections describe the unicast RPF check guidelines and restrictions:

General Unicast RPF Check Guidelines and Restrictions

When configuring unicast RPF check, follow these guidelines and restrictions:

- The PFC does not provide hardware support for the unicast RPF check for policy-based routing (PBR) traffic. (CSCea53554)
- Unicast RPF does not provide complete protection against spoofing. Spoofed packets can enter a network through unicast RPF-enabled interfaces if an appropriate return route to the source IP address exists.
- The switch applies the same unicast RPF mode to all interfaces where unicast RPF check is configured. Any change that you make in the unicast RPF mode on any interfaces is applied to all interfaces where the unicast RPF check is configured.
- The "allow default" options of the unicast RPF modes do not offer significant protection against spoofing.
  - Strict unicast RPF Check with Allow Default—Received IP traffic that is sourced from a prefix that exists in the routing table passes the unicast RPF check if the prefix is reachable through the input interface. If a default route is configured, any IP packet with a source prefix that is not in the routing table passes the unicast RPF check if the ingress interface is a reverse path for the default route.
  - Loose unicast RPF Check with Allow Default—If a default route is configured, any IP packet passes the unicast RPF check.
- Avoid configurations that overload the route processor with unicast RPF checks.
  - Do not configure unicast RPF to filter with an ACL.
  - Do not configure the global unicast RPF "punt" check mode.

Unicast RPF Check Configuration Guidelines and Restrictions

Although the software supports up to 16 reverse-path interfaces, implement these limits in your configuration:

- Unicast RPF Strict Mode—The unicast RPF strict mode provides the greatest security against spoofed traffic. If, on all unicast RPF-check enabled interfaces, the switch receives valid IP traffic through interfaces that are reverse paths for the traffic, then strict mode is an option in these circumstances:
  - If, on a maximum of 24 interfaces, divided into four groups of six interfaces each, the switch receives valid IP packets that have up to six reverse-path interfaces per source prefix, configure the unicast RPF strict mode with the **mls ip cef rpf multipath interface-group** command.

    This option requires you to identify the source prefixes and the interfaces that serve as reverse paths for the source prefixes and to configure interface groups for those reverse path interfaces. All of the reverse-path interfaces for each source prefix must be in the same interface group.

You can configure four interface groups, with each group containing up to six reverse-path interfaces. There is no limit on the number of source prefixes that an interface group can support.

To ensure that no more than six reverse-path interfaces exist in the routing table for each prefix, enter the **maximum-paths 6** command in config-router mode when configuring OSPF, EIGRP, or BGP.

IP traffic with one or two reverse-path interfaces that is received on uPPF-check enabled interfaces outside the interface groups passes the unicast RPF check if the ingress interface and at most one other interface are reverse paths.

With maximum paths set to six, IP traffic with more than two reverse-path interfaces that is received on uPPF-check enabled interfaces outside the interface groups always pass the unicast RPF check.

  – If, on any number of interfaces, the switch receives valid IP packets that have one or two reverse path interfaces per source prefix, configure the unicast RPF strict mode with the **mls ip cef rpf multipath pass** command.

    To ensure that no more than two reverse-path interfaces exist in the routing table for each prefix, enter the **maximum-paths 2** command in config-router mode when configuring OSPF, EIGRP, or BGP.

- Unicast RPF Loose Mode with the Pass Global Mode—The unicast RPF loose mode provides less protection than strict mode, but it is an option on switches that receive valid IP traffic on interfaces that are not reverse paths for the traffic. The unicast RPF loose mode verifies that received traffic is sourced from a prefix that exists in the routing table, regardless of the interface on which the traffic arrives.

Configuring Unicast RPF Check

These sections describe how to configure unicast RPF check:

- Configuring the Unicast RPF Check Mode, page 47-4
- Configuring the Multiple-Path Unicast RPF Check Mode, page 47-6
- Enabling Self-Pinging, page 47-7

Configuring the Unicast RPF Check Mode

There are two unicast RPF check modes:

- Strict check mode, which verifies that the source IP address exists in the FIB table and verifies that the source IP address is reachable through the input port.
- Exist-only check mode, which only verifies that the source IP address exists in the FIB table.

Note The most recently configured mode is automatically applied to all ports configured for unicast RPF check.

To configure unicast RPF check mode, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{vlan vlan_ID}	{type[1] slot/port}	{port-channel number}}`	Selects an interface to configure. **Note** Based on the input port, unicast RPF check verifies the best return path before forwarding the packet on to the next destination.
Step 2	`Router(config-if)# ip verify unicast source reachable-via {rx	any} [allow-default] [list]`	Configures the unicast RPF check mode.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.
Step 4	`Router# show mls cef ip rpf`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring the unicast RPF check mode, note the following information:

- Use the **rx** keyword to enable strict check mode.
- Use the **any** keyword to enable exist-only check mode.
- Use the **allow-default** keyword to allow use of the default route for RPF verification.
- Use the *list* option to identify an access list.
  - If the access list denies network access, spoofed packets are dropped at the port.
  - If the access list permits network access, spoofed packets are forwarded to the destination address. Forwarded packets are counted in the interface statistics.
  - If the access list includes the logging action, information about the spoofed packets is sent to the log server.

Note When you enter the **ip verify unicast source reachable-via** command, the unicast RPF check mode changes on all ports in the switch.

This example shows how to enable unicast RPF exist-only check mode on Gigabit Ethernet port 4/1:

```
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip verify unicast source reachable-via any
Router(config-if)# end
Router#
```

This example shows how to enable unicast RPF strict check mode on Gigabit Ethernet port 4/2:

```
Router(config)# interface gigabitethernet 4/2
Router(config-if)# ip verify unicast source reachable-via rx
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show running-config interface gigabitethernet 4/2
Building configuration...
Current configuration : 114 bytes
!
interface GigabitEthernet4/2
ip address 42.0.0.1 255.0.0.0
ip verify unicast reverse-path
no cdp enable
```

```
end
Router# show running-config interface gigabitethernet 4/1
Building configuration...
Current configuration : 114 bytes
!
interface GigabitEthernet4/1
ip address 41.0.0.1 255.0.0.0
➔ ip verify unicast reverse-path (RPF mode on g4/1 also changed to strict-check RPF mode)
no cdp enable
end
Router#
```

Configuring the Multiple-Path Unicast RPF Check Mode

To configure the multiple-path unicast RPF check mode, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls ip cef rpf mpath** {**punt** | **pass** | **interface-group**}	Configures the multiple path RPF check mode.
Step 2	Router(config)# **end**	Exits configuration mode.
Step 3	Router# **show mls cef ip rpf**	Verifies the configuration.

When configuring multiple path RPF check, note the following information:

- **punt** mode (default)—The PFC3 performs the unicast RPF check in hardware for up to two interfaces per prefix. Packets arriving on any additional interfaces are redirected (punted) to the RP for unicast RPF check in software.
- **pass** mode—The PFC3 performs the unicast RPF check in hardware for single-path and two-path prefixes. Unicast RPF check is disabled for packets coming from multipath prefixes with three or more reverse-path interfaces (these packets always pass the unicast RPF check).
- **interface-group** mode—The PFC3 performs the unicast RPF check in hardware for single-path and two-path prefixes. The PFC3 also performs the unicast RPF check for up to four additional interfaces per prefix through user-configured multipath unicast RPF check interface groups. Unicast RPF check is disabled for packets coming from other multipath prefixes that have three or more reverse-path interfaces (these packets always pass the unicast RPF check).

This example shows how to configure punt as the multiple path RPF check mode:

```
Router(config)# mls ip cef rpf mpath punt
```

Configuring Multiple-Path Interface Groups

To configure multiple-path unicast RPF interface groups, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls ip cef rpf interface-group** [**0** | **1** | **2** | **3**] *interface1* [*interface2* [*interface3* [*interface4*]]]	Configures a multiple path RPF interface group.
Step 2	Router(config)# **mls ip cef rpf interface-group** *group_number*	Removes an interface group.

	Command	Purpose
Step 3	`Router(config)# end`	Exits configuration mode.
Step 4	`Router# show mls cef ip rpf`	Verifies the configuration.

This example shows how to configure interface group 2:

```
Router(config)# mls ip cef rpf interface-group 2 fastethernet 3/3 fastethernet 3/4
fastethernet 3/5 fastethernet 3/6
```

Enabling Self-Pinging

With unicast RPF check enabled, by default the switch cannot ping itself.

To enable self-pinging, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{vlan vlan_ID}	{type[1] slot/port}	{port-channel number}}`	Selects the interface to configure.
Step 2	`Router(config-if)# ip verify unicast source reachable-via any allow-self-ping`	Enables the switch to ping itself or a secondary address.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable self-pinging:

```
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip verify unicast source reachable-via any allow-self-ping
Router(config-if)# end
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Using AutoSecure

This chapter describes how to use the AutoSecure function. Release 12.2(33)SXH and later releases support the AutoSecure function.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding AutoSecure

You can easily secure the switch without understanding all the security capabilities of the switch by using the AutoSecure feature. AutoSecure is a simple security configuration process that disables nonessential system services and enables a basic set of recommended security policies to ensure secure networking services.

Caution Although AutoSecure helps to secure a switch, it does not guarantee the complete security of the switch.

The following sections describe how AutoSecure works:

- Securing the Management Plane, page 48-2
- Securing the Forwarding Plane, page 48-5
- AutoSecure Guidelines and Restrictions, page 48-6

Benefits of AutoSecure

AutoSecure provides a variety of mechanisms to enhance security of the switch.

Simplified Switch Security Configuration

AutoSecure automates a thorough configuration of security features of the switch. AutoSecure disables certain features that are enabled by default that could be exploited for security holes.

You can execute AutoSecure in either of two modes, depending on your individual needs:

- Interactive mode—Prompts with options to enable and disable services and other security features, suggesting a default setting for each option.
- Noninteractive mode—Automatically executes the recommended Cisco default settings.

Enhanced Password Security

AutoSecure provides the following mechanisms to improve the security of access to the switch:

- You can specify a required minimum password length, which can eliminate weak passwords that are prevalent on most networks, such as "lab" and "cisco."

  To configure a minimum password length, use the **security passwords min-length** command.
- You can cause a syslog message to be generated after the number of unsuccessful login attempts exceeds the configured threshold.

  To configure the number of allowable unsuccessful login attempts (the threshold rate), use the **security authentication failure rate** command.

System Logging Message Support

System logging messages capture any subsequent changes to the AutoSecure configuration that are applied on the running configuration. As a result, a more detailed audit trail is provided when AutoSecure is executed.

Securing the Management Plane

AutoSecure provides protection for the switch management interfaces (the management plane) and the data routing and switching functions (the forwarding plane, described in the "Securing the Forwarding Plane" section on page 48-5.) Securing the management plane is done by turning off certain global and interface services that can be potentially exploited for security attacks and turning on global services that help minimize the threat of attacks. Secure access and secure logging are also configured for the switch.

Caution If your device is managed by a network management (NM) application, securing the management plane could turn off some services such as the HTTP server and disrupt the NM application support.

The following sections define how AutoSecure helps to secure the management plane:

- Disables Global Services, page 48-3
- Disables Per-Interface Services, page 48-3
- Enables Global Services, page 48-4
- Secures Access to the Switch, page 48-4
- Enhances Logging for Security, page 48-5

Disables Global Services

AutoSecure will disable the following global services on the switch without prompting the user:

- Finger—Collects information about the system (reconnaissance) before an attack.
- PAD—Enables all packet assembler and disassembler (PAD) commands and connections between PAD devices and access servers.
- Small servers—Causes TCP and User Datagram Protocol (UDP) diagnostic port attacks: a sender transmits a volume of fake requests for UDP diagnostic services on the switch, consuming all CPU resources.
- Bootp server—Bootp is an insecure protocol that can be exploited for an attack.
- HTTP server—Without secure-HTTP or authentication embedded in the HTTP server with an associated ACL, the HTTP server is insecure and can be exploited for an attack. (If you must enable the HTTP server, you will be prompted for the proper authentication or access list.)

Note If you are using Security Device Manager (SDM), you must manually enable the HTTP server using the **ip http server** command.

- Identification service—An unsecure protocol (defined in RFC 1413) that allows an external host to query a TCP port for identification. An attacker can access private information about the user from the ID server.
- CDP—If a large number of Cisco Discovery Protocol (CDP) packets are sent to the switch, the available memory of the switch can be consumed, which causes the switch to crash.

Note NM applications that use CDP to discover network topology will not be able to perform discovery.

- NTP—Without authentication or access control, Network Time Protocol (NTP) is insecure and can be used by an attacker to send NTP packets to crash or overload the switch.

  If you require NTP, you must configure NTP authentication using Message Digest 5 (MD5) and the **ntp access-group** command. If NTP is enabled globally, disable it on all interfaces on which it is not needed.
- Source routing—Source routing is provided only for debugging purposes, and should be disabled in all other cases. Otherwise, packets may avoid some of the access control mechanisms of the switch.

Disables Per-Interface Services

AutoSecure will disable the following per-interface services on the switch without prompting the user:

- ICMP redirects—Disabled on all interfaces. Does not add a useful functionality to a correctly configured network, but could be used by attackers to exploit security holes.
- ICMP unreachables—Disabled on all interfaces. Internet Control Management Protocol (ICMP) unreachables are a known method for some ICMP-based denial of service (DoS) attacks.
- ICMP mask reply messages—Disabled on all interfaces. ICMP mask reply messages can give an attacker the subnet mask for a particular subnetwork in the internetwork.
- Proxy-arp—Disabled on all interfaces. Proxy-arp requests are a known method for DoS attacks because the available bandwidth and resources of the switch can be consumed in an attempt to respond to the repeated requests sent by an attacker.
- Directed broadcast—Disabled on all interfaces. Potential cause of SMURF attacks for DoS.
- Maintenance Operations Protocol (MOP) service—Disabled on all interfaces.

Enables Global Services

AutoSecure will enable the following global services on the switch without prompting the user:

- The **service password-encryption** command—Prevents passwords from being visible in the configuration.
- The **service tcp-keepalives-in** and **service tcp-keepalives-out** commands—Ensures that abnormally terminated TCP sessions are removed.

Secures Access to the Switch

Caution If your device is managed by an NM application, securing access to the switch could turn off vital services and may disrupt the NM application support.

AutoSecure will make the following options available for securing access to the switch:

- If a text banner does not exist, you will be prompted to add a banner. This feature provides the following sample banner:

```
Authorized access only
    This system is the property of ABC Enterprise
    Disconnect IMMEDIATELY if you are not an authorized user!
    Contact abc@example.com +1 408 5551212 for help.
```

- The login and password (preferably a secret password, if supported) are configured on the console, AUX, vty, and tty lines. The **transport input** and **transport output** commands are also configured on all of these lines. (Telnet and secure shell (SSH) are the only valid transport methods.) The **exec-timeout** command is configured on the console and AUX as 10.
- When the image on the device is a crypto image, AutoSecure enables SSH and secure copy (SCP) for access and file transfer to and from the switch. The **timeout** *seconds* and **authentication-retries** *integer* options for the **ip ssh** command are configured to a minimum number. (Telnet and FTP are not affected by this operation and remain operational.)
- If the user specifies that the switch does not use Simple Network Management Protocol (SNMP), one of the following functionalities will occur:
  - In interactive mode, the user is asked whether to disable SNMP regardless of the values of the community strings, which act like passwords to regulate access to the agent on the switch.
  - In noninteractive mode, SNMP will be disabled if the community string is public or private.

> **Note** After AutoSecure has been enabled, tools that use SNMP to monitor or configure a device will be unable to communicate with the device using SNMP.

- If authentication, authorization, and accounting (AAA) is not configured, AutoSecure configures local AAA. AutoSecure will prompt the user to configure a local username and password on the switch.

Enhances Logging for Security

AutoSecure provides the following logging options, which allow you to identify and respond to security incidents:

- Sequence numbers and time stamps for all debug and log messages. This option is useful when auditing logging messages.
- Logging messages for login-related events. For example, the message "Blocking Period when Login Attack Detected" will be displayed when a login attack is detected and the switch enters quiet mode. (Quiet mode means that the switch will not allow any login attempts using Telnet, HTTP, or SSH.)
- The **logging console critical** command, which sends system logging (syslog) messages to all available TTY lines and limits messages based on severity.
- The **logging buffered** command, which copies logging messages to an internal buffer and limits messages logged to the buffer based on severity.
- The **logging trap debugging** command, which allows all commands with a severity higher than debugging to be sent to the logging server.

Securing the Forwarding Plane

To minimize the risk of attacks on the switch forwarding plane, AutoSecure provides the following functions:

- Cisco Express Forwarding (CEF)—AutoSecure enables CEF or distributed CEF (dCEF) on the switch whenever possible. Because there is no need to build cache entries when traffic starts arriving for new destinations, CEF operates more predictably than other modes when presented with large volumes of traffic addressed to many destinations. Switches configured for CEF perform better under SYN attacks than switches using the traditional cache.

> **Note** CEF consumes more memory than a traditional cache.

- If strict Unicast Reverse Path Forwarding (uRPF) is available, it can be configured on the switch to help mitigate problems that are caused by the introduction of forged (spoofed) IP source addresses. uRPF discards IP packets that lack a verifiable IP source address.
- Hardware rate limiting—AutoSecure will enable hardware rate-limiting of the following types of traffic without prompting the user:
  - IP errors
  - RPF failures
  - ICMP no-route messages
  - ICMP acl-drop messages

- IPv4 multicast FIB miss messages
- IPv4 multicast partially switch flow messages

AutoSecure will provide the option for hardware rate-limiting of the following types of traffic:

- ICMP redirects
- TTL failures
- MTU failures
- IP unicast options
- IP multicast options
- Ingress and egress ACL bridged packets

Note Rate-limiting of ingress and egress ACL bridged packets can interfere with ACL logging and can increase session setup rates for hardware accelerated features such as NAT, Layer 3 WCCP, and TCP intercept.

AutoSecure Guidelines and Restrictions

When configuring AutoSecure, follow these guidelines and restrictions:

- Because there is no command to undo configuration changes made by AutoSecure, always save your running configuration before configuring AutoSecure.
- The AutoSecure configuration can be configured at run time or setup time. If any related configuration is modified after AutoSecure has been enabled, the AutoSecure configuration may not be fully effective.
- After AutoSecure has been enabled, tools that use SNMP to monitor or configure a device will be unable to communicate with the device using SNMP.
- If your device is managed by a network management (NM) application, securing the management plane could turn off some services such as HTTP server and disrupt the NM application support.
- If you are using Security Device Manager (SDM), you must manually enable the HTTP server using the **ip http server** command.
- NM applications that use CDP to discover network topology will not be able to perform discovery.

Configuring AutoSecure

These sections describe how to configure AutoSecure:

- Using the AutoSecure Command, page 48-7
- Configuring Additional Security, page 48-8
- Verifying AutoSecure, page 48-8

Using the AutoSecure Command

The **auto secure** command guides you through a semi-interactive session (also known as the AutoSecure session) to secure the management and forwarding planes. You can use this command to secure just the management plane or the forwarding plane; if neither option is selected in the command line, you can choose to configure one or both planes during the session.

This command also allows you to go through all noninteractive configuration portions of the session before the interactive portions. The noninteractive portions of the session can be enabled by selecting the optional **no-interact** keyword.

The AutoSecure session will request the following information from you:

- Is the device going to be connected to the Internet?
- How many interfaces are connected to the Internet?
- What are the names of the interfaces connected to the Internet?
- What will be your local username and password?
- What will be the switch hostname and domain name?

At any prompt you may enter a question mark (**?**) for help or Ctrl-C to abort the session.

In interactive mode, you will be asked at the end of the session whether to commit the generated configuration to the running configuration of the switch. In noninteractive mode, the changes will be automatically applied to the running configuration.

Note There is no undo command for configuration changes made by AutoSecure. You should always save the running configuration before executing the **auto secure** command.

To execute the AutoSecure configuration process, beginning in privileged EXEC mode, perform this task:

Command	Purpose
`Router# auto secure [management	forwarding] [no-interact	full]`	Executes the AutoSecure session for configuring one or both planes of the switch. • **management**—Only the management plane will be secured. • **forwarding**—Only the forwarding plane will be secured. • **no-interact**—The user will not be prompted for any interactive configurations. • **full**—The user will be prompted for all interactive questions. This is the default.

For an example of the AutoSecure session, see the "AutoSecure Configuration Example" section on page 48-9.

Configuring Additional Security

After completing the AutoSecure configuration, you can further enhance the security of management access to your switch by performing this task:

	Command or Action	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# security passwords min-length length`	Ensures that all configured passwords are at least a specified length. • *length*—Minimum length of a configured password. The range is 0 to 16 characters.
Step 3	`Router(config)# enable password {password | [encryption-type] password}`	Sets a local password to control access to various privilege levels. • *encryption-type*—A value of 0 indicates that an unencrypted password follows. A value of 7 indicates that a hidden password follows. **Note** You usually will not enter an encryption type unless you enter a password that has already been encrypted by a Cisco router or switch.
Step 4	`Router(config)# security authentication failure rate threshold-rate log`	Configures the number of allowable unsuccessful login attempts. • *threshold-rate*—Number of allowable unsuccessful login attempts. The range is 1 to 1024. • **log**—Syslog authentication failures if the number of failures in one minute exceeds the threshold.

The following example shows how to configure the switch for a minimum password length of 10 characters and a threshold of 3 password failures in one minute. The example also shows how to set a hidden local password.

```
Router# configure terminal
Router(config)# security passwords min-length 10
Router(config)# security authentication failure rate 3
Router(config)# enable password 7 elephant123
```

Verifying AutoSecure

To verify that the AutoSecure feature has executed successfully, perform this task:

Command	Purpose
`Router# show auto secure config`	Displays all configuration commands that have been added as part of the AutoSecure configuration. The output is the same as the configuration generated output

AutoSecure Configuration Example

The following example is a sample AutoSecure session. After you execute the **auto secure** command, the feature will automatically prompt you with a similar response unless you enable the **no-interact** keyword. (For information on which features are disabled and which features are enabled, see the "Securing the Management Plane" section on page 48-2 and the "Securing the Forwarding Plane" section on page 48-5.)

```
Router# auto secure
                --- AutoSecure Configuration ---

*** AutoSecure configuration enhances the security of
the router, but it will not make it absolutely resistant
to all security attacks ***

 AutoSecure will modify the configuration of your device.

All the configuration changes will be shown. For a detailed
explanation of how the configuration changes enhance security
and any possible side effects, please refer to Cisco.com for
AutoSecure documentation.
At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

If this device is being managed by a network management station,
AutoSecure configuration may block network management traffic.
Continue with AutoSecure? [no]: y

Gathering information about the router for AutoSecure

Is this router connected to internet? [no]: y
Enter the number of interfaces facing the internet [1]: 1
Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES NVRAM  administratively down down
Vlan77                 77.1.1.4        YES NVRAM  down                  down
GigabitEthernet6/1     unassigned      YES NVRAM  administratively down down
GigabitEthernet6/2     21.30.30.1      YES NVRAM  up                    up
Loopback0              3.3.3.3         YES NVRAM  up                    up
Tunnel1                unassigned      YES NVRAM  up                    up
Enter the interface name that is facing the internet: Vlan77

Securing Management plane services...

Disabling service finger
Disabling service pad
Disabling udp & tcp small servers
Enabling service password encryption
Enabling service tcp-keepalives-in
Enabling service tcp-keepalives-out
Disabling the cdp protocol

Disabling the bootp server
Disabling the http server
Disabling the finger service
Disabling source routing
Disabling gratuitous arp

Here is a sample Security Banner to be shown
at every access to device. Modify it to suit your
enterprise requirements.

Authorized Access only
```

```
  This system is the property of <Name of Enterprise>.
  UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED.
  You must have explicit permission to access this
  device. All activities performed on this device
  are logged. Any violations of access policy will result
  in disciplinary action.

Enter the security banner {Put the banner between
k and k, where k is any character}:
k
banner
k
Enter the new enable secret:
Confirm the enable secret :
Enable password is not configured or its length
is less than minimum no. of charactersconfigured
Enter the new enable password:
Confirm the enable password:

Configuration of local user database
Enter the username: cisco
Enter the password:
Confirm the password:
Configuring AAA local authentication
Configuring Console, Aux and VTY lines for
local authentication, exec-timeout, and transport
Securing device against Login Attacks
Configure the following parameters

Blocking Period when Login Attack detected (in seconds): 5

Maximum Login failures with the device: 3

Maximum time period for crossing the failed login attempts (in seconds): ?
% A decimal number between 1 and 32767.

Maximum time period for crossing the failed login attempts (in seconds): 5

Configure SSH server? [yes]: no

Configuring interface specific AutoSecure services
Disabling mop on Ethernet interfaces

Securing Forwarding plane services...

Enabling unicast rpf on all interfaces connected
to internet

The following rate-limiters are enabled by default:
  mls rate-limit unicast ip errors 100 10
  mls rate-limit unicast ip rpf-failure 100 10
  mls rate-limit unicast ip icmp unreachable no-route 100 10
  mls rate-limit unicast ip icmp unreachable acl-drop 100 10
  mls rate-limit multicast ipv4 fib-miss 100000 100
  mls rate-limit multicast ipv4 partial 100000 100

Would you like to enable the following rate-limiters also?
mls rate-limit unicast ip icmp redirect 100 10
mls rate-limit all ttl-failure 100 10
mls rate-limit all mtu-failure 100 10
mls rate-limit unicast ip options 100 10
mls rate-limit multicast ipv4 ip-options 100 10

Enable the above rate-limiters also? [yes/no]: yes
```

```
Would you like to enable the rate-limiters for Ingress/EgressACL bridged packets also?
NOTE: Enabling the ACL in/out rate-limiters can affect ACL logging
      and session setup rate for hardware accelerated features such
      as NAT, Layer 3 WCCP and TCP Intercept
mls rate-limit unicast acl input 100 10
mls rate-limit unicast acl output 100 10

Enable the ACL in/out rate-limiters also? [yes/no]: no

This is the configuration generated:

no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
service password-encryption
service tcp-keepalives-in
service tcp-keepalives-out
no cdp run
no ip bootp server
no ip http server
no ip finger
no ip source-route
no ip gratuitous-arps
no ip identd
banner k
banner
k
security passwords min-length 6
security authentication failure rate 10 log
enable secret 5 $1$30kP$f.KDndYPz/Hv/.yTlJStN/
enable password 7 08204E4D0D48574446
username cisco password 7 08204E4D0D48574446
aaa new-model
aaa authentication login local_auth local
line console 0
 login authentication local_auth
 exec-timeout 5 0
 transport output telnet
line vty 0 15
 login authentication local_auth
 transport input telnet
login block-for 5 attempts 3 within 5
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
logging facility local2
logging trap debugging
service sequence-numbers
logging console critical
logging buffered
int Vlan1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
int Vlan77
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
```

```
 no mop enabled
int GigabitEthernet6/1
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
int GigabitEthernet6/2
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
interface Vlan77
 ip verify unicast source reachable-via rx
mls rate-limit unicast ip icmp redirect 100 10
mls rate-limit all ttl-failure 100 10
mls rate-limit all mtu-failure 100 10
mls rate-limit unicast ip options 100 10
mls rate-limit multicast ipv4 ip-options 100 10
!
end


Apply this configuration to running-config? [yes]: yes

Applying the config generated to running-config

Router#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Understanding Cisco IOS ACL Support

This chapter describes Cisco IOS access control list (ACL) support in Cisco IOS Release 12.2SX:

Note For complete information about configuring Cisco IOS ACLs, see the *Cisco IOS Security Configuration Guide*, Release 12.2, "Traffic Filtering and Firewalls," at this URL:

http://www.cisco.com/en/US/docs/ios-xml/ios/sec_data_acl/configuration/12-2sx/sec-data-acl-12-2sx-book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

ACL Support in Hardware and Software

ACLs can be processed in hardware by the Policy Feature Card (PFC), a Distributed Forwarding Card (DFC), or in software by the route processor (RP):

- ACL flows that match a "deny" statement in standard and extended ACLs (input and output) are dropped in hardware if "ip unreachables" is disabled.
- ACL flows that match a "permit" statement in standard and extended ACLs (input and output) are processed in hardware.

- VLAN ACL (VACL) and port ACL (PACL) flows are processed in hardware. If a field that is specified in a VACL or PACL is not supported by hardware processing, then that field is ignored (for example, the **log** keyword in an ACL), or the whole configuration is rejected (for example, a VACL containing IPX ACL parameters).
- VACL logging is processed in software.
- Dynamic ACL flows are processed in hardware.
- Idle timeout is processed in software.

Note Idle timeout is not configurable. Cisco IOS Release 12.2SX does not support the **access-enable host timeout** command.

- Except on MPLS interfaces, reflexive ACL flows are processed in hardware after the first packet in a session is processed in software on the RP.
- IP accounting for an ACL access violation on a given port is supported by forwarding all denied packets for that port to the RP for software processing without impacting other flows.
- The PFC does not provide hardware support for Cisco IOS IPX ACLs. Cisco IOS IPX ACLs are supported in software on the RP.
- Extended name-based MAC address ACLs are supported in hardware.
- The following ACL types are processed in software:
  - Internetwork Packet Exchange (IPX) access lists
  - Standard XNS access list
  - Extended XNS access list
  - DECnet access list
  - Extended MAC address access list
  - Protocol type-code access list

Note IP packets with a header length of less than five will not be access controlled.

- Unless you configure optimized ACL logging (OAL), flows that require logging are processed in software without impacting nonlogged flow processing in hardware (see the "Optimized ACL Logging" section on page 49-8).
- The forwarding rate for software-processed flows is substantially less than for hardware-processed flows.
- When you enter the **show ip access-list** command, the match count displayed does not include packets processed in hardware.
- Hardware-supported counters for hardware-supported ACLs, displayed by the **show tcam interface** command (not supported in PFC3A mode). See this publication:

  http://www.cisco.com/en/US/docs/ios-xml/ios/interface/command/ir-s6.html#GUID-1D17939B-1C8F-422C-83CE-64B096DAD13D
- When you enter the **show policy-map interface** command, sometimes the counters that are displayed do not include all of the hardware switching platform counters.

Cisco IOS ACL Configuration Guidelines and Restrictions

The following guidelines and restrictions apply to Cisco IOS ACLs configured for use with any feature:

- You can apply Cisco IOS ACLs directly to Layer 3 ports and to VLAN interfaces.
- You can apply VLAN ACLs and port ACLs to Layer 2 interfaces (see Chapter 51, "Configuring Port ACLs and VLAN ACLs").
- Each type of ACL (IP, IPX, and MAC) filters only traffic of the corresponding type. A Cisco IOS MAC ACL never matches IP or IPX traffic.
- The PFC does not provide hardware support for Cisco IOS IPX ACLs. Cisco IOS IPX ACLs are supported in software on the route processor (RP).
- By default, the RP sends Internet Control Message Protocol (ICMP) unreachable messages when a packet is denied by an access group.

  With the **ip unreachables** command enabled (which is the default), the switch processor (SP) drops most of the denied packets in hardware and sends only a small number of packets to the RP to be dropped (10 packets per second, maximum), which generates ICMP-unreachable messages.

  To eliminate the load imposed on the RP CPU by the task of dropping denied packets and generating ICMP-unreachable messages, you can enter the **no ip unreachables** interface configuration command to disable ICMP unreachable messages, which allows all access group-denied packets to be dropped in hardware.
- ICMP unreachable messages are not sent if a packet is denied by a VACL or a PACL.
- Use named ACLs (instead of numbered ACLs) because this causes less CPU usage when creating or modifying ACL configurations and during system restarts. When you create ACL entries (or modify existing ACL entries), the software performs a CPU-intensive operation called an ACL merge to load the ACL configurations into the PFC hardware. An ACL merge also occurs when the startup configuration is applied during a system restart.

  With named ACLs, the ACL merge is triggered only when the user exits the **named-acl** configuration mode. However, with numbered ACLs, the ACL merge is triggered for every ACE definition and results in a number of intermediate merges during ACL configuration.

Policy-Based ACLs

Release 12.2(33)SXH and later releases support policy-based ACLs (PBACLs). The following sections describe PBACLs:

- Understanding PBACLs, page 49-3
- PBACL Guidelines and Restrictions, page 49-4
- Configuring PBACL, page 49-4

Understanding PBACLs

PBACLs provide the capability to apply access control policies across object groups. An object group is a group of users or servers.

You define an object group as a group of IP addresses or as a group of protocol ports. You then create access control entries (ACEs) that apply a policy (such as permit or deny) to the object group. For example, a policy-based ACE can permit a group of users to access a group of servers.

An ACE defined using a group name is equivalent to multiple ACEs (one applied to each entry in the object group). The system expands the PBACL ACE into multiple Cisco IOS ACEs (one ACE for each entry in the group) and populates the ACEs in the TCAM. Therefore, the PBACL feature reduces the number of entries you need to configure but does not reduce TCAM usage.

If you make changes in group membership, or change the contents of an ACE that uses an access group, the system updates the ACEs in the TCAM. The following types of changes trigger the update:

- Adding a member to a group
- Deleting a member from a group
- Modifying the policy statements in an ACE that uses an access group

You configure a PBACL using extended Cisco IOS ACL configuration commands. As with regular ACEs, you can associate the same access policy with one or more interfaces.

When you configure an ACE, you can use an object group to define the source, the destination, or both.

PBACL Guidelines and Restrictions

When configuring PBACLs, note the following guidelines and restrictions:

- PBACLs are supported on Layer 3 interfaces (such as routed interfaces and VLAN interfaces).
- The PBACL feature only supports IPv4 ACEs.
- The PBACL feature supports only Cisco IOS ACLs. It is not supported with any other features. The keywords **reflexive** and **evaluate** are not supported.
- The PBACL feature supports only named Cisco IOS ACLs. It does not support numbered ACLs.
- Feature interactions for policy-based ACLs are the same as for Cisco IOS ACLs.

Configuring PBACL

Configure PBACLs using the following tasks:

- Configuring a PBACL IP Address Object Group, page 49-4
- Configuring a PBACL Protocol Port Object Group, page 49-5
- Configuring ACLs with PBACL Object Groups, page 49-5
- Configuring PBACL on an Interface, page 49-6

Configuring a PBACL IP Address Object Group

To create or modify a PBACL IP address object group, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`object-group ip address`** *`object_group_name`*	Defines object group name and enters IP-address object-group configuration mode.
Step 2	`Router(config-ipaddr-ogroup)# {`*`ip_address mask`*`} | {`**`host`** `{`*`name`* `|` *`ip_address`*`} }`	Configures a member of the group. The member is either a network address plus mask or a host (identified by host name or IP address).

	Command	Purpose
Step 3	Router(config-ipaddr-ogroup)# {**end**} | {**exit**}	To leave the configuration mode, enter the **end** command. To leave the IP-address object-group configuration mode, enter the **exit** command.
Step 4	Router# **show object-group** [*object_group_name*]	Displays the object-group configuration for the named group (or for all groups if no name is entered).

The following example creates an object group with three hosts and a network address:

```
Router(config)# object-group ip address myAG
Router(config-ipaddr-pgroup)# host 10.20.20.1
Router(config-ipaddr-pgroup)# host 10.20.20.5
Router(config-ipaddr-pgroup)# 10.30.0.0 255.255.0.0
```

Configuring a PBACL Protocol Port Object Group

To create or modify a PBACL protocol port object group, perform this task:

	Command	Purpose
Step 1	Router(config)# **object-group ip port** *object_group_name*	Defines object group name and enters port object-group configuration mode.
Step 2	Router(config-port-ogroup)# {**eq** *number*} | {**gt** *number*} | {**lt** *number*} | {**neq** *number*} | {**range** *number number*}	Configures a member of the group. The member is either equal to or not equal to a port number, less than or greater than a port number, or a range of port numbers.
Step 3	Router(config-port-ogroup)# **end** | **exit**	To leave the configuration mode, enter the **end** command. To leave the port object-group configuration mode, enter the **exit** command.
Step 4	Router# **show object-group** [*object_group_name*]	Displays the object-group configuration for the named group (or for all groups if no name is entered).

The following example creates a port object group that matches protocol port 100 and any port greater than 200, except 300:

```
Router(config)# object-group ip port myPG
Router(config-port-pgroup)# eq 100
Router(config-port-pgroup)# gt 200
Router(config-port-pgroup)# neq 300
```

Configuring ACLs with PBACL Object Groups

To configure an ACL to use a PBACL object group, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip access-list extended acl_name`	Defines an extended ACL with the specified name and enters extended-ACL configuration mode.
Step 2	`Router(config-ext-nacl)# permit tcp addrgroup object_group_name addrgroup object_group_name`	Configures an ACE for TCP traffic using IP address object group as the source policy and an object group as the destination policy.
Step 3	`Router(config-ext-nacl)# exit`	Exits extended ACL configuration mode.
Step 4	`Router# show access-lists [acl_name]`	Displays the object-group configuration for the named group (or for all groups if no name is entered).

The following example creates an access list that permits packets from the users in myAG if the protocol ports match the ports specified in myPG:

```
Router(config)# ip access-list extended my-pbacl-policy
Router(config-ext-nacl)# permit tcp addrgroup myAG portgroup myPG any
Router(config-ext-nacl)# deny tcp any any
Router(config-ext-nacl)# exit
Router(config)# exit
Router# show ip access-list my-pbacl-policy
Extended IP access list my-pbacl-policy
10  permit tcp addrgroup AG portgroup PG any
20  permit tcp any any

Router# show ip access-list my-pbacl-policy expand
Extended IP access list my-pbacl-policy expanded
20 permit tcp host 10.20.20.1 eq 100 any
20 permit tcp host 10.20.20.1 gt 200 any
20 permit tcp host 10.20.20.1 neq 300 any
20 permit tcp host 10.20.20.5 eq 100 any
20 permit tcp host 10.20.20.5 gt 200 any
20 permit tcp host 10.20.20.5 neq 300 any
20 permit tcp 10.30.0.0 255.255.0.0 eq 100 any
20 permit tcp 10.30.0.0 255.255.0.0 gt 200 any
20 permit tcp 10.30.0.0 255.255.0.0 neq 300 any
```

Configuring PBACL on an Interface

To configure a PBACL on an interface, use the **ip access-group** command. The command syntax and usage is the same as for Cisco IOS ACLs. For additional information, see the "Cisco IOS ACL Configuration Guidelines and Restrictions" section on page 49-3.

The following example shows how to associate access list my-pbacl-policy with VLAN 100:

```
Router(config)# int vlan 100
Router(config-if)# ip access-group mp-pbacl-policy in
```

Configuring IPv6 Address Compression

ACLs are implemented in hardware in the Policy Feature Card (PFC), which uses the source or destination IP address and port number in the packet to index the ACL table. The index has a maximum address length of 128 bits.

The IP address field in an IPv6 packet is 128 bits, and the port field is 16 bits. To use full IPv6 addresses in the ACL hardware table, you can turn on compression of IPv6 addresses using the **mls ipv6 acl compress address unicast** command. This feature compresses the IPv6 address (including port) into 128 bits by removing 16 unused bits from the IPv6 address. Compressible address types can be compressed without losing any information. See Table 49-1 for details about the compression methods.

By default, the command is set for no compression.

Caution Do not enable the compression mode if you have noncompressible address types in your network. A list of compressible address types and the address compression method are listed in Table 49-1.

Table 49-1 Compressible Address Types and Methods

Address Type	Compression Method
EUI-64 based on MAC address	This address is compressed by removing 16 bits from bit locations [39:24]. No information is lost when the hardware compresses these addresses.
Embedded IPv4 address	This address is compressed by removing the upper 16 bits. No information is lost when the hardware compresses these addresses.
Link Local	These addresses are compressed by removing the zeros in bits [95:80] and are identified using the same packet type as the embedded IPv4 address. No information is lost when the hardware compresses these addresses.
Others	If the IPv6 address does not fall into any of the above categories, it is classified as other. If the IPv6 address is classified as other, the following occurs: • If the compress mode is on, the IPv6 address is compressed similarly to the EUI-64 compression method (removal of bits [39:24]) to allow for the Layer 4 port information to be used as part of the key used to look up the QoS TCAM, but Layer 3 information is lost. • If the global compression mode is off, the entire 128 bits of the IPv6 address are used. The Layer 4 port information cannot be included in the key to look up the QoS TCAM because of the size constraints on the IPv6 lookup key.

To turn on the compression of IPv6 addresses, enter the **mls ipv6 acl compress address unicast** command. To turn off the compression of IPv6 addresses, enter the **no** form of this command.

This example shows how to turn on address compression for IPv6 addresses:

```
Router(config)# mls ipv6 acl compress address unicast
Router(config)#
```

This example shows how to turn off address compression for IPv6 addresses:

```
Router(config)# no mls ipv6 acl compress address unicast
Router(config)#
```

Optimized ACL Logging

These sections describe optimized ACL logging (OAL):

- Understanding OAL, page 49-8
- OAL Guidelines and Restrictions, page 49-8
- Configuring OAL, page 49-8

Understanding OAL

OAL provides hardware support for ACL logging. Unless you configure OAL, packets that require logging are processed completely in software on the RP. OAL permits or drops packets in hardware on the PFC3 and uses an optimized routine to send information to the RP to generate the logging messages.

OAL Guidelines and Restrictions

The following guidelines and restrictions apply to OAL:

- OAL and VACL capture are incompatible. Do not configure both features on the switch. With OAL configured, use SPAN to capture traffic.
- OAL is supported only on the PFC3.
- OAL supports only IPv4 unicast packets.
- OAL supports VACL logging of permitted ingress traffic.
- OAL does not support port ACLs (PACLs).
- OAL does not provide hardware support for the following:
  - Reflexive ACLs
  - ACLs used to filter traffic for other features (for example, QoS)
  - ACLs for unicast reverse path forwarding (uRPF) check exceptions
  - Exception packets (for example, TTL failure and MTU failure)
  - Packets with IP options
  - Packets addressed at Layer 3 to the router
  - Packets sent to the RP to generate ICMP unreachable messages
  - Packets being processed by features not accelerated in hardware
- To provide OAL support for denied packets, enter the **mls rate-limit unicast ip icmp unreachable acl-drop 0** command.
- OAL and the **mls verify ip length minimum** command are incompatible. Do not configure both.

Configuring OAL

These sections describe how to configure OAL:

- Configuring OAL Global Parameters, page 49-9
- Configuring OAL on an Interface, page 49-10

- Displaying OAL Information, page 49-10
- Clearing Cached OAL Entries, page 49-10

Note

- For complete syntax and usage information for the commands used in this section, see the Cisco IOS Master Command List.
- To provide OAL support for denied packets, enter the **mls rate-limit unicast ip icmp unreachable acl-drop 0** command.

Configuring OAL Global Parameters

To configure global OAL parameters, perform this task:

Command	Purpose
`Router(config)#` **`logging ip access-list cache`** `{{`**`entries`** *`number_of_entries`*`}	{`**`interval`** *`seconds`*`}	{`**`rate-limit`** *`number_of_packets`*`}	{`**`threshold`** *`number_of_packets`*`}}`	Sets OAL global parameters.

When configuring OAL global parameters, note the following information:

- **entries** *number_of_entries*
  - Sets the maximum number of entries cached.
  - Range: 0–1,048,576 (entered without commas).
  - Default: 8192.
- **interval** *seconds*
  - Sets the maximum time interval before an entry is sent to be logged. Also if the entry is inactive for this duration it is removed from the cache.
  - Range: 5–86,400 (1440 minutes or 24 hours, entered without commas).
  - Default: 300 seconds (5 minutes).
- **rate-limit** *number_of_packets*
  - Sets the number of packets logged per second in software.
  - Range: 10–1,000,000 (entered without commas).
  - Default: 0 (rate limiting is off and all packets are logged).
- **threshold** *number_of_packets*
  - Sets the number of packet matches before an entry is logged.
  - Range: 1–1,000,000 (entered without commas).
  - Default: 0 (logging is not triggered by the number of packet matches).

Configuring OAL on an Interface

To configure OAL on an interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`*`type`*[1] *`slot/port`*`}`	Specifies the interface to configure.
Step 2	`Router(config-if)#` **`logging ip access-list cache in`**	Enables OAL for ingress traffic on the interface.
Step 3	`Router(config-if)#` **`logging ip access-list cache out`**	Enables OAL for egress traffic on the interface.

1. *type* = any that supports Layer 3-switched traffic.

Displaying OAL Information

To display OAL information, perform this task:

Command	Purpose
`Router #` **`show logging ip access-list cache`**	Displays OAL information.

Clearing Cached OAL Entries

To clear cached OAL entries, perform this task:

Command	Purpose
`Router #` **`clear logging ip access-list cache`**	Clears cached OAL entries.

Guidelines and Restrictions for Using Layer 4 Operators in ACLs

These sections describe guidelines and restrictions when configuring ACLs that include Layer 4 port operations:

- Determining Layer 4 Operation Usage, page 49-10
- Determining Logical Operation Unit Usage, page 49-11

Determining Layer 4 Operation Usage

You can specify these types of operations:

- gt (greater than)
- lt (less than)
- neq (not equal)
- eq (equal)
- range (inclusive range)

We recommend that you do not specify more than *nine different* operations on the same ACL. If you exceed this number, each new operation might cause the affected ACE to be translated into more than one ACE.

Use the following two guidelines to determine Layer 4 operation usage:

- Layer 4 operations are considered different if the operator or the operand differ. For example, in this ACL there are three different Layer 4 operations ("gt 10" and "gt 11" are considered two different Layer 4 operations):

```
... gt 10 permit
... lt 9 deny
... gt 11 deny
```

Note There is no limit to the use of "eq" operators as the "eq" operator does not use a logical operator unit (LOU) or a Layer 4 operation bit. See the "Determining Logical Operation Unit Usage" section on page 49-11 for a description of LOUs.

- Layer 4 operations are considered different if the same operator/operand couple applies once to a source port and once to a destination port. For example, in this ACL there are two different Layer 4 operations because one ACE applies to the source port and one applies to the destination port.

```
... Src gt 10 ...
... Dst gt 10
```

Determining Logical Operation Unit Usage

Logical operation units (LOUs) are registers that store operator-operand couples. All ACLs use LOUs. There can be up to 32 LOUs; each LOU can store two different operator-operand couples with the exception of the range operator. LOU usage per Layer 4 operation is as follows:

- gt uses 1/2 LOU
- lt uses 1/2 LOU
- neq uses 1/2 LOU
- range uses 1 LOU
- eq does not require a LOU

For example, this ACL would use a single LOU to store two different operator-operand couples:

```
... Src gt 10 ...
... Dst gt 10
```

A more detailed example follows:

```
ACL1
... (dst port) gt 10 permit
... (dst port) lt 9 deny
... (dst port) gt 11 deny
... (dst port) neq 6 permit
... (src port) neq 6 deny
... (dst port) gt 10 deny

ACL2
... (dst port) gt 20 deny
... (src port) lt 9 deny
... (src port) range 11 13 deny
... (dst port) neq 6 permit
```

The Layer 4 operations and LOU usage is as follows:

- ACL1 Layer 4 operations: 5
- ACL2 Layer 4 operations: 4
- LOUs: 4

An explanation of the LOU usage follows:

- LOU 1 stores "gt 10" and "lt 9"
- LOU 2 stores "gt 11" and "neq 6"
- LOU 3 stores "gt 20" (with space for one more)
- LOU 4 stores "range 11 13" (range needs the entire LOU)

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Cisco TrustSec

Cisco TrustSec is an umbrella term for security improvements to Cisco network devices based on the capability to strongly identify users, hosts and network devices within a network. TrustSec provides topology independent and scalable access controls by uniquely classifying data traffic for a particular role. TrustSec ensures data confidentiality and integrity by establishing trust among authenticated peer and encrypting links with those peers.

To configure Cisco Trustsec on the Cisco Catalyst 6500 Series switches, see the publication, "*Cisco TrustSec Switch Configuration Guide*" at the following URL:

http://www.cisco.com/en/US/docs/switches/lan/trustsec/configuration/guide/trustsec.html

Release Notes for Cisco TrustSec 1.0 General Availability 2010 Release are at the following URL:

http://www.cisco.com/en/US/docs/switches/lan/trustsec/release/notes/cts1_0.html

Additional information on the Cisco TrustSec Solution, including overviews, datasheets, and case studies, is available at:

http://www.cisco.com/en/US/netsol/ns1051/index.html

Table 1 lists the TrustSec features to be eventually implemented on TrustSec-enabled network devices. Successive general availability releases of TrustSec will expand the number of network devices supported and the number of TrustSec features supported per device. See the section, "Hardware Supported" for information on which TrustSec features are implemented.

Table 1 Cisco TrustSec Key Features—TrustSec 1.0 General Availability 2010 Release

Cisco TrustSec Feature	Description
802.1AE Tagging (MACSec)	Protocol for IEEE 802.1AE-based wire-rate hop-to-hop layer 2 encryption. Between MACSec-capable devices, packets are encrypted on egress from the transmitting device, decrypted on ingress to the receiving device, and in the clear within the devices. This feature is only available between TrustSec hardware-capable devices.
Endpoint Admission Control (EAC)	EAC is an authentication process for an endpoint user or a device connecting to the TrustSec domain. Usually EAC takes place at the access level switch. Successful authentication and authorization in the EAC process results in Security Group Tag assignment for the user or device. Currently EAC can be 802.1X, MAC Authentication Bypass (MAB), and Web Authentication Proxy (WebAuth).
Network Device Admission Control (NDAC)	NDAC is an authentication process where each network device in the TrustSec domain can verify the credentials and trustworthiness of its peer device. NDAC utilizes an authentication framework based on IEEE 802.1X port-based authentication and uses EAP-FAST as its EAP method. Successful authentication and authorization in NDAC process results in Security Association Protocol negotiation for IEEE 802.1AE encryption.
Security Group Access Control List (SGACL)	A Security Group Access Control List (SGACL) associates a Security Group Tag with a policy. The policy is enforced upon SGT-tagged traffic egressing the TrustSec domain.
Security Association Protocol (SAP)	After NDAC authentication, the Security Association Protocol (SAP) automatically negotiates keys and the cipher suite for subsequent MACSec link encryption between TrustSec peers. SAP is defined in IEEE 802.11i.
Security Group Tag (SGT)	An SGT is a 16-bit single label indicating the security classification of a source in the TrustSec domain. It is appended to an Ethernet frame or an IP packet.
SGT Exchange Protocol (SXP)	Security Group Tag Exchange Protocol (SXP). Devices that are not TrustSec-hardware capable can, with SXP, receive from the Cisco ACS, SGT attributes for authenticated users or devices then forward the sourceIP-to-SGT binding to a TrustSec-hardware capable device for tagging and SGACL enforcement.

Hardware Supported

Table 2 lists the TrustSec features supported by platform on the release date of Cisco IOS 12.2(33) SXI4.

Table 2 **Feature and Platform support—TrustSec 1.0 General Availability 2010 Release**

Hardware	Software Release	TrustSec Feature Introduced
Catalyst 3560 Series	Cisco IOS 12.2 (53) SE	EAC; SXP
Catalyst 3750 Series	Cisco IOS 12.2 (53) SE	EAC; SXP
Catalyst 4500 Series	Cisco IOS 12.2 (50) SG5	EAC; SXP
Catalyst 6500 Series	Cisco IOS 12.2(33) SXI3[1]	EAC; SXP; NDAC (no SAP)
Nexus 7000 Series	Cisco NX-OS 4.2.1	EAC; SXP; NDAC; SGACL; MACSec

1. Cisco TrustSec was implemented on the Catalyst 6500 Series in SXI3, but announced as generally available in SXI4.

Configuring Port ACLs and VLAN ACLs

This chapter describes how to configure port ACLs (PACLs) and VLAN ACLs (VACLs) in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- Optimized ACL logging (OAL) and VACL capture are incompatible. Do not configure both features on the switch. With OAL configured (see the "Optimized ACL Logging" section on page 49-8), use SPAN to capture traffic.
- Port ACLs do not support the access-list keywords **log** or **reflexive**. These keywords in the access list are ignored. OAL does not support PACLs.
- PACLs are not supported on private VLANs.

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding ACLs

The following sections describe ACLs in Cisco IOS Release 12.2SX:

Understanding ACLs

Access control lists (ACLs) provide the ability to filter ingress and egress traffic based on conditions specified in the ACL.

Cisco IOS Release 12.2SX supports the following types of ACLs:

- Cisco IOS ACLs are applied to Layer 3 interfaces. They filter traffic routed between VLANs. For more information about Cisco IOS ACLs, see Chapter 49, "Understanding Cisco IOS ACL Support."
- VACLs control access to the VLAN of all packets (bridged and routed). Packets can either enter the VLAN through a Layer 2 port or through a Layer 3 port after being routed. You can also use VACLs to filter traffic between devices in the same VLAN.
- Port ACLs perform access control on all traffic entering the specified Layer 2 port.

PACLs and VACLs can provide access control based on the Layer 3 addresses (for IP protocols) or Layer 2 MAC addresses (for non-IP protocols).

You can apply only one IP access list and one MAC access list to a Layer 2 interface.

Understanding VACLs

VLAN ACLs (VACLs) can provide access control for all packets that are bridged within a VLAN or that are routed into or out of a VLAN or a WAN interface for VACL capture. Unlike Cisco IOS ACLs that are applied on routed packets only, VACLs apply to all packets and can be applied to any VLAN or WAN interface. VACLs are processed in the ACL TCAM hardware. VACLs ignore any Cisco IOS ACL fields that are not supported in hardware.

You can configure VACLs for IP and MAC-layer traffic. VACLs applied to WAN interfaces support only IP traffic for VACL capture.

If a VACL is configured for a packet type, and a packet of that type does not match the VACL, the default action is to deny the packet.

Note IGMP packets are not checked against VACLs.

MAC Policy-Based Forwarding

Cisco IOS Release 12.2(33)SXI and later releases support MAC Policy-Based Forwarding (PBF), a type of MAC-based VACL by which packets can be bridged between VLANs. MAC PBF forwards packets based solely on the source and destination MAC addresses, ignoring any information above Layer 2. Unlike other VACLs, which are processed in the ACL TCAM hardware, MAC PBF is performed in software, with optional rate limiters to control CPU usage. Also, PBF is applied only to incoming packets.

Note Layer 2 port ACLs (PACLs) take precedence over MAC PBF.

Understanding Port ACLs

The port ACL (PACL) feature provides the ability to perform access control on specific Layer 2 ports. A Layer 2 port is a physical LAN or trunk port that belongs to a VLAN. Port ACLs are applied only on the ingress traffic. The port ACL feature is supported only in hardware (port ACLs are not applied to any packets routed in software).

When you create a port ACL, an entry is created in the ACL TCAM. You can use the **show tcam counts** command to see how much TCAM space is available.

The PACL feature does not affect Layer 2 control packets received on the port.

You can use the **access-group mode** command to change the way that PACLs interact with other ACLs.

PACLs use the following modes:

- Prefer port mode—If a PACL is configured on a Layer 2 interface, the PACL takes effect and overwrites the effect of other ACLs (Cisco IOS ACL and VACL). If no PACL feature is configured on the Layer 2 interface, other features applicable to the interface are merged and are applied on the interface.
- Merge mode—In this mode, the PACL, VACL, and Cisco IOS ACLs are merged in the ingress direction following the logical serial model shown in Figure 51-2. This is the default access group mode.

You configure the **access-group mode** command on each interface. The default is merge mode.

Note A PACL can be configured on a trunk port only after prefer port mode has been selected. Trunk ports do not support merge mode.

To illustrate access group mode, assume a physical port belongs to VLAN100, and the following ACLs are configured:

- Cisco IOS ACL R1 is applied on routed interface VLAN100.
- VACL (VLAN filter) V1 is applied on VLAN100.
- PACL P1 is applied on the physical port.

In this situation, the following ACL interactions occur:

- In prefer port mode, Cisco IOS ACL R1 and VACL V1 are ignored.
- In merge mode, Cisco IOS ACL R1, VACL V1 and PACL P1 are merged and applied on the port.

Note The CLI syntax for creating a PACL is identical to the syntax for creating a Cisco IOS ACL. An instance of an ACL that is mapped to a Layer 2 port is called a PACL. An instance of an ACL that is mapped to a Layer 3 interface is called a Cisco IOS ACL. The same ACL can be mapped to both a Layer 2 port and a Layer 3 interface.

The PACL feature supports MAC ACLs and IPv4 ACLs. The PACL feature does not support ACLs for IPV6, ARP, or Multiprotocol Label Switching (MPLS) traffic.

PACLs are explained in more detail in the following sections:

EtherChannel and PACL Interactions

This section describes the guidelines for the EtherChannel and PACL interactions:

- PACLs are supported on the main Layer 2 channel interface but not on the port members. A port that has a PACL configured on it may not be configured as an EtherChannel member port. The EtherChannel configuration commands are unavailable on ports that are configured with a PACL.
- Changing the configuration on the logical port affects all the ports in the channel. When an ACL is mapped to the logical port belonging to a channel, it is mapped to all ports in the channel.

Dynamic ACLs (Applies to Merge Mode Only)

Dynamic ACLs are VLAN-based and are used by GWIP. The merge mode *does not* support the merging of the dynamic ACLs with the PACLs. In merge mode, the following configurations are not allowed:

- Attempting to apply a PACL on a port where its corresponding VLAN has a dynamic ACL mapped. In this case, the PACL is not applied to traffic on the port.
- Configuring a dynamic ACL on a VLAN where one of its constituent ports has a PACL installed. In this case, the dynamic ACL is not applied.

Trunk Ports

To configure a PACL on a trunk port, you must first configure port prefer mode. The configuration commands to apply a PACL on a trunk or dynamic port will not be available until you configure the port in port prefer mode by entering the **access-group mode prefer port** interface command. Trunk ports do not support merge mode.

Layer 2 to Layer 3 Port Conversion

If you reconfigure a port from Layer 2 to Layer 3, any PACL configured on the port becomes inactive but remains in the configuration. If you subsequently configure the port as Layer 2, any PACL configured on the port becomes active again.

Port-VLAN Association Changes

You can enter port configuration commands that alter the port-VLAN association, which triggers an ACL remerge.

Unmapping and then mapping a PACL, VACL, or Cisco IOS ACL automatically triggers a remerge.

In merge mode, online insertion or removal of a switching module also triggers a remerge, if ports on the module have PACLs configured.

PACL and VACL Interactions

The following sections describe interactions between the different types of ACL:

PACL Interaction with VACLs and Cisco IOS ACLs

This section describes the guidelines for the PACL interaction with the VACLs and Cisco IOS ACLs.

For an incoming packet on a physical port, the PACL is applied first. If the packet is permitted by the PACL, the VACL on the ingress VLAN is applied next. If the packet is Layer 3 forwarded and is permitted by the VACL, it is filtered by the Cisco IOS ACL on the same VLAN. The same process happens in reverse in the egress direction. However, there is currently no hardware support for output PACLs.

The PACLs override both the VACLs and Cisco IOS ACLs when the port is configured in prefer port mode. The one exception to this rule is when the packets are forwarded in the software by the route processor (RP). The RP applies the ingress Cisco IOS ACL regardless of the PACL mode. Two examples where the packets are forwarded in the software are as follows:

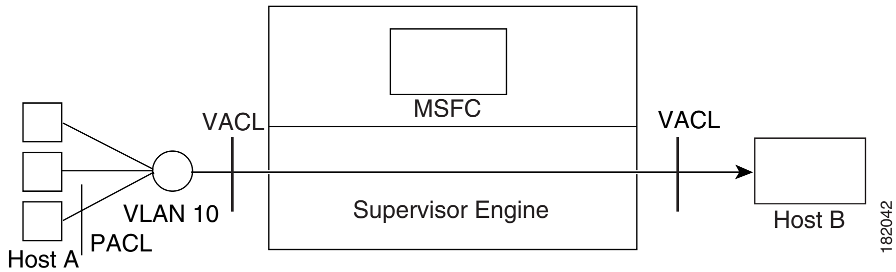
- Packets that are egress bridged (due to logging or features such as NAT)
- Packets with IP options

Bridged Packets

Figure 51-1 shows a PACL and a VACL applied to bridged packets. In merge mode, the ACLs are applied in the following order:

1. PACL for the ingress port
2. VACL for the ingress VLAN
3. VACL for the egress VLAN

Figure 51-1 Applying ACLs on Bridged Packets



In prefer port mode, only the PACL is applied to the ingress packets (the input VACL is not applied).

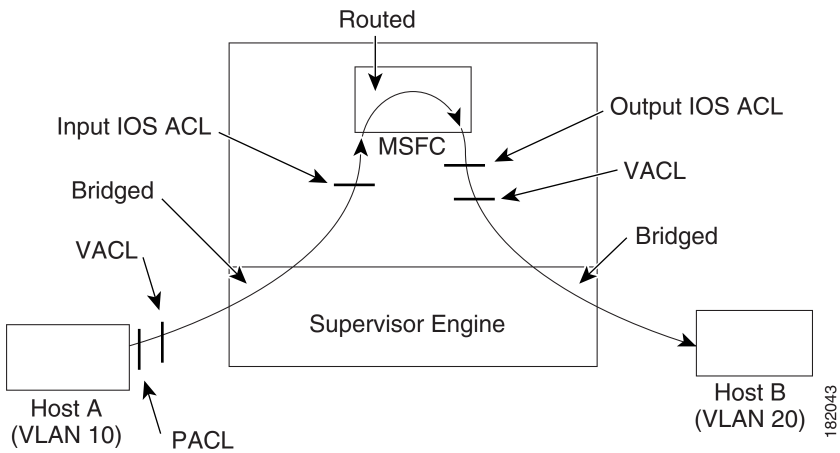
Routed Packets

Figure 51-2 shows how ACLs are applied on routed and Layer 3-switched packets. In merge mode, the ACLs are applied in the following order:

1. PACL for the ingress port
2. VACL for the ingress VLAN
3. Input Cisco IOS ACL
4. Output Cisco IOS ACL
5. VACL for the egress VLAN

In prefer port mode, only the PACL is applied to the ingress packets (the input VACL and Cisco IOS ACL are not applied).

Figure 51-2 Applying ACLs on Routed Packets



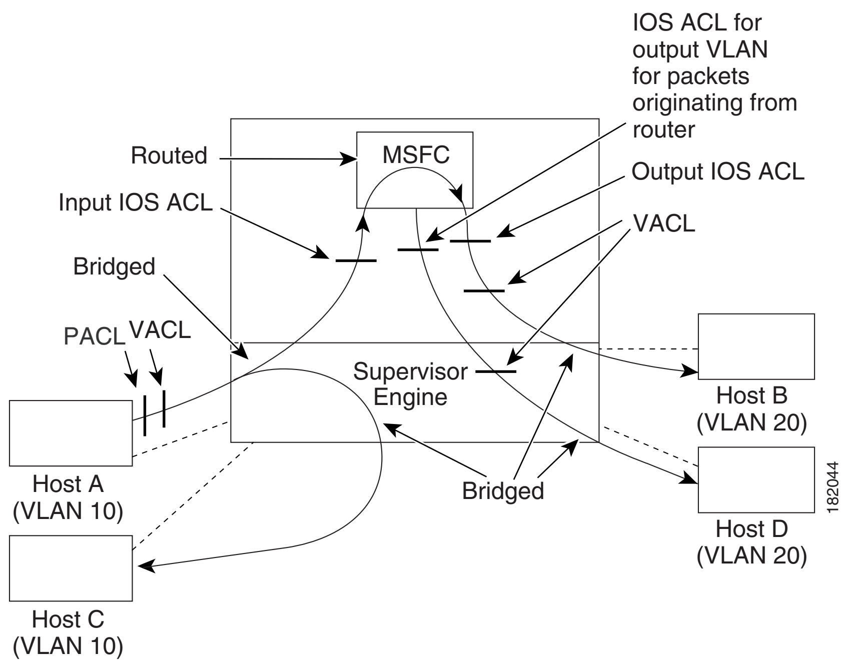
Multicast Packets

Figure 51-3 shows how ACLs are applied on packets that need multicast expansion. For packets that need multicast expansion, the ACLs are applied in the following order:

1. Packets that need multicast expansion:
   a. PACL for the ingress port
   b. VACL for the ingress VLAN
   c. Input Cisco IOS ACL
2. Packets after multicast expansion:
   a. Output Cisco IOS ACL
   b. VACL for the egress VLAN
3. Packets originating from router:
   a. Output Cisco IOS ACL
   b. VACL for the egress VLAN

In prefer port mode, only the PACL is applied to the ingress packets (the input VACL and Cisco IOS ACL are not applied).

Figure 51-3 Applying ACLs on Multicast Packets



Configuring PACLs

Cisco IOS Release 12.2(33)SXH and later releases support PACLs. This section describes how to configure PACLs. PACLs filter incoming traffic on Layer 2 interfaces, using Layer 3 information, Layer 4 header information, or non-IP Layer 2 information.

The PACL feature uses existing Cisco IOS **access-list** commands to create the standard or extended IP ACLs or named MAC-extended ACLs that you want to apply to the port.

Use the **ip access-group** or **mac access-group interface** command to apply an IP ACL or MAC ACL to one or more Layer 2 interfaces.

Note PACLs cannot filter Physical Link Protocols and Logical Link Protocols, such as CDP, VTP, DTP, PAgP, UDLD, and STP, because the protocols are redirected to the switch processor (SP) before the ACL takes effect.

This section contains the following topics:

PACL Configuration Guidelines

Consider the following guidelines when configuring PACLs:

- There can be at most one IP access list and one MAC access list applied to the same Layer 2 interface per direction.
- PACLs are not applied to IPv6, MPLS, or ARP messages.
- An IP access list filters only IPv4 packets, For IP access lists, you can define a standard, extended, or named access-list.
- A MAC access list filters ingress packets that are of an unsupported type (not IP, IPv6, ARP, or MPLS packets) based on the fields of the Ethernet datagram. A MAC access list is not applied to IP, IPv6, MPLS, or ARP messages. You can define only named MAC access lists.
- The number of ACLs and ACEs that can be configured as part of a PACL are bounded by the hardware resources on the switch. Those hardware resources are shared by various ACL features (such as VACLs) that are configured on the system. If there are insufficient hardware resources to program a PACL in hardware, the PACL is not applied.
- PACL does not support the access-list **log** and **reflect/evaluate** keywords. These keywords are ignored if you add them to the access list for a PACL.
- OAL does not support PACLs.
- The access group mode can change the way PACLs interact with other ACLs. To maintain consistent behavior across Cisco platforms, use the default access group mode (merge mode).

Configuring IP and MAC ACLs on a Layer 2 Interface

IP and MAC ACLs can be applied to Layer 2 physical interfaces. Standard (numbered, named) and Extended (numbered, named) IP ACLs, and Extended Named MAC ACLs are supported.

To apply IP or MAC ACLs on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface`	Enters interface configuration mode for a Layer 2 port.
Step 3	`Switch(config-if)# {ip | mac} access-group {name | number | in | out}`	Applies a numbered or named ACL to the Layer 2 interface.
Step 4	`Switch(config)# show running-config`	Displays the access list configuration.

This example shows how to configure the Extended Named IP ACL simple-ip-acl to permit all TCP traffic and implicitly deny all other IP traffic:

```
Switch(config)# ip access-list extended simple-ip-acl
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# end
```

This example shows how to configure the Extended Named MAC ACL simple-mac-acl to permit source host 000.000.011 to any destination host:

```
Switch(config)# mac access-list extended simple-mac-acl
Switch(config-ext-macl)# permit host 000.000.011 any
Switch(config-ext-macl)# end
```

Configuring Access-group Mode on Layer 2 Interface

To configure the access mode on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface`	Enters interface configuration mode for a Layer 2 port.
Step 3	`Switch(config-if)# [no] access-group mode {prefer port | merge}`	Sets the mode for this Layer 2 interface. The **no** prefix sets the mode to the default value (which is merge).
Step 4	`Switch(config)# show running-config`	Displays the access list configuration.

This example shows how to configure an interface to use prefer port mode:

```
Switch# configure terminal
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# access-group mode prefer port
```

This example shows how to configure an interface to use merge mode:

```
Switch# configure terminal
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# access-group mode merge
```

Applying ACLs to a Layer 2 Interface

To apply IP and MAC ACLs to a Layer 2 interface, perform one of these tasks:

Command	Purpose
Switch(config-if)# **ip access-group** *ip-acl* **in**	Applies an IP ACL to the Layer 2 interface.
Switch(config-if)# **mac access-group** *mac-acl* **in**	Applies a MAC ACL to the Layer 2 interface.

This example applies the extended named IP ACL simple-ip-acl to interface GigabitEthernet 6/1 ingress traffic:

```
Switch# configure t
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# ip access-group simple-ip-acl in
```

This example applies the extended named MAC ACL simple-mac-acl to interface GigabitEthernet 6/1 ingress traffic:

```
Switch# configure t
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# mac access-group simple-mac-acl in
```

Applying ACLs to a Port Channel

To apply IP and MAC ACLs to a port channel logical interface, perform this task:

Command	Purpose
Switch(config-if)# **interface port-channel** *number*	Enters configuration mode for the port channel.
Switch(config-if)# **ip access-group** *ip-acl* {**in** | **out**}	Applies an IP ACL to the port channel interface.
Switch(config-if)# **mac access-group** *mac-acl* {**in** | **out**}	Applies a MAC ACL to the port channel interface.

This example applies the extended named IP ACL simple-ip-acl to port channel 3 ingress traffic:

```
Switch# configure t
Switch(config)# interface port-channel 3
Switch(config-if)# ip access-group simple-ip-acl in
```

Displaying an ACL Configuration on a Layer 2 Interface

To display information about an ACL configuration on Layer 2 interfaces, perform one of these tasks:

Command	Purpose
Switch# **show ip access-lists** [**interface** *interface-name*]	Shows the IP access group configuration on the interface.
Switch# **show mac access-group** [**interface** *interface-name*]	Shows the MAC access group configuration on the interface.
Switch# **show access-group mode** [**interface** *interface-name*]	Shows the access group mode configuration on the interface.

This example shows that the IP access group simple-ip-acl is configured on the inbound direction of interface fa6/1:

```
Switch# show ip interface fast 6/1
FastEthernet6/1 is up, line protocol is up
  Inbound  access list is simple-ip-acl
  Outgoing access list is not set
```

This example shows that MAC access group simple-mac-acl is configured on the inbound direction of interface fa6/1:

```
Switch# show mac access-group interface fast 6/1
Interface FastEthernet6/1:
   Inbound access-list is simple-mac-acl
   Outbound access-list is not set
```

This example shows that access group merge is configured on interface fa6/1:

```
Switch# show access-group mode interface fast 6/1
Interface FastEthernet6/1:
   Access group mode is: merge
```

Configuring VACLs

These sections describe how to configure VACLs:

VACL Configuration Guidelines

Consider the following guidelines when configuring VACLs:

- VACLs use standard and extended Cisco IOS IP and MAC layer-named ACLs (see the "Configuring MAC ACLs" section on page 43-68) and VLAN access maps.
- VLAN access maps can be applied to VLANs or to WAN interfaces for VACL capture. VACLs attached to WAN interfaces support only standard and extended Cisco IOS IP ACLs.
- Each VLAN access map can consist of one or more map sequences; each sequence has a match clause and an action clause. The match clause specifies IP or MAC ACLs for traffic filtering and the action clause specifies the action to be taken when a match occurs. When a flow matches a permit ACL entry, the associated action is taken and the flow is not checked against the remaining

sequences. When a flow matches a deny ACL entry, it will be checked against the next ACL in the same sequence or the next sequence. If a flow does not match any ACL entry and at least one ACL is configured for that packet type, the packet is denied.

- To apply access control to both bridged and routed traffic, you can use VACLs alone or a combination of VACLs and ACLs. You can define ACLs on the VLAN interfaces to apply access control to both the ingress and egress routed traffic. You can define a VACL to apply access control to the bridged traffic.
- The following caveats apply to ACLs when used with VACLs:
  - Packets that require logging on the outbound ACLs are not logged if they are denied by a VACL.
  - VACLs are applied on packets before NAT translation. If the translated flow is not subject to access control, the flow might be subject to access control after the translation because of the VACL configuration.
- When VACL capture is configured with Policy Based Routing (PBR) on the same interface, do not select BDD as the ACL merge algorithm. We recommend using ODM, the default ACL merge algorithm for the Supervisor Engine 720.
- When VACL capture is configured on an egress interface together with another egress feature that requires software processing of the traffic, packets of the overlapping traffic may be captured twice.
- By default, software-switched WAN packets are not subjected to ACL lookup in the ACL TCAM and are therefore not affected by hardware-only features. As a result, VACL capture will fail for software-switched WAN packets. In Cisco IOS Release 12.2(33)SXI2 and later releases, you can allow ACLs to be applied to egress or ingress software-switched WAN packets by entering the **platform cwan acl software-switched** {**egress** | **ingress**} command in global configuration mode. To verify whether ACLs will be applied to software-switched WAN packets, enter the **show platform acl software-switched** command as shown in this example:

```
Router (config)# platform cwan acl software-switched ingress
Router (config)# exit
Router# show platform acl software-switched
 CWAN: ACL treatment for software switched in INGRESS is enabled
 CWAN: ACL treatment for software switched in EGRESS is disabled
```

- The action clause in a VACL can be forward, drop, capture, or redirect. Traffic can also be logged. VACLs applied to WAN interfaces do not support the redirect or log actions.
- VACLs cannot be applied to IGMP, MLD, or PIM traffic.
- When the WAN logical interface (Multilink or Multilink Frame Relay) is removed, the corresponding VACL filter applied to the WAN logical interface is also removed and the error message VACL-4-VLANFILTER_CWAN_DELETE appears. The following example displays an illustration of this behavior:

```
Router (config)# do show vlan filter
VLAN Map capture_all:
        Configured on VLANs:  100
            Active on VLANs:  100

        Configured on Interfaces:  Multilink100
            Active on Interfaces:

Router (config)# no interface multilink 100
% Please 'shutdown' this interface before trying to delete it
Router (config)# interface multilink 100
Router (config-if)# show
Router (config-if)# exit
Router (config)# no interface multilink 100
Router (config)#
```

```
%VACL-4-VLANFILTER_CWAN_DELETE: VLAN ACCESS-MAP capture_all applied on Multilink100
will be removed.
Router (config)# do show vlan filter
VLAN Map capture_all:
        Configured on VLANs:  100
            Active on VLANs:  100

Router (config)#
```

Note

- VACLs have an implicit deny at the end of the map; a packet is denied if it does not match any ACL entry, and at least one ACL is configured for the packet type.
- If an empty or undefined ACL is specified in a VACL, any packets will match the ACL, and the associated action is taken.

Defining a VLAN Access Map

To define a VLAN access map, perform this task:

Command	Purpose
`Router(config)# vlan access-map map_name [0-65535]`	Defines the VLAN access map. Optionally, you can specify the VLAN access map sequence number.
`Router(config)# no vlan access-map map_name 0-65535`	Deletes a map sequence from the VLAN access map.
`Router(config)# no vlan access-map map_name`	Deletes the VLAN access map.

When defining a VLAN access map, note the following information:

- To insert or modify an entry, specify the map sequence number.
- If you do not specify the map sequence number, a number is automatically assigned.
- You can specify only one match clause and one action clause per map sequence.
- Use the **no** keyword with a sequence number to remove a map sequence.
- Use the **no** keyword without a sequence number to remove the map.

See the "VLAN Access Map Configuration and Verification Examples" section on page 51-16.

Configuring a Match Clause in a VLAN Access Map Sequence

To configure a match clause in a VLAN access map sequence, perform this task:

Command	Purpose
`Router(config-access-map)# match {ip address {1-199 | 1300-2699 | acl_name} | {mac address acl_name}}`	Configures the match clause in a VLAN access map sequence.
`Router(config-access-map)# no match {ip address {1-199 | 1300-2699 | acl_name} | {mac address acl_name}}`	Deletes the match clause in a VLAN access map sequence.

When configuring a match clause in a VLAN access map sequence, note the following information:

- You can select one or more ACLs.
- VACLs attached to WAN interfaces support only standard and extended Cisco IOS IP ACLs.
- Use the **no** keyword to remove a match clause or specified ACLs in the clause.
- For information about named MAC-Layer ACLs, see the "Configuring MAC ACLs" section on page 43-68.
- For information about Cisco IOS ACLs, see the "Traffic Filtering and Firewalls" section of the *Cisco IOS Security Configuration Guide, Release 12.2*, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

See the "VLAN Access Map Configuration and Verification Examples" section on page 51-16.

Configuring an Action Clause in a VLAN Access Map Sequence

To configure an action clause in a VLAN access map sequence, perform this task:

Command	Purpose
Router(config-access-map)# **action** {**drop** [**log**]} | {**forward** [**capture** | **vlan** *vlan_ID*]} | {**redirect** {{**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot*/*port*} | {**port-channel** *channel_id*}}	Configures the action clause in a VLAN access map sequence.
Router(config-access-map)# **no action** {**drop** [**log**]} | {**forward** [**capture** | **vlan** *vlan_ID*]} | {**redirect** {{**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot*/*port*} | {**port-channel** *channel_id*}}	Deletes the action clause in from the VLAN access map sequence.

When configuring an action clause in a VLAN access map sequence, note the following information:

- You can set the action to drop, forward, forward capture, or redirect packets.
- VACLs applied to WAN interfaces support only the forward capture action. VACLs applied to WAN interfaces do not support the drop, forward, or redirect actions.
- Forwarded packets are still subject to any configured Cisco IOS security ACLs.
- The **capture** action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets. Only forwarded packets can be captured. For more information about the **capture** action, see the "Configuring a Capture Port" section on page 51-16.
- The **forward vlan** action implements Policy-Based Forwarding (PBF), bridging between VLANs.
- VACLs applied to WAN interfaces do not support the **log** action.
- When the **log** action is specified, dropped packets are logged in software. Only dropped IP packets can be logged.
- The **redirect** action allows you to specify up to five interfaces, which can be physical interfaces or EtherChannels. You cannot specify packets to be redirected to an EtherChannel member or a VLAN interface.
- The redirect interface must be in the VLAN for which the VACL access map is configured.

- If a VACL is redirecting traffic to an egress SPAN source port, SPAN does not copy the VACL-redirected traffic.
- SPAN and RSPAN destination ports transmit VACL-redirected traffic.
- Use the **no** keyword to remove an action clause or specified redirect interfaces.

See the "VLAN Access Map Configuration and Verification Examples" section on page 51-16.

Applying a VLAN Access Map

To apply a VLAN access map, perform this task:

Command	Purpose
`Router(config)# **vlan filter** *map_name* {**vlan-list** *vlan_list* | **interface** *type*[1] *number*[2]}`	Applies the VLAN access map to the specified VLANs or WAN interfaces.

1. *type* = **pos**, **atm**, or **serial**
2. number = *slot/port* or *slot/port_adapter/port*; can include a subinterface or channel group descriptor

When applying a VLAN access map, note the following information:

- You can apply the VLAN access map to one or more VLANs or WAN interfaces.
- The *vlan_list* parameter can be a single VLAN ID or a comma-separated list of VLAN IDs or VLAN ID ranges (*vlan_ID–vlan_ID*).
- If you delete a WAN interface that has a VACL applied, the VACL configuration on the interface is also removed.
- You can apply only one VLAN access map to each VLAN or WAN interface.
- VACLs applied to VLANs are active only for VLANs with a Layer 3 VLAN interface configured. Applying a VLAN access map to a VLAN without a Layer 3 VLAN interface creates an administratively down Layer 3 VLAN interface to support the VLAN access map.
- VACLs applied to VLANs are inactive if the Layer 2 VLAN does not exist or is not operational.
- You cannot apply a VACL to a secondary private VLAN. VACLs applied to primary private VLANs also apply to secondary private VLANs.
- Use the **no** keyword to clear VLAN access maps from VLANs or WAN interfaces.

See the "VLAN Access Map Configuration and Verification Examples" section on page 51-16.

Verifying VLAN Access Map Configuration

To verify VLAN access map configuration, perform this task:

Command	Purpose
`Router# **show vlan access-map** [*map_name*]`	Verifies VLAN access map configuration by displaying the content of a VLAN access map.
`Router# **show vlan filter** [**access-map** *map_name* | **vlan** *vlan_id* | **interface** *type*[1] *number*[2]]`	Verifies VLAN access map configuration by displaying the mappings between VACLs and VLANs.

1. *type* = **pos**, **atm**, or **serial**

2. number = *slot/port* or *slot/port_adapter/port*; can include a subinterface or channel group descriptor

VLAN Access Map Configuration and Verification Examples

Assume IP-named ACL **net_10** and **any_host** are defined as follows:

```
Router# show ip access-lists net_10
Extended IP access list net_10
    permit ip 10.0.0.0 0.255.255.255 any

Router# show ip access-lists any_host
Standard IP access list any_host
    permit any
```

This example shows how to define and apply a VLAN access map to forward IP packets. In this example, IP traffic matching net_10 is forwarded and all other IP packets are dropped due to the default drop action. The map is applied to VLAN 12 to 16.

```
Router(config)# vlan access-map thor 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action forward
Router(config-access-map)# exit
Router(config)# vlan filter thor vlan-list 12-16
```

This example shows how to define and apply a VLAN access map to drop and log IP packets. In this example, IP traffic matching net_10 is dropped and logged and all other IP packets are forwarded:

```
Router(config)# vlan access-map ganymede 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action drop log
Router(config-access-map)# exit
Router(config)# vlan access-map ganymede 20
Router(config-access-map)# match ip address any_host
Router(config-access-map)# action forward
Router(config-access-map)# exit
Router(config)# vlan filter ganymede vlan-list 7-9
```

This example shows how to define and apply a VLAN access map to forward and capture IP packets. In this example, IP traffic matching net_10 is forwarded and captured and all other IP packets are dropped:

```
Router(config)# vlan access-map mordred 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action forward capture
Router(config-access-map)# exit
Router(config)# vlan filter mordred vlan-list 2, 4-6
```

Configuring a Capture Port

A port configured to capture VACL-filtered traffic is called a capture port.

Note To apply IEEE 802.1Q or ISL tags to the captured traffic, configure the capture port to trunk unconditionally (see the "Configuring the Layer 2 Switching Port as an ISL or 802.1Q Trunk" section on page 17-10 and the "Configuring the Layer 2 Trunk Not to Use DTP" section on page 17-11).

To configure a capture port, perform this task:

	Command	Purpose
Step 1	`Router(config)# **interface** {{*type*[1] *slot/port*}`	Specifies the interface to configure.
Step 2	`Router(config-if)# **switchport capture allowed vlan** {**add** | **all** | **except** | **remove**} *vlan_list*`	(Optional) Filters the captured traffic on a per-destination-VLAN basis. The default is **all**.
Step 3	`Router(config-if)# **switchport capture**`	Configures the port to capture VACL-filtered traffic.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring a capture port, note the following information:

- You can configure any port as a capture port.
- The *vlan_list* parameter can be a single VLAN ID or a comma-separated list of VLAN IDs or VLAN ID ranges (*vlan_ID*–*vlan_ID*).
- To encapsulate captured traffic, configure the capture port with the **switchport trunk encapsulation** command (see the "Configuring a Layer 2 Switching Port as a Trunk" section on page 17-10) before you enter the **switchport capture** command.
- For unencapsulated captured traffic, configure the capture port with the **switchport mode access** command (see the "Configuring a LAN Interface as a Layer 2 Access Port" section on page 17-16) before you enter the **switchport capture** command.
- The capture port supports only egress traffic. No traffic can enter the switch through a capture port.

This example shows how to configure a Fast Ethernet interface 5/1 as a capture port:

```
Router(config)# interface gigabitEthernet 5/1
Router(config-if)# switchport capture
Router(config-if)# end
```

This example shows how to display VLAN access map information:

```
Router# show vlan access-map mymap
Vlan access-map "mymap"  10
        match: ip address net_10
        action: forward capture
Router#
```

This example shows how to display mappings between VACLs and VLANs. For each VACL map, there is information about the VLANs that the map is configured on and the VLANs that the map is active on. A VACL is not active if the VLAN does not have an interface.

```
Router# show vlan filter
VLAN Map mordred:
        Configured on VLANs:  2,4-6
            Active on VLANs:  2,4-6
Router#
```

Configuring MAC PBF

To configure MAC policy-based forwarding (PBF), perform this task on each source VLAN:

	Command	Purpose
Step 1	`Router(config)#` **`mac host`** *`my_host mac_addr`*	(Optional) Assigns a name to the MAC address of the source host.
Step 2	`Router(config)#` **`mac access-list extended`** *`macl_name`*	Configures a MAC ACL.
Step 3	`Router(config-ext-macl)#` **`permit host`** *`my_host`* **`any`**	Configures an access control entry (ACE) to permit traffic from the named host to any other address. Hosts can be specified by a name or by a MAC address.
	`Router(config-ext-macl)#` **`permit host`** *`my_host`* **`host`** *`other_host`*	Configures an ACE to permit traffic from the named host to one other host.
Step 4	`Router(config-ext-macl)#` **`exit`**	Exits ACL configuration.
Step 5	`Router(config)#` **`vlan access-map`** *`map_name`*	Defines a VLAN access map.
Step 6	`Router(config-access-map)#` **`match mac address`** *`macl_name`*	Applies the MAC ACL to this VLAN access map.
Step 7	`Router(config-access-map)#` **`action forward vlan`** *`other_vlan_ID`* `[`**`local`**`]`	Forwards matching traffic to the other VLAN. **Note** By default, PBF-specified devices on the same VLAN cannot communicate with each other. To allow local communication by the host, use the **local** keyword.
Step 8	`Router(config-access-map)#` **`exit`**	Exits access map configuration.
Step 9	`Router(config)#` **`vlan filter`** *`map_name`* **`vlan-list`** *`my_vlan_ID`*	Applies the VLAN access map to the specified VLAN.
Step 10	`Router(config)#` **`interface vlan`** *`my_vlan_ID`*	Enters interface configuration mode for the VLAN.
Step 11	`Router(config-if)#` **`mac packet-classify`**	Classifies incoming or outgoing Layer 3 packets on this VLAN as Layer 2 packets.
Step 12	`Router(config-if)#` **`exit`**	Exits interface configuration.
Step 13	`Router(config)#` **`mls rate-limit unicast acl mac-pbf`** *`pps`* `[`*`burst_size`*`]`	(Optional) Sets a rate limit on PBF packets. • *pps*—Maximum number of packets per second. The range is 10 to 1000000 packets per second. • *burst_siz*e—Maximum number of packets in a burst. The range is 1 to 255 packets.
Step 14	`Router(config)#` **`exit`**	Exits global configuration mode.
Step 15	`Router#` **`show vlan mac-pbf config`**	Displays MAC PBF configuration and statistics.
Step 16	`Router#` **`clear vlan mac-pbf counters`**	(Optional) Clears MAC PBF packet counters.

When configuring MAC PBF, note the following information:

- To allow traffic in both directions between two VLANs, you must configure MAC PBF in both VLANs.
- You can configure MAC PBF between hosts in different switches.

- By default, MAC PBF hosts in the same VLAN cannot communicate with each other. To allow local communication, use the **local** keyword.
- When configuring the **vlan filter** command, specify only one VLAN after the **vlan-list** keyword. If you specify more than one VLAN, MAC PBF will ignore all but the last VLAN in the list.
- The output of the **show vlan mac-pbf config** command displays the following fields for configured PBF paths:
  - Rcv Vlan — The number of the VLAN to which packets are forwarded by PBF.
  - Snd Vlan — The number of the VLAN which will forward packets by PBF.
  - DMAC — The MAC address of the destination host on the receiving VLAN.
  - SMAC — The MAC address of the source host on the sending VLAN.
  - (Local) — Displays 1 if the **local** keyword is configured in the **action forward vlan** command on the sending VLAN; displays 0 if the **local** keyword is not configured.
  - (Packet counter) — The number of packets that have been forwarded from the sending VLAN to the receiving VLAN. To clear this counter, enter the **clear vlan mac-pbf counters** command.
  - Pkts dropped — The number of packets that have been dropped by the sending VLAN. To clear this counter, enter the **clear vlan mac-pbf counters** command.
- If the sending VLAN is shut down, MAC PBF will still function. Shutting down a VLAN disables Layer 3 functionality, but MAC PBF is a Layer 2 function.

This example shows how to configure and display MAC PBF to allow two hosts in separate VLANs (“red” VLAN 100 and “blue” VLAN 200) on the same switch to exchange packets:

```
Router(config)# mac host host_red3 0001.0002.0003
Router(config)# mac access-list extended macl_red
Router(config-ext-macl)# permit host host_red host host_blue
Router(config-ext-macl)# exit
Router(config)# vlan access-map red_to_blue
Router(config-access-map)# match mac address macl_red
Router(config-access-map)# action forward vlan 200 local
Router(config-access-map)# exit
Router(config)# vlan filter red_to_blue vlan-list 100
Router(config)# interface vlan 100
Router(config-if)# mac packet-classify
Router(config-if)# exit
Router(config)#
Router(config)# mac host host_blue5 0001.0002.0005
Router(config)# mac access-list extended macl_blue
Router(config-ext-macl)# permit host host_blue host host_red
Router(config-ext-macl)# exit
Router(config)# vlan access-map blue_to_red
Router(config-access-map)# match mac address macl_blue
Router(config-access-map)# action forward vlan 100
Router(config-access-map)# exit
Router(config)# vlan filter blue_to_red vlan-list 200
Router(config)# interface vlan 200
Router(config-if)# mac packet-classify
Router(config-if)# exit
Router#
Router# show vlan mac-pbf config
 Rcv Vlan 100, Snd Vlan 200, DMAC 0001.0002.0003, SMAC 0001.0002.0005 1 15
 Rcv Vlan 200, Snd Vlan 100, DMAC 0001.0002.0005, SMAC 0001.0002.0003 0 23
 Pkts Dropped 0
Router#
```

Configuring VACL Logging

When you configure VACL logging, IP packets that are denied generate log messages in these situations:

- When the first matching packet is received
- For any matching packets received during the last 5-minute interval
- If the threshold is reached before the 5-minute interval

Log messages are generated on a per-flow basis. A flow is defined as packets with the same IP addresses and Layer 4 (UDP or TCP) port numbers. When a log message is generated, the timer and packet count is reset.

These restrictions apply to VACL logging:

- Because of the rate-limiting function for redirected packets, VACL logging counters may not be accurate.
- Only denied IP packets are logged.

To configure VACL logging, use the **action drop log** command action in VLAN access map submode (see the "Configuring PACLs" section on page 51-8 for configuration information) and perform this task in global configuration mode to specify the global VACL logging parameters:

	Command	Purpose
Step 1	`Router(config)#` **`vlan access-log maxflow`** *`max_number`*	Sets the log table size. The content of the log table can be deleted by setting the maxflow number to 0. The default is 500 with a valid range of 0 to 2048. When the log table is full, logged packets from new flows are dropped by the software.
Step 2	`Router(config)#` **`vlan access-log ratelimit`** *`pps`*	Sets the maximum redirect VACL logging packet rate. The default packet rate is 2000 packets per second with a valid range of 0 to 5000. Packets exceeding the limit are dropped by the hardware.
Step 3	`Router(config)#` **`vlan access-log threshold`** *`pkt_count`*	Sets the logging threshold. A logging message is generated if the threshold for a flow is reached before the 5-minute interval. By default, no threshold is set.
Step 4	`Router(config)#` **`exit`**	Exits VLAN access map configuration mode.
Step 5	`Router#` **`show vlan access-log config`**	(Optional) Displays the configured VACL logging properties.
Step 6	`Router#` **`show vlan access-log flow`** *`protocol`* `{{`*`src_addr src_mask`*`} |` **`any`** `| {`**`host`** `{`*`hostname`* `|` *`host_ip`*`}}} {{`*`dst_addr dst_mask`*`} |` **`any`** `| {`**`host`** `{`*`hostname`* `|` *`host_ip`*`}}} [`**`vlan`** *`vlan_id`*`]`	(Optional) Displays the content of the VACL log table.
Step 7	`Router#` **`show vlan access-log statistics`**	(Optional) Displays packet and message counts and other statistics.

This example shows how to configure global VACL logging in hardware:

```
Router(config)# vlan access-log maxflow 800
Router(config)# vlan access-log ratelimit 2200
Router(config)# vlan access-log threshold 4000
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Denial of Service Protection

This chapter contains information on how to protect your switch against Denial of Service (DoS) attacks. The information covered in this chapter is unique to Cisco IOS Release 12.2SX, and it supplements the network security information and procedures in the "Configuring Network Security" chapter in this publication as well as the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.2, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

- *Cisco IOS Security Command Reference*, Release 12.2, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The Release 12.2 publications at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding DoS Protection

This section contains information about the available methods to counteract DoS attacks and includes configuration examples. The PFC3 provides a layered defense against DoS attacks using the following methods:

- CPU rate limiters—Controls traffic types.
- Control plane policing (CoPP)—Filters and rate limits control plane traffic. For information about CoPP, see Chapter 53, "Configuring Control Plane Policing."

These sections describe DoS protection:

Security ACLs and VACLs

If the network is under a DoS attack, ACLs can be an efficient method for dropping the DoS packets before they reach the intended target. Use security ACLs if an attack is detected from a particular host.

In this example, the host 10.1.1.10 and all traffic from that host is denied:

```
Router(config)# access-list 101 deny ip host 10.1.1.10 any
Router(config)# access-list 101 permit ip any any
```

Security ACLs also protect against the spoofing of addresses. For example, assume that a source address A is on the inside of a network and a switch interface that is pointing to the Internet. You can apply an inbound ACL on the switch Internet interface that denies all addresses with a source of A (the inside address). This action stops attacks where the attackers spoof inside source addresses. When the packet arrives at the switch interface, it matches on that ACL and drops the packet before it causes damage.

When the switch is used with a Cisco Intrusion Detection Module (CIDM), you can dynamically install the security ACL as a response to the detection of the attack by the sensing engine.

VACLs are a security enforcement tool based on Layer 2, Layer 3, and Layer 4 information. The result of a VACL lookup against a packet can be a permit, a deny, a permit and capture, or a redirect. When you associate a VACL with a particular VLAN, all traffic must be permitted by the VACL before the traffic is allowed into the VLAN. VACLs are enforced in hardware, so there is no performance penalty for applying VACLs to a VLAN.

QoS Rate Limiting

QoS ACLs limit the amount of a particular type of traffic that is processed by the RP. If a DoS attack is initiated against the RP, QoS ACLs can prevent the DoS traffic from reaching the RP data path and congesting it. The PFC3 performs QoS in hardware, which offers an efficient means of limiting DoS traffic (once that traffic has been identified) to protect the switch from impacting the RP.

For example, if the network is experiencing ping-of-death or smurf attacks, the administrator should rate limit the ICMP traffic to counteract the DoS attack and still allow legitimate traffic through the processor, or allow it to be forwarded to the RP or host. This rate limiting configuration must be done for each flow that should be rate limited and the rate-limiting policy action should be applied to the interface.

In the following example, the access-list 101 permits and identifies ping (echo) ICMP messages from any source to any destination as traffic. Within the policy map, a policing rule defines a specified committed information rate (CIR) and burst value (96000 bps and 16000 bps) to rate limit the ping (ICMP) traffic through the chassis. The policy map then is applied to an interface or VLAN. If the ping traffic exceeds the specified rate on the VLAN or interface where the policy map is applied, it is dropped as specified in the markdown map (the markdown map for the normal burst configurations is not shown in the example).

```
Router(config)# access-list 101 permit icmp any any echo
Router(config)# class-map match-any icmp_class
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
Router(config)# policy-map icmp_policer
Router(config-pmap)# class icmp_class
Router(config-pmap-c)# police 96000 16000 conform-action transmit exceed-action
policed-dscp-transmit drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

uRPF Check

When you enable the unicast reverse path forwarding (uRPF) check, packets that lack a verifiable source IP address, such as spoofed IP source addresses, are discarded. Cisco Express Forwarding (CEF) tables are used to verify that the source addresses and the interfaces on which they were received are consistent with the switch processor (SP) FIB tables.

After you enable uRPF check on an interface (per-VLAN basis), the incoming packet is compared to the CEF tables through a reverse lookup. If the packet is received from one of the reverse path routes, the packet is forwarded. If there is no reverse path route on the interface on which the packet was received, the packet fails the uRPF check and is either dropped or forwarded, depending on whether an ACL is applied to the uRPF check fail traffic. If no ACL is specified in the CEF tables, then the forged packets are immediately dropped.

You can only specify an ACL for the uRPF check for packets that fail the uRPF check. The ACL checks whether the packet should immediately be dropped or forwarded. The uRPF check with ACL is not supported in any PFC3 in hardware. Packets that are denied in the uRPF ACL are forwarded in hardware. Packets that are permitted are sent to the CPU.

The uRPF check is supported in hardware. However, all packets that fail the uRPF check, and are forwarded because of an applied ACL, can be sent and rate limited to the RP to generate ICMP unreachable messages; these actions are all software driven. The uRPF check in hardware is supported for routes with up to two return paths (interfaces) and up to six return paths with interface groups configured (two from the FIB table and four from the interface groups).

Traffic Storm Control

A traffic storm occurs when packets flood the LAN, which creates excessive traffic and degrades network performance. The traffic storm control feature prevents LAN ports from being disrupted by a broadcast, multicast, or unicast traffic storm on physical interfaces from either mistakes in network configurations or from users issuing a DoS attack. Traffic storm control (also called traffic suppression) monitors incoming traffic levels over a 1-second traffic storm control interval. During the interval, traffic storm control compares the traffic level with the configured traffic storm control level. The traffic storm control level is a percentage of the total available bandwidth of the port. Each port has a single traffic storm control level that is used for all types of traffic (broadcast, multicast, and unicast).

Traffic storm control is configured on an interface and is disabled by default. The configuration example here enables broadcast address storm control on interface FastEthernet 2/3 to a level of 20 percent. When the broadcast traffic exceeds the configured level of 20 percent of the total available bandwidth of the port within a 1-second traffic-storm-control interval, traffic storm control will drop all broadcast traffic until the end of the traffic-storm-control interval.

```
Router(config-if)# storm-control broadcast level 20
```

The switch supports broadcast storm control on all LAN ports and multicast and unicast storm control on Gigabit Ethernet ports.

When two or three suppression modes are configured simultaneously, they share the same level settings. If broadcast suppression is enabled, and if multicast suppression is also enabled and configured at a 70-percent threshold, the broadcast suppression will also have a setting for 70 percent.

Network Under SYN Attack

A network under a SYN attack is easily recognized. The target host becomes unusually slow, crashes, or suspends operation. Traffic returned from the target host can also cause trouble on the RP because return traffic goes to randomized source addresses of the original packets, lacks the locality of “real” IP traffic, and may overflow route caches, or CEF tables.

When the network is under a SYN attack, the TCP intercept feature becomes aggressively defensive. Two factors determine when aggressive behavior on the switch begins and ends:

- The total incomplete connections
- Connection requests during the last one-minute sample period

Both factors are configured with low and high values.

If the number of incomplete connections exceed 1,100, or the number of connections arriving in the last one-minute period exceed 1,100, each new arriving connection causes the oldest partial connection (or a random connection) to be deleted. These are the default values, which can be altered. When either of the thresholds is exceeded, the TCP intercept assumes the server is under attack and goes into aggressive mode with the following reactions:

- Each new arriving connection causes the oldest partial (or random partial) to be deleted.
- The initial retransmission timeout is reduced by half to 0.5 seconds, and so the total time trying to establish the connection is cut in half.
- In watch mode, the watch timeout is reduced by half.

Note When both thresholds fall below the configured low value, the aggressive behavior ceases (default value is 900 in both factors).

TCP flows are hardware assisted on the PFC3.

Protocol Packet Policing

Attackers may try to overwhelm the RP CPU with routing protocol or ARP control packets. The **mls qos protocol** command rate limits this traffic in hardware. Enter the **`mls qos protocol ?`** to display the supported routing protocols.

For example, the **mls qos protocol arp police** command rate limits ARP packets.

ARP packets are approximately 40 bytes long and ARP reply packets are approximately 60 bytes long. The policer rate value is in bits per second. The burst value is in bytes per second. Together, an ARP request and reply are approximately 800 bits.

This example shows how to allow 200 ARP requests and replies per second:

```
Router(config)# mls qos protocol arp police 200000 6000
```

Note

- The minimum values supported by the **mls qos protocol arp police** command are too small for use in production networks.
- The configured rate limits are applied separately to the PFC and each DFC. The RP CPU will receive the configured value times the number of forwarding engines.
- The protocol packet policing mechanism effectively protects the RP CPU against attacks such as line-rate ARP attacks, but it polices both routing protocols and ARP packets to the switch and also polices traffic through the switch with less granularity than CoPP.
- The policing mechanism shares the root configuration with a policing-avoidance mechanism. The policing-avoidance mechanism lets the routing protocol and ARP packets flow through the network when they reach a QoS policer. This mechanism can be configured using the **mls qos protocol** *protocol_name* **pass-through** command.

This example shows how to display the available protocols to use with protocol packet policing:

```
Router(config)# mls qos protocol ?
  isis
  eigrp
  ldp
  ospf
  rip
  bgp
  ospfv3
  bgpv2
  ripng
  neigh-discover
  wlccp
  arp
```

This example shows how to display the available keywords to use with the **mls qos protocol** command:

```
Router(config)# mls qos protocol protocol_name ?
  pass-through  pass-through keyword
  police        police keyword
  precedence    change ip-precedence(used to map the dscp to cos value)
```

Recommended Rate-Limiter Configuration

The recommended rate-limiter configuration is as follows:

- Enable the rate limiters for the traffic types most likely to be used in a DoS attack.
- Do not use a rate limiter on VACL logging unless you configure VACL logging.
- Disable redirects.
- Disable unreachables.
- Do not enable the MTU rate limiter if all interfaces have the same MTU.
- When configuring the Layer 2 PDU rate limiter, note the following information:
  - Calculate the expected or possible number of valid PDUs and double or triple the number.
  - PDUs include BPDUs, DTP, VTP, PAgP, LACP, UDLD, etc.
  - Rate limiters do not discriminate between good frames or bad frames.

Hardware-Based Rate Limiters on the PFC3

The PFC3 supports additional hardware-based rate limiters. The PFC3 provides eight rate-limiter registers for the new rate limiters, which are configured globally on the switch. These rate-limiter registers are present in the Layer 3 forwarding engine (PFC) and are responsible for containing rate-limiting information for result packets that match the various available configured rate limiters.

Because eight rate-limiter registers are present on the PFC3, these registers can force different rate-limiting scenarios to share the same register. The registers are assigned on a first-come, first-serve basis. If all registers are being utilized, the only way to configure another rate limiter is to free one register.

These are the hardware-based rate limiters available on the PFC3:

- Ingress-Egress ACL Bridged Packets (Unicast Only), page 52-7
- uRPF Check Failure, page 52-7
- TTL Failure, page 52-8
- ICMP Unreachable (Unicast Only), page 52-8
- FIB (CEF) Receive Cases (Unicast Only), page 52-8
- FIB Glean (Unicast Only), page 52-9
- Layer 3 Security Features (Unicast Only), page 52-9
- ICMP Redirect (Unicast Only), page 52-9
- VACL Log (Unicast Only), page 52-9
- MTU Failure, page 52-10
- Layer 2 PDU, page 52-10
- Layer 2 Protocol Tunneling, page 52-10
- IP Errors, page 52-11
- Layer 2 Multicast IGMP Snooping, page 52-10
- IPv4 Multicast, page 52-11
- IPv6 Multicast, page 52-12

Ingress-Egress ACL Bridged Packets (Unicast Only)

This rate limiter rate limits packets sent to the RP because of an ingress/egress ACL bridge result. The switch accomplishes this by altering existing and new ACL TCAM entries with a TCAM bridge result to a Layer 3 redirect result pointing to the RP. Packets hitting the TCAM entries with the altered Layer 3 redirect rate limit result will be rate limited according to the instructions set in CLI by the network administrator. Both the ingress and egress values will be the same, as they both share the same rate-limiter register. If the ACL bridge ingress/egress rate limiting is disabled, the Layer 3 redirect rate limit results are converted to the bridge result.

Ingress or egress ACL-bridged packet cases share a single rate-limiter register. If the feature is turned on, ingress and egress ACLs use the same rate-limiter value.

Burst values regulate how many packets can be allowed in a burst. Each allowed packet consumes a token and a token must be available for a packet to be allowed. One token is generated per millisecond. When packets are not coming in, tokens can be accumulated up to the burst value. For example, if the burst value is set to 50, the switch can accumulate up to 50 tokens and absorb a burst of 50 packets.

This example shows how to rate limit the unicast packets from an ingress ACL bridge result to 50000 packets per second, and 50 packets in burst:

```
Router(config)# mls rate-limit unicast acl input 50000 50
```

This example shows how to rate limit the unicast packets from an ingress ACL bridge result to the same rate (50000 pps and 50 packets in burst) for egress ACL bridge results:

```
Router(config)# mls rate-limit unicast acl output 50000 50
```

If the values of the rate limiter are altered on either the ingress or the egress when both are enabled, both values are changed to that new value. In the following example, the output rate is changed to 40000 pps:

```
Router(config)# mls rate-limit unicast acl output 40000 50
```

When you enter the **show mls rate-limit** command, both the ACL bridged in and the ACL bridged out display the new value of 40000 pps:

```
Router# show mls rate-limit
  Rate Limiter Type       Status      Packets/s   Burst
---------------------   ----------    ---------   -----
MCAST NON RPF           Off           -           -
MCAST DFLT ADJ          On            100000      100
MCAST DIRECT CON        Off           -           -
ACL BRIDGED IN          On            40000       50
ACL BRIDGED OUT         On            40000       50
IP FEATURES             Off
…
```

uRPF Check Failure

The uRPF check failure rate limiter allows you to configure a rate for the packets that need to be sent to the RP because they failed the uRPF check. The uRPF checks validate that incoming packets on an interface are from a valid source, which minimizes the potential threat of DoS attacks from users using spoofed addresses. When spoofed packets fail the uRPF check, those failures can be sent to the RP. The uRPF check rate limiters allow you to rate limit the packets per second that are bridged to the RP CPU when a uRPF check failure occurs.

This example shows how to rate limit the uRPF check failure packets sent to the RP to 100000 pps with a burst of 100 packets:

```
Router(config)# mls rate-limit unicast ip rpf-failure 100000 100
```

TTL Failure

This rate limiter rate limits packets sent to the RP because of a time-to-live (TTL) check failure. As indicated by the **all** keyword in the following example, this rate limiter applies to both multicast and unicast traffic.

Note The TTL failure rate limiter is not supported for IPv6 multicast.

This example shows how to rate limit the TTL failures to 70000 pps with a burst of 150:

```
Router(config)# mls rate-limit all ttl-failure 70000 150
```

For packets having TTL value equal to 1, configure MLS ttl-failure rate-limiter for CPU protection. The following are a few exceptions:

- When the incoming traffic also requires ARP resolution (known network, unknown host), use glean rate-limiter.
- When the incoming traffic is also destined to the IP of the router, use receive-rate limiter.
- When the incoming traffic is also receiving input ACL bridged (for example, ACL log), use acl bridge input rate-limiter.

ICMP Unreachable (Unicast Only)

In an ICMP unreachable attack, a device is flooded with a large number of packets that contain a destination address that is unreachable from the flooded device (in this case, the RP). The ICMP unreachable rate limiter allows you to rate limit the packets that are sent to the RP containing unreachable addresses.

This example shows how to rate limit the packets that are sent to the RP because of an ACL drop to 10000 pps and a burst of 100:

```
Router(config)# mls rate-limit unicast ip icmp unreachable acl-drop 10000 100
```

This example shows how to rate limit the packets that require generation of ICMP-unreachable messages because of a FIB miss to 80000 pps and burst to 70:

```
Router(config)# mls rate-limit unicast ip icmp unreachable no-route 80000 70
```

The four rate limiters, ICMP unreachable no route, ICMP unreachable ACL drop, IP errors, and IP RPF failure, share a single rate-limiter register. If any of these limiters are enabled, all of the limiters in this group will share the same value and sometimes the same state (for example, ON/ON/ON). When verifying the rate limiters, if the members of this register are enabled through another feature, an ON-Sharing status (instead of an ON status) is displayed. The exception is the TTL failure rate limiter: its value shares the same value as the other members in the register if you have manually enabled the feature.

FIB (CEF) Receive Cases (Unicast Only)

The FIB receive rate limiter provides the capability to rate limit all packets that contain the RP IP address as the destination address. The rate limiters do not discriminate between good frames and bad frames.

Note Do not enable the FIB receive rate limiter if you are using CoPP. The FIB receive rate limiter overrides the CoPP policies.

This example shows how to rate limit the traffic to 25000 pps with a burst of 60:

```
Router(config)# mls rate-limit unicast cef receive 25000 60
```

FIB Glean (Unicast Only)

The FIB glean rate limiter does not limit ARP traffic, but provides the capability to rate limit traffic that requires address resolution (ARP) and requires that it be sent to the RP. This situation occurs when traffic enters a port and contains the destination of a host on a subnet that is locally connected to the RP, but no ARP entry exists for that destination host. In this case, because the MAC address of the destination host will not be answered by any host on the directly connected subnet that is unknown, the “glean” adjacency is hit and the traffic is sent directly to the RP for ARP resolution. This rate limiter limits the possibility of an attacker overloading the CPU with such ARP requests.

This example shows how to rate limit the rate at which this traffic is sent to the RP to 20000 pps and a burst of 60:

```
Router(config)# mls rate-limit unicast cef glean 20000 60
```

Layer 3 Security Features (Unicast Only)

Some security features are processed by first being sent to the RP. For these security features, you need to rate limit the number of these packets being sent to the RP to reduce any potential overloading. The security features include authentication proxy (auth-proxy), IPSEC, and inspection.

Authentication proxy is used to authenticate inbound or outbound users or both. These users are normally blocked by an access list, but with auth-proxy, the users can bring up a browser to go through the firewall and authenticate on a terminal access controller access control system plus (TACACS+) or RADIUS server (based on the IP address). The server passes additional access list entries down to the switch to allow the users through after authentication. These ACLs are stored and processed in software, and if there are many users utilizing auth-proxy, the RP may be overwhelmed. Rate limiting would be advantageous in this situation.

IPsec and inspection are also done by the RP and may require rate limiting. When the Layer 3 security feature rate limiter is enabled, all Layer 3 rate limiters for auth-proxy, IPsec and inspection are enabled at the same rate.

This example shows how to rate limit the security features to the RP to 100000 pps with a burst of 10 packets:

```
Router(config)# mls rate-limit unicast ip features 100000 10
```

ICMP Redirect (Unicast Only)

The ICMP-redirect rate limiter allows you to rate limit ICMP traffic. For example, when a host sends packets through a nonoptimal switch, the RP sends ICMP-redirect messages to the host to correct its sending path. If this traffic occurs continuously, and is not rate limited, the RP will continuously generate ICMP-redirect messages.

This example shows how to rate limit the ICMP redirects to 20000 pps, with a burst of 20 packets:

```
Router(config)# mls rate-limit unicast ip icmp redirect 20000 20
```

VACL Log (Unicast Only)

Packets that are sent to the RP because of VLAN-ACL logging can be rate limited to ensure that the CPU is not overwhelmed with logging tasks. VACLs are processed in hardware, but the RP does the logging. When VACL logging is configured on the switch, IP packets that are denied in the VACL generate log messages.

This example shows how to rate limit logging requests to 5000 pps (the range for this rate limiter is from 10 to 5000 pps):

```
Router(config)# mls rate-limit unicast acl vacl-log 5000
```

MTU Failure

Similar to the TTL failure rate limiter, the rate limiter for MTU failures is supported for both unicast and multicast traffic. Packets that fail an MTU check are sent to the RP CPU. This might cause the RP to be overwhelmed.

This example shows how to rate limit packets failing the MTU failures from being sent to the RP to 10000 pps with a burst of 10:

```
Router(config)# mls rate-limit all mtu 10000 10
```

Layer 2 Multicast IGMP Snooping

The IGMP snooping rate limiter limits the number of Layer 2 IGMP packets destined for the SP. IGMP snooping listens to IGMP messages between the hosts and the switch. You cannot enable the Layer 2 PDU rate limiter if the switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel. In releases earlier than Cisco IOS Release 12.2(33)SXH, the IGMP snooping rate limiter also rate limits PIM messages. In Cisco IOS Release 12.2(33)SXH and later releases, the IGMP snooping rate limiter does not rate limit PIM messages.

This example shows how to rate limit IGMP-snooping traffic:

```
Router(config)# mls rate-limit multicast ipv4 igmp 20000 40
```

Layer 2 PDU

The Layer 2 protocol data unit (PDU) rate limiter allows you to limit the number of Layer 2 PDU protocol packets (including BPDUs, DTP, PAgP, CDP, STP, and VTP packets) destined for the SP and not the RP CPU. You cannot enable the Layer 2 PDU rate limiter if the switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.

This example shows how to rate limit Layer 2 PDUs to 20000 pps with a burst of 20 packets:

```
Router(config)# mls rate-limit layer2 pdu 20000 20
```

Layer 2 Protocol Tunneling

This rate limiter limits the Layer 2 protocol tunneling packets, which include control PDUs, CDP, STP, and VTP packets destined for the SP. These packets are encapsulated in software (rewriting the destination MAC address in the PDU), and then forwarded to a proprietary multicast address (01-00-0c-cd-cd-d0). You cannot enable the Layer 2 PDU rate limiter if the switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.

This example shows how to rate limit Layer 2 protocol tunneling packets to 10000 pps with a burst of 10 packets:

```
Router(config)# mls rate-limit layer2 l2pt 10000 10
```

IP Errors

This rate limiter limits the packets with IP checksum and length errors. When a packet reaches the PFC3 with an IP checksum error or a length inconsistency error, it must be sent to the RP for further processing. An attacker might use the malformed packets to carry out a DoS attack, but the network administrator can configure a rate for these types of packets to protect the control path.

This example shows how to rate limit IP errors sent to the RP to 1000 pps with a burst of 20 packets:

```
Router(config)# mls rate-limit unicast ip errors 1000 20
```

IPv4 Multicast

This rate limiter limits the IPv4 multicast packets. The rate limiters can rate limit the packets that are sent from the data path in the hardware up to the data path in the software. The rate limiters protect the control path in the software from congestion and drop the traffic that exceeds the configured rate. Within the IPv4 multicast rate limiter, there are three rate limiters that you can also configure: the FIB-miss rate limiter, the multicast partially switched flows rate limiter, and the multicast directly connected rate limiter.

The FIB-miss rate limiter allows you to rate limit the multicast traffic that does not match an entry in the mroute table.

The partially switched flow rate limiter allows you to rate limit the flows destined to the RP for forwarding and replication. For a given multicast traffic flow, if at least one outgoing Layer 3 interface is multilayer switched, and at least one outgoing interface is not multilayer switched (no H-bit set for hardware switching), the particular flow is considered partially switched, or partial-SC (partial shortcut). The outgoing interfaces that have the H-bit flag are switched in hardware and the remaining traffic is switched in software through the RP. For this reason, it may be desirable to rate limit the flow destined to the RP for forwarding and replication, which might otherwise increase CPU utilization.

The multicast directly connected rate limiter limits the multicast packets from directly connected sources.

This example shows how to rate limit the multicast packets to 30000 pps with a burst of 30:

```
Router(config)# mls rate-limit multicast ipv4 connected 30000 30
```

The **ip-option** keyword and the ip-option rate limiter are not supported in PFC3A mode.

This example shows how to set the rate limiters for the IPv4 multicast packets failing the uRPF check:

```
Router(config)# mls rate-limit multicast ipv4 non-rpf 100
```

This example shows how to rate limit the multicast FIB miss packets to 10000 pps with a burst of 10:

```
Router(config)# mls rate-limit multicast ipv4 fib-miss 10000 10
```

This example shows how to rate limit the partial shortcut flows to 20000 pps with a burst of 20 packets:

```
Router(config)# mls rate-limit multicast ipv4 partial 20000 20
```

This example shows how to rate limit the multicast packets to 30000 pps with a burst of 20:

```
Router(config)# mls rate-limit multicast ipv4 connected 30000 20
```

This example shows how to rate limit IGMP-snooping traffic:

```
Router(config)# mls rate-limit multicast ipv4 igmp 20000 40
```

IPv6 Multicast

This rate limiter limits the IPv6 multicast packets. Table 52-1 lists the IPv6 rate limiters and the class of traffic that each rate limiter serves.

Table 52-1 IPv6 Rate Limiters

Rate Limiter	Traffic Classes to be Rate Limited
Connected	Directly connected source traffic
Default-drop	* (*, G/m) SSM * (*, G/m) SSM non-rpf
Route-control	* (*, FF02::X/128)
Starg-bridge	* (*, G/128) SM * SM non-rpf traffic when (*, G) exists
Starg-M-bridge	* (*, G/m) SM * (*, FF/8) * SM non-rpf traffic when (*, G) doesn't exist

You can configure rate limiters for IPv6 multicast traffic using one of the following methods:

- Direct association of the rate limiters for a traffic class—Select a rate and associate the rate with a rate limiter. This example shows how to pick a rate of 1000 pps and 20 packets per burst and associate the rate with the **default-drop** rate limiter:

  ```
  Router(config)# mls rate-limit multicast ipv6 default-drop 1000 20
  ```

- Static sharing of a rate limiter with another preconfigured rate limiter—When there are not enough adjacency-based rate limiters available, you can share a rate limiter with an already configured rate limiter (target rate limiter). This example shows how to share the **route-cntl** rate limiter with the **default-drop** target rate limiter:

  ```
  Router(config)# mls rate-limit multicast ipv6 route-cntl share default-drop
  ```

  If the target rate limiter is not configured, a message is displayed that indicates that the target rate limiter must be configured for it to be shared with other rate limiters.

- Dynamic sharing of rate limiters—If you are not sure about which rate limiter to share with, use the **share auto** keywords to enable dynamic sharing. When you enable dynamic sharing, the system selects a preconfigured rate limiter and shares the given rate limiter with the preconfigured rate limiter. This example shows how to choose dynamic sharing for the route-cntrl rate limiter:

  ```
  Router(config)# mls rate-limit multicast ipv6 route-cntl share auto
  ```

This example shows how to set the rate limiters for the IPv6 multicast packets from a directly connected source:

```
Router(config)# mls rate-limit multicast ipv6 connected 1500 20
```

This example shows how to configure a direct association of the rate limiters for a traffic class:

```
Router(config)# mls rate-limit multicast ipv6 default-drop 1000 20
```

This example shows how to configure the static sharing of a rate limiter with another preconfigured rate limiter:

```
Router(config)# mls rate-limit multicast ipv6 route-cntl share default-drop
```

This example shows how to enable dynamic sharing for the route control rate limiter:

```
Router(config)# mls rate-limit multicast ipv6 route-cntl share auto
```

DoS Protection Default Configuration

Table 52-2 shows the DoS protection default configuration for the PFC3 hardware-based rate limiters.

Table 52-2 PFC3 Hardware-based Rate Limiter Default Setting

Rate Limiter	Default Status (ON/OFF)	Default Value
Ingress/Egress ACL Bridged Packets	OFF	
RPF Failures	ON	100 pps, burst of 10 packets
FIB Receive cases	OFF	
FIB Glean Cases	OFF	
Layer 3 Security features	OFF	
ICMP Redirect	OFF	
ICMP Unreachable	ON	100 pps, burst of 10 packets
VACL Log	ON	2000 pps, burst of 10 packets
TTL Failure	OFF	
MTU Failure	OFF	
Layer 2 PDU	OFF	
Layer 2 Protocol Tunneling	OFF	
IP Errors	ON	100 pps, burst of 10 packets
Multicast IGMP	OFF	
Multicast FIB-Miss	ON	100000 pps, burst of 100 packets
Multicast Partial-SC	ON	100000 pps, burst of 100 packets
Multicast Directly Connected	OFF	
Multicast Non-RPF	OFF	
Multicast IPv6	ON	If the *packets-in-burst* is not set, a default of **100** is programmed for multicast cases.

DoS Protection Configuration Guidelines and Restrictions

When configuring DoS protection, follow these CPU rate limiter guidelines and restrictions:

Note For the CoPP guidelines and restrictions, see the "CoPP Configuration Guidelines and Restrictions" section on page 53-2.

- Do not use these rate limiters if multicast is enabled in systems configured with a PFC3A:
  - TTL failure
  - MTU failure
- These rate limiters are not supported in PFC3A mode:
  - Unicast IP options
  - Multicast IP options
- These are Layer 2 rate limiters:
  - Layer 2 PDUs
  - Layer 2 protocol tunneling
  - Layer 2 Multicast IGMP
- There are eight Layer 3 registers and two Layer 2 registers that can be used as CPU rate limiters.
- Do not use the CEF receive limiter if CoPP is being used. The CEF receive limiter will override the CoPP traffic.
- Rate limiters override the CoPP traffic.
- Configured rate limits is applied to each forwarding engine (except for the Layer 2 hardware rate limiter which is applied globally).
- Layer 2 rate limiters are not supported in truncated mode.
- The following restrictions apply when using the ingress and egress ACL-bridged packet rate limiters:
  - The ingress and egress ACL-bridged packet rate limiter is available for unicast traffic only.
  - The ingress and egress ACL-bridged packet rate limiters share a single rate-limiter register. If you enable the ACL-bridge ingress and egress rate limiters, both the ingress and the egress ACLs must share the same rate-limiter value.
- Use the **mls rate-limit unicast** command to rate limit unicast traffic.
- Use the **mls rate-limit multicast** command to rate limit multicast traffic.
- Use the **mls rate-limit multicast layer 2** command to rate limit Layer 2 multicast traffic.

Monitoring Packet Drop Statistics

You can capture the incoming or outgoing traffic on an interface and send a copy of this traffic to an external interface for monitoring by a traffic analyzer. To capture traffic and forward it to an external interface, use the **monitor session** command.

When capturing traffic, these restrictions apply:

- The incoming captured traffic is not filtered.
- The incoming captured traffic is not rate limited to the capture destination.

Monitoring Dropped Packets Using Monitor Session Commands

This example shows how to use the **monitor session** command to capture and forward traffic to an external interface:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router(config)# monitor session 1 source vlan 44 both
Router(config)# monitor session 1 destination interface g9/1
Router(config)# end
Router#
2w0d: %SYS-5-CONFIG_I: Configured from console by console
```

This example shows how to use the **show monitor session** command to display the destination port location:

```
Router# show monitor session 1
Session 1
---------
Source Ports:
    RX Only:       None
    TX Only:       None
    Both:          None
Source VLANs:
    RX Only:       None
    TX Only:       None
    Both:          44
Destination Ports: Gi9/1
Filter VLANs:      None
```

Monitoring Dropped Packets Using show tcam interface Command

All modes except PFC3A mode support ACL hit counters in hardware. You can use the **show tcam interface** command to display each entry in the ACL TCAM.

This example shows how to use the **show tcam interface** command to display the number of times the entry was hit:

```
Router# show tcam interface fa5/2 acl in ip detail

-------------------------------------------------------------------------------------------------------------
DPort - Destination Port   SPort - Source Port       TCP-F - U -URG Pro    - Protocol
I     - Inverted LOU       TOS   - TOS Value               - A -ACK rtr    - Router
MRFM  - M -MPLS Packet     TN    - T -Tcp Control          - P -PSH COD    - C -Bank Care Flag
      - R -Recirc. Flag          - N -Non-cachable         - R -RST        - I -OrdIndep. Flag
      - F -Fragment Flag   CAP   - Capture Flag            - S -SYN        - D -Dynamic Flag
      - M -More Fragments  F-P   - FlowMask-Prior.         - F -FIN T      - V(Value)/M(Mask)/R(Result)
X     - XTAG               (*)   - Bank Priority
-------------------------------------------------------------------------------------------------------------


Interface: 1018   label: 1   lookup_type: 0
protocol: IP   packet-type: 0

+-+-----+---------------+---------------+---------------+---------------+-------+---+----+-+---+--+---+---+
|T|Index|  Dest Ip Addr | Source Ip Addr|     DPort     |     SPort     | TCP-F|Pro|MRFM|X|TOS|TN|COD|F-P|
+-+-----+---------------+---------------+---------------+---------------+------+---+----+-+---+--+---+---+
 V 18396         0.0.0.0         0.0.0.0       P=0             P=0       ------   0 ---- 0   0 -- --- 0-0
 M 18404         0.0.0.0         0.0.0.0         0               0                0 ---- 0   0
 R rslt: L3_DENY_RESULT                rtr_rslt: L3_DENY_RESULT


 V 36828         0.0.0.0         0.0.0.0       P=0             P=0        ------  0 ---- 0   0 -- --- 0-0
 M 36836         0.0.0.0         0.0.0.0         0               0                0 ---- 0   0
 R rslt: L3_DENY_RESULT (*)            rtr_rslt: L3_DENY_RESULT (*)
Router#
```

You can also use the TTL and IP options counters to monitor the performance of the Layer 3 forwarding engine.

This example shows how to use the **show mls statistics** command to display packet statistics and errors associated with the Layer 3 forwarding engine:

```
Router# show mls statistics

Statistics for Earl in Module 6

L2 Forwarding Engine
  Total packets Switched                : 25583421

L3 Forwarding Engine
  Total packets L3 Switched             : 25433414 @ 24 pps

  Total Packets Bridged                 : 937860
  Total Packets FIB Switched            : 23287640
  Total Packets ACL Routed              : 0
  Total Packets Netflow Switched        : 0
  Total Mcast Packets Switched/Routed   : 96727
  Total ip packets with TOS changed     : 2
  Total ip packets with COS changed     : 2
  Total non ip packets COS changed      : 0
  Total packets dropped by ACL          : 33
  Total packets dropped by Policing     : 0

Errors
  MAC/IP length inconsistencies         : 0
  Short IP packets received             : 0
  IP header checksum errors             : 0
  TTL failures                          : 0
<----------------- TTL counters
  MTU failures                          : 0
<------------------MTU failure counters

Total packets L3 Switched by all Modules: 25433414 @ 24 pps
```

Monitoring Dropped Packets Using VACL Capture

The VACL capture feature allows you to direct traffic to ports configured to forward captured traffic. The capture action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets. Only forwarded packets can be captured.

You can use VACL capture to assign traffic from each VLAN to a different interface.

VACL capture does not allow you to send one type of traffic, such as HTTP, to one interface and another type of traffic, such as DNS, to another interface. Also, VACL capture granularity is only applicable to traffic switched locally; you cannot preserve the granularity if you direct traffic to a remote switch.

This example shows how to use VACL capture to capture and forward traffic to a local interface:

```
Router(config-if)# switchport capture
Router(config-if)# switchport capture allowed vlan add 100
```

Displaying Rate-Limiter Information

The **show mls rate-limit** command displays information about the configured rate limiters.

The **show mls rate-limit usage** command displays the hardware register that is used by a rate-limiter type. If the register is not used by any rate-limiter type, Free is displayed in the output. If the register is used by a rate-limiter type, Used and the rate-limiter type are displayed.

In the command output, the rate-limit status could be one of the following:

- On indicates that a rate for that particular case has been set.
- Off indicates that the rate-limiter type has not been configured, and the packets for that case are not rate limited.
- On/Sharing indicates that a particular case (not manually configured) is affected by the configuration of another rate limiter belonging to the same sharing group.
- A hyphen indicates that the multicast partial-SC rate limiter is disabled.

In the command output, the rate-limit sharing indicates the following information:

- Whether sharing is static or dynamic
- Group dynamic sharing codes

To display the configured rate limiters, use the **show mls rate-limit** command:

```
Router# show mls rate-limit
 Sharing Codes: S - static, D - dynamic
 Codes dynamic sharing: H - owner (head) of the group, g - guest of the group

   Rate Limiter Type       Status     Packets/s   Burst  Sharing
 ---------------------   ----------   ---------   -----  -------
         MCAST NON RPF   Off                  -       -     -
        MCAST DFLT ADJ   On              100000     100  Not sharing
      MCAST DIRECT CON   Off                  -       -     -
        ACL BRIDGED IN   Off                  -       -     -
       ACL BRIDGED OUT   Off                  -       -     -
           IP FEATURES   Off                  -       -     -
          ACL VACL LOG   On                2000       1  Not sharing
           CEF RECEIVE   Off                  -       -     -
             CEF GLEAN   Off                  -       -     -
      MCAST PARTIAL SC   On              100000     100  Not sharing
        IP RPF FAILURE   On                 100      10  Group:0 S
           TTL FAILURE   Off                  -       -     -
 ICMP UNREAC. NO-ROUTE   On                 100      10  Group:0 S
 ICMP UNREAC. ACL-DROP   On                 100      10  Group:0 S
         ICMP REDIRECT   Off                  -       -     -
           MTU FAILURE   Off                  -       -     -
       MCAST IP OPTION   Off                  -       -     -
       UCAST IP OPTION   Off                  -       -     -
           LAYER_2 PDU   Off                  -       -     -
            LAYER_2 PT   Off                  -       -     -
             IP ERRORS   On                 100      10  Group:0 S
           CAPTURE PKT   Off                  -       -     -
            MCAST IGMP   Off                  -       -     -
 MCAST IPv6 DIRECT CON   Off                  -       -     -
 MCAST IPv6 *G M BRIDG   Off                  -       -     -
  MCAST IPv6 *G BRIDGE   Off                  -       -     -
  MCAST IPv6 SG BRIDGE   Off                  -       -     -
 MCAST IPv6 ROUTE CNTL   Off                  -       -     -
  MCAST IPv6 DFLT DROP   Off                  -       -     -
 MCAST IPv6 SECOND. DR   Off                  -       -     -
Router#
```

To display the usage of the hardware rate limiters, use the **show mls rate-limit usage** command:

```
Router# show mls rate-limit usage
                             Rate Limiter Type     Packets/s   Burst
                           ---------------------   ---------   -----
Layer3 Rate Limiters:
             RL# 0: Free                      -           -       -
             RL# 1: Free                      -           -       -
```

```
                          RL# 2: Free                       -            -       -
                          RL# 3: Used
                                                MCAST DFLT ADJ      100000     100
                          RL# 4: Free                       -            -       -
                          RL# 5: Free                       -            -       -
                          RL# 6: Used
                                                IP RPF FAILURE         100      10
                                         ICMP UNREAC. NO-ROUTE         100      10
                                         ICMP UNREAC. ACL-DROP         100      10
                                                     IP ERRORS         100      10
                          RL# 7: Used
                                                  ACL VACL LOG        2000       1
                          RL# 8: Rsvd for capture           -            -       -

Layer2 Rate Limiters:
                          RL# 9: Reserved
                          RL#10: Reserved
                          RL#11: Free                       -            -       -
                          RL#12: Free                       -            -       -
Router#
```

Configuring Sticky ARP

Sticky ARP prevents MAC address spoofing by ensuring that ARP entries (IP address, MAC address, and source VLAN) do not get overridden. The switch maintains ARP entries in order to forward traffic to end devices or other switches. ARP entries are usually updated periodically or modified when ARP broadcasts are received. During an attack, ARP broadcasts are sent using a spoofed MAC address (with a legitimate IP address) so that the switch learns the legitimate IP address with the spoofed MAC address and begins to forward traffic to that MAC address. With sticky ARP enabled, the switch learns the ARP entries and does not accept modifications received through ARP broadcasts. If you attempt to override the sticky ARP configuration, you will receive an error message. For a complete description of the system error messages, see the *Catalyst 6500 Series Switch Cisco IOS System Message Guide*, Release 12.2SX at this URL:

http://www.cisco.com/en/US/docs/ios/12_2sx/system/messages/122sxsms.html

To configure sticky ARP on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the interface on which sticky ARP is applied.
Step 2	`Router(config-if)#` **`ip sticky-arp`**	Enables sticky ARP.
	`Router(config-if)#` **`no ip sticky-arp ignore`**	Removes the previously configured sticky ARP command.
Step 3	`Router(config-if)#` **`ip sticky-arp ignore`**	Disables sticky ARP.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable sticky ARP on interface 5/1:

```
Router# configure terminal
Router(config)# interface gigabitethernet 5/1
Router(config-if)# ip sticky-arp
Router(config-if)# end
Router#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Control Plane Policing

This chapter describes how to configure control plane policing (CoPP) with Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Control Plane Policing

The traffic managed by the RP is divided into three functional components or *planes*:

- Data plane
- Management plane
- Control plane

The control plane policing (CoPP) feature increases security on the switch by protecting the RP from unnecessary or DoS traffic and giving priority to important control plane and management traffic. The PFC3 and DFC3 provide hardware support for CoPP. CoPP works with the PFC3 rate limiters.

The PFC3 supports the built-in "special case" rate limiters that can be used when an ACL cannot classify particular scenarios, such as IP options cases, TTL and MTU failure cases, packets with errors, and multicast packets. When enabling the special-case rate limiters, the special-case rate limiters override the CoPP policy for packets matching the rate-limiter criteria.

The majority of traffic managed by the RP is handled by way of the control and management planes. You can use CoPP to protect the control and management planes, and ensure routing stability, reachability, and packet delivery. CoPP uses a dedicated control plane configuration through the modular QoS CLI (MQC) to provide filtering and rate-limiting capabilities for the control plane packets.

CoPP Default Configuration

CoPP is disabled by default.

CoPP Configuration Guidelines and Restrictions

When configuring CoPP, follow these guidelines and restrictions:

- CoPP is not be enabled in hardware unless you have enabled PFC QoS globally with the **mls qos** command. If you do not enter the **mls qos** command, CoPP is not hardware-accelerated.
- CoPP is supported in software for the following:
  - Multicast traffic.
  - Broadcast traffic.

    Note The combination of ACLs, traffic storm control, and CoPP software protection provides protection against broadcast DoS attacks.

  - CoPP-policy ACLs configured with the **log** keyword. To avoid software-supported CoPP processing, do not use the **log** keyword in CoPP-policy ACLs.
  - When there is a large QoS configuration for other interfaces, you can run out of TCAM space. When this situation occurs, CoPP may be performed entirely in software and result in performance degradation and CPU cycle consumption. Enter the **show tcam utilization** command to verify the TCAM utilization.
  - CoPP policies configured with the **match protocol arp** command.
  - Release 12.2(33)SXI4 and later releases support CoPP policies configured with the **match access-group** *arp_acl* command.
- CoPP is performed on a per-forwarding-engine basis and software CoPP is performed on an aggregate basis.
- CoPP does not support MAC ACLs.
- CoPP does not support non-IP classes except for the default non-IP class. ACLs can be used instead of non-IP classes to drop non-IP traffic, and the default non-IP CoPP class can be used to limit to non-IP traffic that reaches the RP CPU.

- In PFC3A mode, egress QoS and CoPP cannot be configured at the same time. A warning message is displayed to inform you that egress QoS and CoPP cannot be configured at the same time.
- You must ensure that the CoPP policy does not filter critical traffic such as routing protocols or interactive access to the switches. Filtering this traffic could prevent remote access to the switch, requiring a console connection.
- The PFC3 supports built-in special-case rate limiters, which are useful for situations where an ACL cannot be used (for example, TTL, MTU, and IP options). When you enable the special-case rate limiters, you should be aware that the special-case rate limiters will override the CoPP policy for packets matching the rate-limiter criteria.
- Neither egress CoPP nor silent mode is supported. CoPP is only supported on ingress (service-policy output CoPP cannot be applied to the control plane interface).
- ACE hit counters in hardware are only for ACL logic. You can rely on software ACE hit counters and the **show access-list**, **show policy-map control-plane**, and **show mls ip qos** commands to troubleshoot evaluate CPU traffic.

Configuring CoPP

CoPP uses MQC to define traffic classification criteria and to specify the configurable policy actions for the classified traffic. You must first identify the traffic to be classified by defining a class map. The class map defines packets for a particular traffic class. After you have classified the traffic, you can create policy maps to enforce policy actions for the identified traffic. The **control-plane** global configuration command allows the CoPP service policies to be directly attached to the control plane.

For information on how to define the traffic classification criteria, see the “Defining Traffic Classification” section on page 53-5.

To configure CoPP, perform this task:

	Command	Purpose
Step 1	`Router(config)# ` **`mls qos`**	Enables MLS QoS globally.
Step 2	`Router(config)# ` **`ip access-list extended`** *`access-list-name`* `Router(config-ext-nacl)# {`**`permit`**`	`**`deny`**`}` *`protocol`* **`source`** *`source-wildcard`* **`destination`** *`destination-wildcard`* `[`**`precedence`** *`precedence`*`] [`**`tos`** *`tos`*`] [`**`established`**`] [`**`log`**`	`**`log-input`**`] [`**`time-range`** *`time-range-name`*`] [`**`fragments`**`]`	Defines ACLs to match traffic: • **permit** sets the conditions under which a packet passes a named IP access list. • **deny** sets the conditions under which a packet does not pass a named IP access list. **Note** You must configure ACLs in most cases to identify the important or unimportant traffic.

	Command	Purpose
Step 3	Router(config)# **class-map** *traffic-class-name* Router(config-cmap)# **match** {**ip precedence**} |{**ip dscp**} | *access-group*	Defines the packet classification criteria. Use the **match** statements to identify the traffic associated with the class.
	Router(config)# **policy-map** *service-policy-name* Router(config-pmap)# **class** *traffic-class-name* Router(config-pmap-c)# **police** {*bits-per-second* [*normal-burst-bytes*] [*maximum-burst-bytes*] [**pir** *peak-rate-bps*]} | [**conform-action** *action*] [**exceed-action** *action*] [**violate-action** *action*]	Defines a service policy map. Use the **class** *traffic-class-name* command to associate classes to the service policy map. Use the **police** statements to associate actions to the service policy map.
Step 4	Router(config)# **control-plane** Router(config-cp)#	Enters the control plane configuration mode.
Step 5	Router(config-cp)# **service-policy input** *service-policy-name*	Applies the QoS service policy to the control plane.

When defining the packet classification criteria, follow these guidelines and restrictions:

- To avoid matching the filtering and policing that are configured in a subsequent class, configure policing in each class. CoPP does not apply the filtering in a class that does not contain a police command. A class without a police command matches no traffic.
- The ACLs used for classification are QoS ACLs. The supported QoS ACLs are IP standard, extended, and named.
- These are the only match types supported:
  - **ip precedence**
  - **ip dscp**
  - **access-group**
- Only IP ACLs are supported in hardware.
- MAC-based matching is done in software only.
- You can enter one **match** command in a single class map only.

When defining the service policy, the **police** policy-map action is the only supported action.

When applying the service policy to the control plane, the **input** direction is only supported.

Monitoring CoPP

You can enter the **show policy-map control-plane** command for developing site-specific policies, monitoring statistics for the control plane policy, and troubleshooting CoPP. This command displays dynamic information about the actual policy applied, including rate information and the number of bytes (and packets) that conformed or exceeded the configured policies both in hardware and in software.

The output of the **show policy-map control-plane** command is as follows:

```
Router# show policy-map control-plane
Control Plane Interface
  Service policy CoPP-normal
Hardware Counters:
class-map: CoPP-normal (match-all)
```

```
        Match: access-group 130
        police :
          96000 bps 3000 limit 3000 extended limit
        Earl in slot 3 :
          0 bytes
          5 minute offered rate 0 bps
          aggregate-forwarded 0 bytes action: transmit
          exceeded 0 bytes action: drop
          aggregate-forward 0 bps exceed 0 bps
        Earl in slot 5 :
          0 bytes
          5 minute offered rate 0 bps
          aggregate-forwarded 0 bytes action: transmit
          exceeded 0 bytes action: drop
          aggregate-forward 0 bps exceed 0 bps

Software Counters:
    Class-map: CoPP-normal (match-all) 0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 130
      police:
        96000 bps, 3125 limit, 3125 extended limit
        conformed 0 packets, 0 bytes; action: transmit
        exceeded 0 packets, 0 bytes; action: drop
        conformed 0 bps, exceed 0 bps, violate 0 bps
Router#
```

To display the hardware counters for bytes dropped and forwarded by the policy, enter the **show mls qos ip** command:

```
Router# show mls qos ip
 QoS Summary [IP]:      (* - shared aggregates, Mod - switch module)

Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                             Id         Id
-------------------------------------------------------------------------------
CPP  5  In CoPP-normal    0    1   dscp  0         505408       83822272
CPP  9  In CoPP-normal    0    4   dscp  0              0              0
Router#
```

To display the CoPP access list information, enter the **show access-lists coppacl-bgp** command:

```
Router# show access-lists coppacl-bgp
Extended IP access list coppacl-bgp
10 permit tcp host 47.1.1.1 host 10.9.9.9 eq bgp (4 matches)
20 permit tcp host 47.1.1.1 eq bgp host 10.9.9.9
30 permit tcp host 10.86.183.120 host 10.9.9.9 eq bgp (1 match)
40 permit tcp host 10.86.183.120 eq bgp host 10.9.9.9
```

Defining Traffic Classification

The following sections contain information on how to classify CoPP traffic:

- Traffic Classification Overview, page 53-6
- Traffic Classification Guidelines, page 53-7
- Sample Basic ACLs for CoPP Traffic Classification, page 53-7

Traffic Classification Overview

You can define any number of classes, but typically traffic is grouped into classes that are based on relative importance. The following provides a sample grouping:

- Border Gateway Protocol (BGP)—Traffic that is crucial to maintaining neighbor relationships for BGP routing protocol, for example, BGP keepalives and routing updates. Maintaining BGP routing protocol is crucial to maintaining connectivity within a network or to a service provider. Sites that do not run BGP do not need to use this class.
- Interior Gateway Protocol (IGP)—Traffic that is crucial to maintaining IGP routing protocols, for example, open shortest path first OSPF, enhanced interior gateway routing protocol (EIGRP), and routing information protocol (RIP). Maintaining IGP routing protocols is crucial to maintaining connectivity within a network.
- Management—Necessary, frequently used traffic that is required during day-to-day operations. For example, traffic used for remote network access, and Cisco IOS image upgrades and management, such as Telnet, secure shell (SSH), network time protocol (NTP), simple network management protocol (SNMP), terminal access controller access control system (TACACS), hypertext transfer protocol (HTTP), trivial file transfer protocol (TFTP), and file transfer protocol (FTP).
- Reporting—Traffic used for generating network performance statistics for the purpose of reporting. For example, using Cisco IOS IP service level agreements (SLAs) to generate ICMP with different DSCP settings in order to report on response times within different QoS data classes.
- Monitoring—Traffic used for monitoring a switch. Traffic should be permitted but should never be a risk to the switch; with CoPP, this traffic can be permitted but limited to a low rate. For example, ICMP echo request (ping) and traceroute.
- Critical Applications—Critical application traffic that is specific and crucial to a particular customer environment. Traffic included in this class should be tailored specifically to the required application requirements of the user (in other words, one customer may use multicast, while another uses IPsec or generic routing encapsulation (GRE). For example, GRE, hot standby router protocol (HSRP), virtual router redundancy protocol (VRRP), session initiation protocol (SIP), data link switching (DLSw), dynamic host configuration protocol (DHCP), multicast source discovery protocol (MSDP), Internet group management protocol (IGMP), protocol independent multicast (PIM), multicast traffic, and IPsec.
- Layer 2 Protocols—Traffic used for address resolution protocol (ARP). Excessive ARP packets can potentially monopolize RP resources, starving other important processes; CoPP can be used to rate limit ARP packets to prevent this situation. Currently, ARP is the only Layer 2 protocol that can be specifically classified using the match protocol classification criteria.
- Undesirable—Explicitly identifies bad or malicious traffic that should be unconditionally dropped and denied access to the RP. The undesirable classification is particularly useful when known traffic destined for the switch should always be denied and not placed into a default category. If you explicitly deny traffic, then you can enter **show** commands to collect approximate statistics on the denied traffic and estimate its rate.
- Default—All remaining traffic destined for the RP that has not been identified. MQC provides the default class, so the user can specify the treatment to be applied to traffic not explicitly identified in the other user-defined classes. This traffic has a highly reduced rate of access to the RP. With a default classification in place, statistics can be monitored to determine the rate of otherwise unidentified traffic destined for the control plane. After this traffic is identified, further analysis can be performed to classify it and, if needed, the other CoPP policy entries can be updated to accomodate this traffic.

After you have classified the traffic, the ACLs build the classes of traffic that are used to define the policies. For sample basic ACLs for CoPP classification, see the "Sample Basic ACLs for CoPP Traffic Classification" section on page 53-7.

Traffic Classification Guidelines

When defining traffic classification, follow these guidelines and restrictions:

- Before you develop the actual CoPP policy, you must identify and separate the required traffic into different classes. Traffic is grouped into nine classes that are based on relative importance. The actual number of classes needed might differ and should be selected based on your local requirements and security policies.
- You do not have to define policies that match bidirectionally. You only need to identify traffic unidirectionally (from the network to the RP) since the policy is applied on ingress only.

Sample Basic ACLs for CoPP Traffic Classification

This section shows sample basic ACLs for CoPP classification. In the samples, the commonly required traffic is identified with these ACLs:

- ACL 120—Critical traffic
- ACL 121—Important traffic
- ACL 122—Normal traffic
- ACL 123—Explicitly denies unwanted traffic
- ACL 124—All other traffic

This example shows how to define ACL 120 for critical traffic:

```
Router(config)# access-list 120 remark CoPP ACL for critical traffic
```

This example shows how to allow BGP from a known peer to this switch's BGP TCP port:

```
Router(config)# access-list 120 permit tcp host 47.1.1.1 host 10.9.9.9 eq bgp
```

This example shows how to allow BGP from a peer's BGP port to this switch:

```
Router(config)# access-list 120 permit tcp host 47.1.1.1 eq bgp host 10.9.9.9
Router(config)# access-list 120 permit tcp host 10.86.183.120 host 10.9.9.9 eq bgp
Router(config)# access-list 120 permit tcp host 10.86.183.120 eq bgp host 10.9.9.9
```

This example shows how to define ACL 121 for the important class:

```
Router(config)# access-list 121 remark CoPP Important traffic
```

This example shows how to permit return traffic from TACACS host:

```
Router(config)# access-list 121 permit tcp host 1.1.1.1 host 10.9.9.9 established
```

This example shows how to permit SSH access to the switch from a subnet:

```
Router(config)# access-list 121 permit tcp 10.0.0.0 0.0.0.255 host 10.9.9.9 eq 22
```

This example shows how to allow full access for Telnet to the switch from a host in a specific subnet and police the rest of the subnet:

```
Router(config)# access-list 121 deny tcp host 10.86.183.3 any eq telnet
Router(config)# access-list 121 permit tcp 10.86.183.0 0.0.0.255 any eq telnet
```

This example shows how to allow SNMP access from the NMS host to the switch:

```
Router(config)# access-list 121 permit udp host 1.1.1.2 host 10.9.9.9 eq snmp
```

This example shows how to allow the switch to receive NTP packets from a known clock source:

```
Router(config)# access-list 121 permit udp host 1.1.1.3 host 10.9.9.9 eq ntp
```

This example shows how to define ACL 122 for the normal traffic class:

```
Router(config)# access-list 122 remark CoPP normal traffic
```

This example shows how to permit switch-originated traceroute traffic:

```
Router(config)# access-list 122 permit icmp any any ttl-exceeded
Router(config)# access-list 122 permit icmp any any port-unreachable
```

This example shows how to permit receipt of responses to the switch that originated the pings:

```
Router(config)# access-list 122 permit icmp any any echo-reply
```

This example shows how to allow pings to the switch:

```
Router(config)# access-list 122 permit icmp any any echo
```

This example shows how to define ACL 123 for the undesirable class.

```
Router(config)# access-list 123 remark explicitly defined "undesirable" traffic
```

Note In the following example, ACL 123 is a permit entry for classification and monitoring purposes, and traffic is dropped as a result of the CoPP policy.

This example shows how to permit all traffic destined to UDP 1434 for policing:

```
Router(config)# access-list 123 permit udp any any eq 1434
```

This example shows how to define ACL 124 for all other traffic:

```
Router(config)# access-list 124 remark rest of the IP traffic for CoPP
Router(config)# access-list 124 permit ip any any
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring DHCP Snooping

This chapter describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

- Understanding DHCP Snooping, page 54-1
- Default Configuration for DHCP Snooping, page 54-7
- DHCP Snooping Configuration Restrictions and Guidelines, page 54-7
- Configuring DHCP Snooping, page 54-9

Understanding DHCP Snooping

These sections describe the DHCP snooping feature:

- Overview of DHCP Snooping, page 54-2
- Trusted and Untrusted Sources, page 54-2
- DHCP Snooping Binding Database, page 54-3
- Packet Validation, page 54-3
- DHCP Snooping Option-82 Data Insertion, page 54-3
- Overview of the DHCP Snooping Database Agent, page 54-5

- DHCP Snooping Host Tracking, page 54-6

Overview of DHCP Snooping

DHCP snooping is a security feature that acts like a firewall between untrusted hosts and trusted DHCP servers. The DHCP snooping feature performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Rate-limits DHCP traffic from trusted and untrusted sources.
- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Utilizes the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

Other security features, such as dynamic ARP inspection (DAI), also use information stored in the DHCP snooping binding database.

DHCP snooping is enabled on a per-VLAN basis. By default, the feature is inactive on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

The DHCP snooping feature is implemented in software on the route processor (RP). Therefore, all DHCP messages for enabled VLANs are intercepted in the PFC and directed to the RP for processing.

Trusted and Untrusted Sources

The DHCP snooping feature determines whether traffic sources are trusted or untrusted. An untrusted source may initiate traffic attacks or other hostile actions. To prevent such attacks, the DHCP snooping feature filters messages and rate-limits traffic from untrusted sources.

In an enterprise network, devices under your administrative control are trusted sources. These devices include the switches, routers, and servers in your network. Any device beyond the firewall or outside your network is an untrusted source. Host ports and unknown DHCP servers are generally treated as untrusted sources.

A DHCP server that is on your network without your knowledge on an untrusted port is called a *spurious DHCP server*. A spurious DHCP server is any piece of equipment that is loaded with DHCP server enabled. Some examples are desktop systems and laptop systems that are loaded with DHCP server enabled, or wireless access points honoring DHCP requests on the wired side of your network. If spurious DHCP servers remain undetected, you will have difficulties troubleshooting a network outage. You can detect spurious DHCP servers by sending dummy DHCPDISCOVER packets out to all of the DHCP servers so that a response is sent back to the switch.

In a service provider environment, any device that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In the switch, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to devices (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.

Note For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces, as untrusted DHCP messages will be forwarded only to trusted interfaces.

DHCP Snooping Binding Database

The DHCP snooping binding database is also referred to as the DHCP snooping binding table.

The DHCP snooping feature dynamically builds and maintains the database using information extracted from intercepted DHCP messages. The database contains an entry for each untrusted host with a leased IP address if the host is associated with a VLAN that has DHCP snooping enabled. The database does not contain entries for hosts connected through trusted interfaces.

The DHCP snooping feature updates the database when the switch receives specific DHCP messages. For example, the feature adds an entry to the database when the switch receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the switch receives a DHCPRELEASE message from the host.

Each entry in the DHCP snooping binding database includes the MAC address of the host, the leased IP address, the lease time, the binding type, and the VLAN number and interface information associated with the host.

Packet Validation

The switch validates DHCP packets received on the untrusted interfaces of VLANs with DHCP snooping enabled. The switch forwards the DHCP packet unless any of the following conditions occur (in which case the packet is dropped):

- The switch receives a packet (such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet) from a DHCP server outside the network or firewall.
- The switch receives a packet on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match. This check is performed only if the DHCP snooping MAC address verification option is turned on.
- The switch receives a DHCPRELEASE or DHCPDECLINE message from an untrusted host with an entry in the DHCP snooping binding table, and the interface information in the binding table does not match the interface on which the message was received.
- The switch receives a DHCP packet that includes a relay agent IP address that is not 0.0.0.0.

To support trusted edge switches that are connected to untrusted aggregation-switch ports, you can enable the DHCP option-82 on untrusted port feature, which enables untrusted aggregation-switch ports to accept DHCP packets that include option-82 information. Configure the port on the edge switch that connects to the aggregation switch as a trusted port.

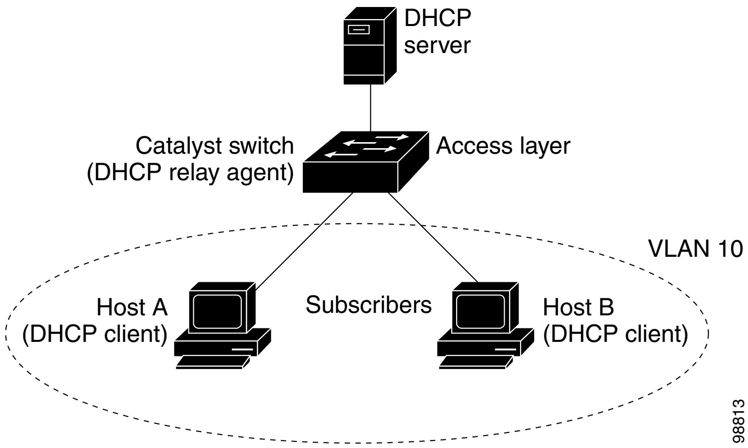
Note With the DHCP option-82 on untrusted port feature enabled, use dynamic ARP inspection on the aggregation switch to protect untrusted input interfaces.

DHCP Snooping Option-82 Data Insertion

In residential, metropolitan Ethernet-access environments, DHCP can centrally manage the IP address assignments for a large number of subscribers. When the DHCP snooping option-82 feature is enabled on the switch, a subscriber device is identified by the switch port through which it connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can be connected to the same port on the access switch and are uniquely identified.

Figure 54-1 is an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the switch at the access layer. Because the DHCP clients and their associated DHCP server do not reside on the same IP network or subnet, a DHCP relay agent is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.

Figure 54-1 DHCP Relay Agent in a Metropolitan Ethernet Network



When you enable the DHCP snooping information option-82 on the switch, this sequence of events occurs:

- The host (DHCP client) generates a DHCP request and broadcasts it on the network.
- When the switch receives the DHCP request, it adds the option-82 information in the packet. The option-82 information contains the switch MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption).
- If IEEE 802.1X port-based authentication is enabled, the switch will also add the host's 802.1X authenticated user identity information (the RADIUS attributes suboption) to the packet. See the "Understanding 802.1X Authentication with DHCP Snooping" section on page 60-11.
- If the IP address of the relay agent is configured, the switch adds the IP address in the DHCP packet.
- The switch forwards the DHCP request that includes the option-82 field to the DHCP server.
- The DHCP server receives the packet. If the server is option-82 capable, it can use the remote ID, or the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server then echoes the option-82 field in the DHCP reply.
- The DHCP server unicasts the reply to the switch if the request was relayed to the server by the switch. When the client and server are on the same subnet, the server broadcasts the reply. The switch verifies that it originally inserted the option-82 data by inspecting the remote ID and possibly the circuit ID fields. The switch removes the option-82 field and forwards the packet to the switch port that connects to the DHCP client that sent the DHCP request.

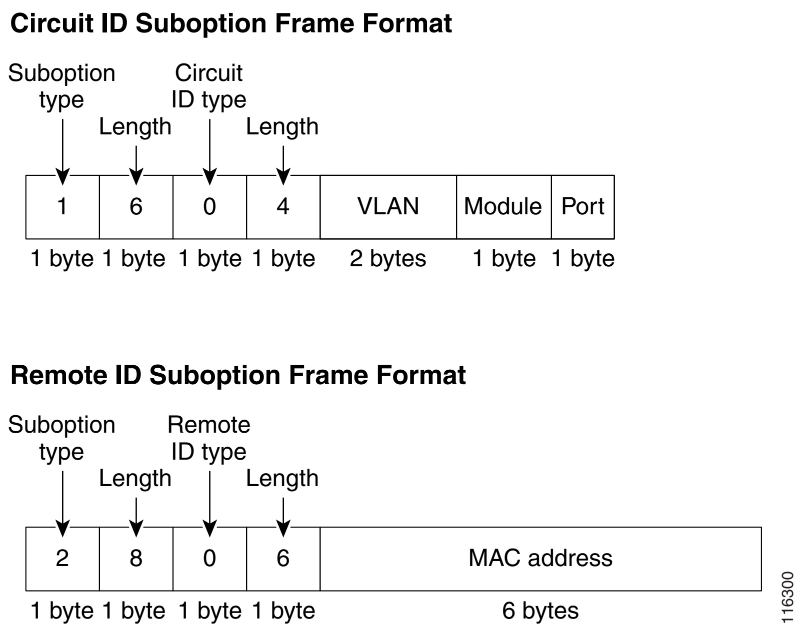
When the previously described sequence of events occurs, the values in these fields in Figure 54-2 do not change:

- Circuit ID suboption fields
  - Suboption type
  - Length of the suboption type

    - Circuit ID type
    - Length of the circuit ID type
- Remote ID suboption fields
    - Suboption type
    - Length of the suboption type
    - Remote ID type
    - Length of the circuit ID type

Figure 54-2 shows the packet formats for the remote ID suboption and the circuit ID suboption. The switch uses the packet formats when DHCP snooping is globally enabled and when the **ip dhcp snooping information option** global configuration command is entered. For the circuit ID suboption, the module field is the slot number of the module.

Figure 54-2 Suboption Packet Formats



Overview of the DHCP Snooping Database Agent

To retain the bindings across reloads, you must use the DHCP snooping database agent. Without this agent, the bindings established by DHCP snooping are lost upon reload, and connectivity is lost as well.

The database agent stores the bindings in a file at a configured location. Upon reload, the switch reads the file to build the database for the bindings. The switch keeps the file current by writing to the file as the database changes.

The format of the file that contains the bindings is as follows:

```
<initial-checksum>
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
<entry-1> <checksum-1>
```

```
<entry-2> <checksum-1-2>
...
...
<entry-n> <checksum-1-2-..-n>
END
```

Each entry in the file is tagged with a checksum that is used to validate the entries whenever the file is read. The <initial-checksum> entry on the first line helps distinguish entries associated with the latest write from entries that are associated with a previous write.

This is a sample bindings file:

```
3ebe1518
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
1.1.1.1 512 0001.0001.0005 3EBE2881 Gi1/1                e5e1e733
1.1.1.1 512 0001.0001.0002 3EBE2881 Gi1/1                4b3486ec
1.1.1.1 1536 0001.0001.0004 3EBE2881 Gi1/1               f0e02872
1.1.1.1 1024 0001.0001.0003 3EBE2881 Gi1/1               ac41adf9
1.1.1.1 1 0001.0001.0001 3EBE2881 Gi1/1                  34b3273e
END
```

Each entry holds an IP address, VLAN, MAC address, lease time (in hex), and the interface associated with a binding. At the end of each entry is a checksum that is based on all the bytes from the start of the file through all the bytes associated with the entry. Each entry consists of 72 bytes of data, followed by a space, followed by a checksum.

Upon bootup, when the calculated checksum equals the stored checksum, the switch reads entries from the file and adds the bindings to the DHCP snooping database. If the calculated checksum does not equal the stored checksum, the entry read from the file is ignored and so are all the entries following the failed entry. The switch also ignores all those entries from the file whose lease time has expired. (This is possible because the lease time might indicate an expired time.) An entry from the file is also ignored if the interface referred to in the entry no longer exists on the system, or if it is a router port or a DHCP snooping-trusted interface.

When the switch learns of new bindings or when it loses some bindings, the switch writes the modified set of entries from the snooping database to the file. The writes are performed with a configurable delay to batch as many changes as possible before the actual write happens. Associated with each transfer is a timeout after which a transfer is aborted if it is not completed. These timers are referred to as the write delay and abort timeout.

DHCP Snooping Host Tracking

Release 12.2(33)SXJ2 and later releases support DHCP snooping host tracking. The DHCP snooping host tracking feature implements a cache to learn VLAN and MAC addresses to port the mapping of clients from snooped DHCP request packets and uses this information to forward snooped DHCP reply packets.

This feature improves DHCP snooping packet processing performance for DHCP reply packets by not needing to lookup the hardware VLAN and MAC address table in order to determine the port on which to send the DHCP reply packets. This feature is useful in deployments where it is not possible to use the DHCP snooping information option along with DHCP (for example, when the server does not support DHCP information option). If DHCP is configured it takes hugher precedence than the DHCP snooping host tracking feature in determining the port on which to forward reply packets.

The DHCP snooping host tracking feature is off by default (see the "Enabling DHCP Snooping Host Tracking" section on page 54-11).

Default Configuration for DHCP Snooping

Table 54-1 shows all the default configuration values for each DHCP snooping option.

Table 54-1 Default Configuration Values for DHCP Snooping

Option	Default Value/State
DHCP snooping	Disabled
DHCP snooping host tracking feature	Disabled
DHCP snooping information option	Enabled
DHCP option-82 on untrusted port feature	Disabled
DHCP snooping limit rate	None
DHCP snooping trust	Untrusted
DHCP snooping vlan	Disabled
DHCP snooping spurious server detection	Disabled
DHCP snooping detect spurious interval	30 minutes

DHCP Snooping Configuration Restrictions and Guidelines

These sections provide DHCP snooping configuration restrictions and guidelines:

- DHCP Snooping Configuration Restrictions, page 54-7
- DHCP Snooping Configuration Guidelines, page 54-8
- Minimum DHCP Snooping Configuration, page 54-8

DHCP Snooping Configuration Restrictions

When configuring DHCP snooping, note these restrictions:

- The DHCP snooping database stores at least 8,000 bindings.
- When DHCP snooping is enabled, these Cisco IOS DHCP commands are not available on the switch:
  - **ip dhcp relay information check** global configuration command
  - **ip dhcp relay information policy** global configuration command
  - **ip dhcp relay information trust-all** global configuration command
  - **ip dhcp relay information option** global configuration command
  - **ip dhcp relay information trusted** interface configuration command

  If you enter these commands, the switch returns an error message, and the configuration is not applied.

DHCP Snooping Configuration Guidelines

When configuring DHCP snooping, follow these guidelines:

- DHCP snooping is not active until you enable the feature on at least one VLAN, and enable DHCP globally on the switch.
- Before globally enabling DHCP snooping on the switch, make sure that the devices acting as the DHCP server and the DHCP relay agent are configured and enabled.
- For DHCP server configuration information, see "Configuring DHCP" in the *Cisco IOS IP and IP Routing Configuration Guide* at:

  http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfdhcp.html
- If a Layer 2 LAN port is connected to a DHCP server, configure the port as trusted by entering the **ip dhcp snooping trust** interface configuration command.
- If a Layer 2 LAN port is connected to a DHCP client, configure the port as untrusted by entering the **no ip dhcp snooping trust** interface configuration command.
- You can enable DHCP snooping on private VLANs:
  - If DHCP snooping is enabled, any primary VLAN configuration is propagated to its associated secondary VLANs.
  - If DHCP snooping is configured on the primary VLAN and you configure DHCP snooping with different settings on an associated secondary VLAN, the configuration on the secondary VLAN does not take effect.
  - If DHCP snooping is not configured on the primary VLAN and you configure DHCP snooping on a secondary VLAN, the configuration takes affect only on the secondary VLAN.
  - When you manually configure DHCP snooping on a secondary VLAN, this message appears:

    ```
    DHCP Snooping configuration may not take effect on secondary vlan XXX
    ```
  - The **show ip dhcp snooping** command displays all VLANs (both primary and secondary) that have DHCP snooping enabled.
- If DHCP snooping information option is configured, it takes higher precedence than the DHCP snooping host tracking feature in determining the port on which to forward reply packets.

Minimum DHCP Snooping Configuration

The minimum configuration steps for the DHCP snooping feature are as follows:

1. Define and configure the DHCP server.

   For DHCP server configuration information, see "Configuring DHCP" in the *Cisco IOS IP and IP Routing Configuration Guide* at:

   http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfdhcp.html
2. Enable DHCP snooping on at least one VLAN.

   By default, DHCP snooping is inactive on all VLANs. See the "Enabling DHCP Snooping on VLANs" section on page 54-13
3. Ensure that DHCP server is connected through a trusted interface.

   By default, the trust state of all interfaces is untrusted. See the "Configuring the DHCP Trust State on Layer 2 LAN Interfaces" section on page 54-14

4. Configure the DHCP snooping database agent.

   This step ensures that database entries are restored after a restart or switchover. See the "Configuring the DHCP Snooping Database Agent" section on page 54-16

5. Enable DHCP snooping globally.

   The feature is not active until you complete this step. See the "Enabling DHCP Snooping Globally" section on page 54-9

If you are configuring the switch for DHCP relay, the following additional steps are required:

1. Define and configure the DHCP relay agent IP address.

   If the DHCP server is in a different subnet from the DHCP clients, configure the server IP address in the helper address field of the client side VLAN.

2. Configure DHCP option-82 on untrusted port.

   See the "Enabling the DHCP Option-82 on Untrusted Port Feature" section on page 54-11

Configuring DHCP Snooping

These sections describe how to configure DHCP snooping:

Enabling DHCP Snooping Globally

Note Configure this command as the last configuration step (or enable the DHCP feature during a scheduled maintenance period) because after you enable DHCP snooping globally, the switch drops DHCP requests until you configure the ports.

To enable DHCP snooping globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip dhcp snooping`	Enables DHCP snooping globally.
Step 2	`Router(config)# do show ip dhcp snooping | include Switch`	Verifies the configuration.

This example shows how to enable DHCP snooping globally:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip dhcp snooping
Router(config)# do show ip dhcp snooping | include Switch
Switch DHCP snooping is enabled
Router(config)#
```

Note When DHCP snooping is disabled and DAI is enabled, the switch shuts down all the hosts because all ARP entries in the ARP table will be checked against a nonexistent DHCP database. When DHCP snooping is disabled or in non-DHCP environments, use ARP ACLs to permit or to deny ARP packets.

Enabling DHCP Option-82 Data Insertion

To enable DHCP option-82 data insertion, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip dhcp snooping information option`	Enables DHCP option-82 data insertion.

Step 2	`Router(config)# ip dhcp snooping information option replace`
	Or:
	`Router(config-if)# ip dhcp snooping information option replace`
		(Optional) Replaces the DHCP relay information option received in snooped packets with the switch's option-82 data. Available in releases where CSCto29645 is resolved and when DHCP option-82 data insertion is enabled.

Step 3	`Router(config)# do show ip dhcp snooping | include 82`	Verifies the configuration.

This example shows how to disable DHCP option-82 data insertion:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# no ip dhcp snooping information option
Router(config)# do show ip dhcp snooping | include 82
Insertion of option 82 is disabled
Router(config)#
```

This example shows how to enable DHCP option-82 data insertion:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip dhcp snooping information option
Router(config)# do show ip dhcp snooping | include 82
Insertion of option 82 is enabled
Router(config)#
```

Enabling the DHCP Option-82 on Untrusted Port Feature

Note With the DHCP option-82 on untrusted port feature enabled, the switch does not drop DHCP packets that include option-82 information that are received on untrusted ports. Do not enter the **ip dhcp snooping information option allowed-untrusted** command on an aggregation switch to which any untrusted devices are connected.

To enable untrusted ports to accept DHCP packets that include option-82 information, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip dhcp snooping information option allow-untrusted`** Or: `Router(config-if)#` **`ip dhcp snooping information option allow-untrusted`**	Enables untrusted ports to accept incoming DHCP packets with option-82 information. Available in interface configuration mode in releases where CSCto29645 is resolved. The default setting is disabled.
Step 2	`Router(config)#` **`do show ip dhcp snooping`**	Verifies the configuration.

This example shows how to enable the DHCP option-82 on untrusted port feature:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip dhcp snooping information option allow-untrusted
Router(config)#
```

Enabling DHCP Snooping Host Tracking

To configure the DHCP snooping host tracking feature, perform one or more of the following tasks:

Command	Purpose
`Router(config)#` **`ip dhcp snooping track host`**	Enables the DHCP snooping host tracking feature.
`Router#` **`show ip dhcp snooping track host`**	Displays the contents of the DHCP snooping host tracking cache.
`Router#` **`show ip dhcp snooping track host statistics`**	Displays the DHCP snooping host track statistics.
`Router#` **`clear ip dhcp snooping track host`**	Clears the DHCP snooping host track cache.
`Router#` **`clear ip dhcp snooping track hosts statistics`**	Clears the DHCP snooping host track statistics.

This example shows how to enable the DHCP snooping host tracking feature:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# no ip dhcp snooping information option
Router(config)# ip dhcp snooping track host
Router(config)# exit
```

This example shows how to display the contents of the DHCP snooping host tracking cache:

```
Router# show ip dhcp snooping track host
VLAN   interface     mac            time left
 -----------------------------------------
 203     Gi3/47      000a.cb00.126d     expired
 204     Gi11/47     000a.cc00.1262     expired
 202     Gi2/47      000a.ca00.125d     expired
 204     Gi11/47     000a.cc00.1263     expired
 203     Gi3/47      000a.cb00.1276     expired
 201     Gi1/47      000a.c900.1273     expired
```

This example shows how to display the statistics associated with DHCP snooping host tracking feature:

```
Router# show ip dhcp snooping track host statistics
DHCP host track entries            = 168
DHCP host track hits               = 34028
DHCP host track misses             = 0
DHCP host track limit exceeded     = 0
```

Enabling DHCP Snooping MAC Address Verification

With DHCP snooping MAC address verification enabled, DHCP snooping verifies that the source MAC address and the client hardware address match in DHCP packets that are received on untrusted ports. The source MAC address is a Layer 2 field associated with the packet, and the client hardware address is a Layer 3 field in the DHCP packet.

To enable DHCP snooping MAC address verification, perform this task:

	Command	Purpose
Step 1	Router(config)# **ip dhcp snooping verify mac-address**	Enables DHCP snooping MAC address verification.
Step 2	Router(config)# **do show ip dhcp snooping | include hwaddr**	Verifies the configuration.

This example shows how to disable DHCP snooping MAC address verification:

```
Router(config)# no ip dhcp snooping verify mac-address
Router(config)# do show ip dhcp snooping | include hwaddr
Verification of hwaddr field is disabled
Router(config)#
```

This example shows how to enable DHCP snooping MAC address verification:

```
Router(config)# ip dhcp snooping verify mac-address
Router(config)# do show ip dhcp snooping | include hwaddr
Verification of hwaddr field is enabled
Router(config)#
```

Enabling DHCP Snooping on VLANs

By default, the DHCP snooping feature is inactive on all VLANs. You may enable the feature on a single VLAN or a range of VLANs.

When enabled on a VLAN, the DHCP snooping feature creates four entries in the VACL table in the MFC3. These entries cause the PFC3 to intercept all DHCP messages on this VLAN and send them to the RP. The DHCP snooping feature is implemented in software on the RP.

To enable DHCP snooping on VLANs, perform this task:

	Command	Purpose
Step 1	Router(config)# **ip dhcp snooping vlan** {{*vlan_ID* [*vlan_ID*]} | {*vlan_range*}	Enables DHCP snooping on a VLAN or VLAN range.
Step 2	Router(config)# **do show ip dhcp snooping**	Verifies the configuration.

You can configure DHCP snooping for a single VLAN or a range of VLANs:

- To configure a single VLAN, enter a single VLAN number.
- To configure a range of VLANs, enter a beginning and an ending VLAN number or a dash-separated pair of VLAN numbers.
- You can enter a comma-separated list of VLAN numbers and dash-separated pairs of VLAN numbers.

This example shows how to enable DHCP snooping on VLANs 10 through 12:

```
Router# configure terminal
Router(config)# ip dhcp snooping vlan 10 12
Router(config)#
```

This example shows another way to enable DHCP snooping on VLANs 10 through 12:

```
Router# configure terminal
Router(config)# ip dhcp snooping vlan 10-12
```

This example shows another way to enable DHCP snooping on VLANs 10 through 12:

```
Router# configure terminal
Router(config)# ip dhcp snooping vlan 10,11,12
```

This example shows how to enable DHCP snooping on VLANs 10 through 12 and VLAN 15:

```
Router# configure terminal
Router(config)# ip dhcp snooping vlan 10-12,15
```

This example shows how to verify the configuration:

```
Router(config)# do show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10-12,15
DHCP snooping is operational on following VLANs:
none
DHCP snooping is configured on the following Interfaces:

Insertion of option 82 is enabled
Verification of hwaddr field is enabled
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
Router#
```

Configuring the DHCP Trust State on Layer 2 LAN Interfaces

To configure DHCP trust state on a Layer 2 LAN interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{`*`type`*[1] *`slot/port`* `|` **`port-channel`** *`number`*`}`	Selects the interface to configure. **Note** Select only LAN ports configured with the **switchport** command or Layer 2 port-channel interfaces.
Step 2	`Router(config-if)#` **`ip dhcp snooping trust`**	Configures the interface as trusted.
Step 3	`Router(config-if)#` **`do show ip dhcp snooping	begin pps`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure Fast Ethernet port 5/12 as trusted:

```
Router# configure terminal
Router(config)# interface FastEthernet 5/12
Router(config-if)# ip dhcp snooping trust
Router(config-if)# do show ip dhcp snooping | begin pps
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
FastEthernet5/12             yes         unlimited
Router#
```

This example shows how to configure Fast Ethernet port 5/12 as untrusted:

```
Router# configure terminal
Router(config)# interface FastEthernet 5/12
Router(config-if)# no ip dhcp snooping trust
Router(config-if)# do show ip dhcp snooping | begin pps
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
FastEthernet5/12             no          unlimited
Router#
```

Configuring Spurious DHCP Server Detection

To detect and locate spurious DHCP servers, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip dhcp snooping detect spurious vlan`** *`range`*	Enables detection of spurious DHCP servers on a specified VLAN range.
Step 2	`Router(config)#` **`ip dhcp snooping detect spurious interval`** *`time`*	Sets the interval time, the default is 30 minutes.
Step 3	`Router#` **`show ip dhcp snooping detect spurious`**	Verifies spurious DHCP server detection.

This example shows how to configure DHCP spurious server detection on VLANs 20 to 25 and set the interval to 50 minutes:

```
Router# configure terminal
Router(config)# ip dhcp snooping detect spurious vlan 20-25
```

```
Router(config)# ip dhcp snooping detect spurious interval 50
Router# do show ip dhcp snooping detect spurious
Spurious DHCP server detection is enabled.

Detection VLAN list : 20-25
Detection interval : 50 minutes
Router#
```

Configuring DHCP Snooping Rate Limiting on Layer 2 LAN Interfaces

To configure DHCP snooping rate limiting on a Layer 2 LAN interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {type[1] slot/port | port-channel number}`	Selects the interface to configure. **Note** Select only LAN ports configured with the **switchport** command or Layer 2 port-channel interfaces.
Step 2	`Router(config-if)# ip dhcp snooping limit rate rate`	Configures DHCP packet rate limiting.
Step 3	`Router(config-if)# do show ip dhcp snooping | begin pps`	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring DHCP snooping rate limiting on a Layer 2 LAN interface, note the following information:

- We recommend an untrusted rate limit of not more than 100 packets per second (pps).
- If you configure rate limiting for trusted interfaces, you might need to increase the rate limit on trunk ports carrying more than one VLAN on which DHCP snooping is enabled.
- DHCP snooping puts ports where the rate limit is exceeded into the error-disabled state.

This example shows how to configure DHCP packet rate limiting to 100 pps on Fast Ethernet port 5/12:

```
Router# configure terminal
Router(config)# interface FastEthernet 5/12
Router(config-if)# ip dhcp snooping limit rate 100
Router(config-if)# do show ip dhcp snooping | begin pps
Interface                    Trusted     Rate limit (pps)
------------------------     -------     ----------------
FastEthernet5/12             no          100
Router#
```

Configuring the DHCP Snooping Database Agent

To configure the DHCP snooping database agent, perform one or more of the following tasks:

Command	Purpose
Router(config)# **ip dhcp snooping database** { *_url* | **write-delay** *seconds* | **timeout** *seconds* }	Configures a URL for the database agent (or file) and the related timeout values.
Router# **show ip dhcp snooping database** [**detail**]	Displays the current operating state of the database agent and statistics associated with the transfers.
Router# **clear ip dhcp snooping database statistics**	Clears the statistics associated with the database agent.
Router# **renew ip dhcp snooping database** [**validation none**] [*url*]	Requests the read entries from a file at the given URL.
Router# **ip dhcp snooping binding** *mac_address* **vlan** *vlan_ID ip_address* **interface** *ifname* **expiry** *lease_in_seconds*	Adds bindings to the snooping database.

When configuring the DHCP snooping database agent, note the following information:

- The DHCP snooping database stores at least 8,000 bindings.
- Store the file on a TFTP server to avoid consuming storage space on the switch storage devices.
- When a switchover occurs, if the file is stored in a remote location accessible through TFTP, the newly active supervisor engine can use the binding list.
- Network-based URLs (such as TFTP and FTP) require that you create an empty file at the configured URL before the switch can write the set of bindings for the first time.

Configuration Examples for the Database Agent

These sections provide examples for the database agent:

- Example 1: Enabling the Database Agent, page 54-16
- Example 2: Reading Binding Entries from a TFTP File, page 54-18
- Example 3: Adding Information to the DHCP Snooping Database, page 54-19

Example 1: Enabling the Database Agent

The following example shows how to configure the DHCP snooping database agent to store the bindings at a given location and to view the configuration and operating state:

```
Router# configure terminal
Router(config)# ip dhcp snooping database tftp://10.1.1.1/directory/file
Router(config)# end
Router# show ip dhcp snooping database detail
Agent URL : tftp://10.1.1.1/directory/file
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : 7 (00:00:07)
Abort Timer Expiry : Not Running
```

```
Last Succeeded Time : None
Last Failed Time : 17:14:25 UTC Sat Jul 7 2001
Last Failed Reason : Unable to access URL.

Total Attempts       :       21   Startup Failures :        0
Successful Transfers :        0   Failed Transfers :       21
Successful Reads     :        0   Failed Reads     :        0
Successful Writes    :        0   Failed Writes    :       21
Media Failures       :        0

First successful access: Read

Last ignored bindings counters :
Binding Collisions    :        0   Expired leases    :        0
Invalid interfaces    :        0   Unsupported vlans :        0
Parse failures        :        0
Last Ignored Time : None

Total ignored bindings counters:
Binding Collisions    :        0   Expired leases    :        0
Invalid interfaces    :        0   Unsupported vlans :        0
Parse failures        :        0

Router#
```

The first three lines of output show the configured URL and related timer-configuration values. The next three lines show the operating state and the amount of time left for expiry of write delay and abort timers.

Among the statistics shown in the output, startup failures indicate the number of attempts to read or create the file that failed on bootup.

Note Create a temporary file on the TFTP server with the **touch** command in the TFTP server daemon directory. With some UNIX implementations, the file should have full read and write access permissions (777).

DHCP snooping bindings are keyed on the MAC address and VLAN combination. If an entry in the remote file has an entry for a given MAC address and VLAN set for which the switch already has a binding, the entry from the remote file is ignored when the file is read. This condition is referred to as the *binding collision*.

An entry in a file may no longer be valid because the lease indicated by the entry may have expired by the time it is read. The expired leases counter indicates the number of bindings that are ignored because of this condition. The Invalid interfaces counter refers to the number of bindings that have been ignored when the interface referred by the entry either does not exist on the system or is a router or DHCP snooping trusted interface (if it exists) when the read happened. Unsupported VLANs refers to the number of entries that have been ignored because the indicated VLAN is not supported on the system. The Parse failures counter provides the number of entries that have been ignored when the switch is unable to interpret the meaning of the entries from the file.

The switch maintains two sets of counters for these ignored bindings. One provides the counters for a read that has at least one binding ignored by at least one of these conditions. These counters are shown as the “Last ignored bindings counters.” The total ignored bindings counters provides a sum of the number of bindings that have been ignored because of all the reads since the switch bootup. These two sets of counters are cleared by the **clear** command. The total counter set may indicate the number of bindings that have been ignored since the last clear.

Example 2: Reading Binding Entries from a TFTP File

To manually read the entries from a TFTP file, perform this task:

	Command	Purpose
Step 1	Router# **show ip dhcp snooping database**	Displays the DHCP snooping database agent statistics.
Step 2	Router# **renew ip dhcp snoop data** *url*	Directs the switch to read the file from the URL.
Step 3	Router# **show ip dhcp snoop data**	Displays the read status.
Step 4	Router# **show ip dhcp snoop bind**	Verifies whether the bindings were read successfully.

This is an example of how to manually read entries from the tftp://10.1.1.1/directory/file:

```
Router# show ip dhcp snooping database
Agent URL :
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : Not Running
Abort Timer Expiry : Not Running

Last Succeeded Time : None
Last Failed Time : None
Last Failed Reason : No failure recorded.

Total Attempts       :        0   Startup Failures :        0
Successful Transfers :        0   Failed Transfers :        0
Successful Reads     :        0   Failed Reads     :        0
Successful Writes    :        0   Failed Writes    :        0
Media Failures       :        0
Router# renew ip dhcp snoop data tftp://10.1.1.1/directory/file
Loading directory/file from 10.1.1.1 (via GigabitEthernet1/1): !
[OK - 457 bytes]
Database downloaded successfully.

Router#
00:01:29: %DHCP_SNOOPING-6-AGENT_OPERATION_SUCCEEDED: DHCP snooping database Read
succeeded.
Router# show ip dhcp snoop data
Agent URL :
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : Not Running
Abort Timer Expiry : Not Running

Last Succeeded Time : 15:24:34 UTC Sun Jul 8 2001
Last Failed Time : None
Last Failed Reason : No failure recorded.

Total Attempts       :        1   Startup Failures :        0
Successful Transfers :        1   Failed Transfers :        0
Successful Reads     :        1   Failed Reads     :        0
Successful Writes    :        0   Failed Writes    :        0
Media Failures       :        0
Router#
Router# show ip dhcp snoop bind
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
```

```
------------------  ---------------  ----------  -------------  ----  --------------------
00:01:00:01:00:05   1.1.1.1          49810       dhcp-snooping  512   GigabitEthernet1/1
00:01:00:01:00:02   1.1.1.1          49810       dhcp-snooping  512   GigabitEthernet1/1
00:01:00:01:00:04   1.1.1.1          49810       dhcp-snooping  1536  GigabitEthernet1/1
00:01:00:01:00:03   1.1.1.1          49810       dhcp-snooping  1024  GigabitEthernet1/1
00:01:00:01:00:01   1.1.1.1          49810       dhcp-snooping  1     GigabitEthernet1/1
Router# clear ip dhcp snoop bind
Router# show ip dhcp snoop bind
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
Router#
```

Example 3: Adding Information to the DHCP Snooping Database

To manually add a binding to the DHCP snooping database, perform this task:

	Command	Purpose
Step 1	Router# **show ip dhcp snooping binding**	Views the DHCP snooping database.
Step 2	Router# **ip dhcp snooping binding** *binding_id* **vlan** *vlan_id* **interface** *interface* **expiry** *lease_time*	Adds the binding using the **ip dhcp snooping** exec command.
Step 3	Router# **show ip dhcp snooping binding**	Checks the DHCP snooping database.

This example shows how to manually add a binding to the DHCP snooping database:

```
Router# show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
Router#
Router# ip dhcp snooping binding 1.1.1 vlan 1 1.1.1.1 interface gi1/1 expiry 1000

Router# show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:01:00:01:00:01   1.1.1.1          992         dhcp-snooping  1     GigabitEthernet1/1
Router#
```

Displaying a Binding Table

The DHCP snooping binding table for each switch contains binding entries that correspond to untrusted ports. The table does not contain information about hosts interconnected with a trusted port because each interconnected switch will have its own DHCP snooping binding table.

This example shows how to display the DHCP snooping binding information for a switch:

```
Router# show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:02:B3:3F:3B:99   55.5.5.2         6943        dhcp-snooping  10    FastEthernet6/10
```

Table 54-2 describes the fields in the **show ip dhcp snooping binding** command output.

Table 54-2 show ip dhcp snooping binding Command Output

Field	Description
MAC Address	Client hardware MAC address
IP Address	Client IP address assigned from the DHCP server
Lease (seconds)	IP address lease time
Type	Binding type: dynamic binding learned by DHCP snooping or statically-configured binding
VLAN	VLAN number of the client interface
Interface	Interface that connects to the DHCP client host

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring IP Source Guard

This chapter describes how to configure IP Source Guard. Cisco IOS Release 12.2(33)SXH and later releases support IP Source Guard.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Overview of IP Source Guard

IP Source Guard provides source IP address filtering on a Layer 2 port to prevent a malicious host from impersonating a legitimate host by assuming the legitimate host's IP address. The feature uses dynamic DHCP snooping and static IP source binding to match IP addresses to hosts on untrusted Layer 2 access ports.

Initially, all IP traffic on the protected port is blocked except for DHCP packets. After a client receives an IP address from the DHCP server, or after static IP source binding is configured by the administrator, all traffic with that IP source address is permitted from that client. Traffic from other hosts is denied.

This filtering limits a host's ability to attack the network by claiming a neighbor host's IP address.
IP Source Guard is a port-based feature that automatically creates an implicit port access control list (PACL).

IP Source Guard Interaction with VLAN-Based Features

Use the **access-group mode** command to specify how IP Source Guard interacts with VLAN-based features (such as VACL and Cisco IOS ACL and RACL).

In prefer port mode, if IP Source Guard is configured on an interface, IP Source Guard overrides other VLAN-based features. If IP Source Guard is not configured on the interface, other VLAN-based features are merged in the ingress direction and applied on the interface.

In merge mode, IP Source Guard and VLAN-based features are merged in the ingress direction and applied on the interface. This is the default access-group mode.

Channel Ports

IP Source Guard is supported on the main Layer 2 channel interface but not on the port members. When IP Source Guard is applied on the main Layer 2 channel interface, it is applied to all the member ports in the channel.

Trunk Ports

IP Source Guard is not supported on trunk ports.

Layer 2 and Layer 3 Port Conversion

When an IP Source Guard policy is applied to a Layer 2 port, and then you change that port to be a Layer 3 port, the IP Source Guard policy no longer functions but is still present in the configuration. When the port is changed back to a Layer 2 port, IP Source Guard policy becomes effective again.

IP Source Guard and Voice VLAN

IP Source Guard is supported on a Layer 2 port that belongs to a voice VLAN. To configure the voice VLAN for the Layer 2 port, use the **switchport voice vlan** command. For IP Source Guard to be active on the voice VLAN, DHCP snooping must be enabled on the voice VLAN. In merge mode, the IP Source Guard feature is merged with VACL and Cisco IOS ACL configured on the access VLAN.

IP Source Guard and Web-Based Authentication

In releases earlier than Cisco IOS Release 12.2(33)SXI2, configuring IP Source Guard and web-based authentication on the same interface is not supported.

In Cisco IOS Release 12.2(33)SXI2 and later releases, you can configure IP Source Guard and web-based authentication on the same interface. If DHCP snooping is also enabled on the access VLAN, you must enter the **mls acl tcam override dynamic dhcp-snooping** command in global configuration mode to avoid conflicts between the two features. Other VLAN-based features are not supported when IP Source Guard and web-based authentication are combined.

IP Source Guard Restrictions

Because the IP Source Guard feature is supported only in hardware, IP Source Guard is not applied if there are insufficient hardware resources available. These hardware resources are shared by various other ACL features that are configured on the system. The following restrictions apply to IP Source Guard:

- Only supported on ingress Layer 2 ports.
- Only supported in hardware.
- Not applied to any traffic that is processed in software.
- Does not support filtering of traffic based on MAC address.
- Is not supported on private VLANs.

Configuring IP Source Guard on the Switch

To enable IP Source Guard, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip dhcp snooping`**	Enables DHCP snooping globally. You can use the **no** keyword to disable DHCP snooping.
Step 2	`Router(config)#` **`ip dhcp snooping vlan`** *`number`* `[`*`number`*`]`	Enables DHCP snooping on your VLANs.
Step 3	`Router(config)#` **`interface`** *`interface-name`*	Selects the interface to be configured.
Step 4	`Router(config-if)#` **`no ip dhcp snooping trust`**	Use the **no** keyword to configure the interface as untrusted.
Step 5	`Router(config-if)#` **`ip verify source vlan dhcp-snooping`** `[`**`port-security`**`]`	Enables IP Source Guard, source IP address filtering on the port. The following are the command parameters: • **vlan** applies the feature to only specific VLANs on the interface. The **dhcp-snooping** option applies the feature to all VLANs on the interface that have DHCP snooping enabled. • **port-security** enables MAC address filtering. This feature is currently not supported.
Step 6	`Router(config-if)#` **`exit`**	Returns to global configuration mode.
Step 7	`Router(config)#` **`ip source binding`** *`mac-address`* **`vlan`** *`vlan-id ip-address`* **`interface`** *`interface-name`*	(Optional) Configures a static IP binding on the port.
Step 8	`Router(config)#` **`end`**	Exits configuration mode.
Step 9	`Router#` **`show ip verify source`** `[`**`interface`** *`interface-name`*`]`	Verifies the configuration.

Note The static IP source binding can only be configured on a Layer 2 port. If you enter the **ip source binding vlan interface** command on a Layer 3 port, you receive this error message:

```
Static IP source binding can only be configured on switch port.
```

The **no** keyword deletes the corresponding IP source binding entry. This command requires an exact match of all the required parameters in order for the deletion to be successful.

This example shows how to enable per-Layer 2 port IP Source Guard on VLANs 10 through 20:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip dhcp snooping
Router(config)# ip dhcp snooping vlan 10 20
Router(config)# interface fa6/1
Router(config-if)# switchport mode access
Router(config-if)# switchport access vlan 10
Router(config-if)# no ip dhcp snooping trust
Router(config-if)# ip verify source vlan dhcp-snooping
Router(config-if)# end
Router# show ip verify source interface f6/1
Interface  Filter-type  Filter-mode  IP-address       Mac-address        Vlan
---------  -----------  -----------  ---------------  -----------------  ----------
Fa6/1      ip           active       10.0.0.1                            10
Fa6/1      ip           active       deny-all                            11-20
Router#
```

The output shows that there is one valid DHCP binding to VLAN 10.

This example shows how to configure an interface to use prefer port mode:

```
Router# configure terminal
Router(config)# interface gigabitEthernet 6/1
Router(config-if)# access-group mode prefer port
```

This example shows how to configure an interface to use merge mode:

```
Router# configure terminal
Router(config)# interface gigabitEthernet 6/1
Router(config-if)# access-group mode merge
```

Displaying IP Source Guard Information

To display IP Source Guard PACL information for all interfaces on a switch, perform this task:

Command	Purpose
`Router# show ip verify source [interface interface-name]`	Displays IP Source Guard PACL information for all interfaces on a switch or for a specified interface.

This example shows that DHCP snooping is enabled on VLAN 10 through 20, interface fa6/1 is configured for IP filtering, and there is an existing IP address binding 10.0.01 on VLAN 10:

```
Router# show ip verify source interface fa6/1
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/1      ip           active       10.0.0.1                         10
```

```
fa6/1      ip          active       deny-all                        11-20
```

Note The second entry shows that a default PACL (deny all IP traffic) is installed on the port for those snooping-enabled VLANs that do not have a valid IP source binding.

This example shows the displayed PACL information for a trusted port:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/2      ip           inactive-trust-port
```

This example shows the displayed PACL information for a port in a VLAN not configured for DHCP snooping:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/3      ip           inactive-no-snooping-vlan
```

This example shows the displayed PACL information for a port with multiple bindings configured for an IP/MAC filtering:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/4      ip           active       10.0.0.2         aaaa.bbbb.cccc  10
fa6/4      ip           active       11.0.0.1         aaaa.bbbb.cccd  11
fa6/4      ip           active       deny-all         deny-all        12-20
```

This example shows the displayed PACL information for a port configured for IP/MAC filtering but not for port security:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/5      ip           active       10.0.0.3         permit-all      10
fa6/5      ip           active       deny-all         permit-all      11-20
```

Note The MAC address filter shows permit-all because port security is not enabled, so the MAC filter cannot apply to the port/VLAN and is effectively disabled. Always enable port security first.

This example shows an error message when you enter the **show ip verify source** command on a port that does not have an IP source filter mode configured:

```
Router# show ip verify source interface fa6/6
IP Source Guard is not configured on the interface fa6/6.
```

This example shows how to display all interfaces on the switch that have IP Source Guard enabled:

```
Router# show ip verify source
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/1      ip           active       10.0.0.1                         10
fa6/1      ip           active       deny-all                         11-20
fa6/2      ip           inactive-trust-port
fa6/3      ip           inactive-no-snooping-vlan
fa6/4      ip           active       10.0.0.2         aaaa.bbbb.cccc  10
fa6/4      ip           active       11.0.0.1         aaaa.bbbb.cccd  11
fa6/4      ip           active       deny-all         deny-all        12-20
fa6/5      ip           active       10.0.0.3         permit-all      10
fa6/5      ip           active       deny-all         permit-all      11-20
```

Displaying IP Source Binding Information

To display all IP source bindings configured on all interfaces on a switch, perform this task:

Command	Purpose
`Router# show ip source binding [ip-address] [mac-address] [dhcp-snooping	static] [vlan vlan-id] [interface interface-name]`	Displays IP source bindings using the optional specified display filters. The **dhcp-snooping** filter displays all VLANs on the interface that have DHCP snooping enabled.

This example shows how to display all IP source bindings configured on all interfaces on the switch.

```
Router# show ip source binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:02:B3:3F:3B:99   55.5.5.2         6522        dhcp-snooping  10    FastEthernet6/10
00:00:00:0A:00:0B   11.0.0.1         infinite    static         10    FastEthernet6/10
Router#
```

Table 55-1 describes the fields in the **show ip source binding** command output.

Table 55-1 show ip source binding Command Output

Field	Description
MAC Address	Client hardware MAC address
IP Address	Client IP address assigned from the DHCP server
Lease (seconds)	IP address lease time
Type	Binding type; static bindings configured from CLI to dynamic binding learned from DHCP snooping
VLAN	VLAN number of the client interface
Interface	Interface that connects to the DHCP client host

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding DAI

These sections describe how DAI helps prevent ARP spoofing attacks:

Understanding ARP

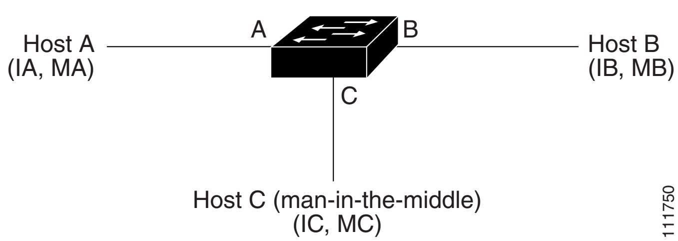
ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, Host B wants to send information to Host A but does not have the MAC address of Host A in its ARP cache. Host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of Host A. All hosts within the broadcast domain receive the ARP request, and Host A responds with its MAC address.

Understanding ARP Spoofing Attacks

ARP spoofing attacks and ARP cache poisoning can occur because ARP allows a gratuitous reply from a host even if an ARP request was not received. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

An ARP spoofing attack can target hosts, switches, and routers connected to your Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. Figure 56-1 shows an example of ARP cache poisoning.

Figure 56-1 ARP Cache Poisoning



Hosts A, B, and C are connected to the switch on interfaces A, B and C, all of which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, Host A uses IP address IA and MAC address MA. When Host A needs to communicate to Host B at the IP layer, it broadcasts an ARP request for the MAC address associated with IP address IB. When the switch and Host B receive the ARP request, they populate their ARP caches with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When Host B responds, the switch and Host A populate their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can poison the ARP caches of the switch for Host A, and Host B by broadcasting forged ARP responses with bindings for a host with an IP address of IA (or IB) and a MAC address of MC. Hosts with poisoned ARP caches use the MAC address MC as the destination MAC address for traffic intended for IA or IB. This means that Host C intercepts that traffic. Because Host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. Host C has inserted itself into the traffic stream from Host A to Host B, which is the topology of the classic *man-in-the middle* attack.

Understanding DAI and ARP Spoofing Attacks

DAI is a security feature that validates ARP packets in a network. DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. This capability protects the network from some man-in-the-middle attacks.

DAI ensures that only valid ARP requests and responses are relayed. The switch performs these activities:

- Intercepts all ARP requests and responses on untrusted ports
- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

DAI determines the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a trusted database, the DHCP snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the switch. If the ARP packet is received on a trusted interface, the switch forwards the packet without any checks. On untrusted interfaces, the switch forwards the packet only if it is valid.

DAI can validate ARP packets against user-configured ARP access control lists (ACLs) for hosts with statically configured IP addresses (see "Applying ARP ACLs for DAI Filtering" section on page 56-8). The switch logs dropped packets (see the "Logging of Dropped Packets" section on page 56-5).

You can configure DAI to drop ARP packets when the IP addresses in the packets are invalid or when the MAC addresses in the body of the ARP packets do not match the addresses specified in the Ethernet header (see the "Enabling Additional Validation" section on page 56-11).

Interface Trust States and Network Security

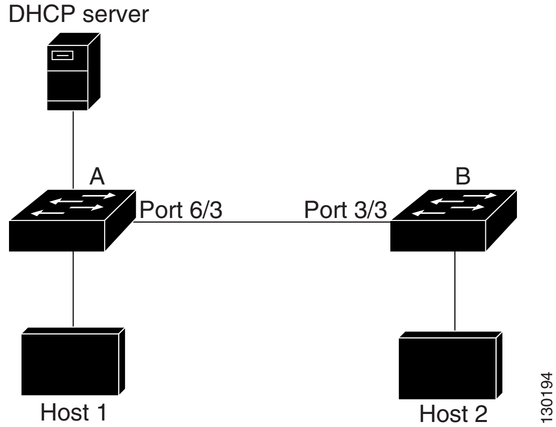
DAI associates a trust state with each interface on the switch. Packets arriving on trusted interfaces bypass all DAI validation checks, and those arriving on untrusted interfaces undergo the DAI validation process.

In a typical network configuration, you configure all switch ports connected to host ports as untrusted and configure all switch ports connected to switches as trusted. With this configuration, all ARP packets entering the network from a given switch bypass the security check. No other validation is needed at any other place in the VLAN or in the network. You configure the trust setting by using the **ip arp inspection trust** interface configuration command.

Caution Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

In Figure 56-2, assume that both Switch A and Switch B are running DAI on the VLAN that includes Host 1 and Host 2. If Host 1 and Host 2 acquire their IP addresses from the DHCP server connected to Switch A, only Switch A binds the IP-to-MAC address of Host 1. Therefore, if the interface between Switch A and Switch B is untrusted, the ARP packets from Host 1 are dropped by Switch B. Connectivity between Host 1 and Host 2 is lost.

Figure 56-2 ARP Packet Validation on a VLAN Enabled for DAI



Configuring interfaces to be trusted when they are actually untrusted leaves a security hole in the network. If Switch A is not running DAI, Host 1 can easily poison the ARP cache of Switch B (and Host 2, if the link between the switches is configured as trusted). This condition can occur even though Switch B is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a switch running DAI do not poison the ARP caches of other hosts in the network. However, DAI does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a switch running DAI.

In cases in which some switches in a VLAN run DAI and other switches do not, configure the interfaces connecting such switches as untrusted. However, to validate the bindings of packets from switches where DAI is not configured, configure ARP ACLs on the switch running DAI. When you cannot determine such bindings, isolate switches running DAI at Layer 3 from switches not running DAI. For configuration information, see the "Sample Two: One Switch Supports DAI" section on page 56-20.

Note Depending on the setup of the DHCP server and the network, it might not be possible to validate a given ARP packet on all switches in the VLAN.

Rate Limiting of ARP Packets

The switch performs DAI validation checks, which rate limits incoming ARP packets to prevent a denial-of-service attack. By default, the rate for untrusted interfaces is 15 packets per second (pps). Trusted interfaces are not rate limited. You can change this setting by using the **ip arp inspection limit** interface configuration command.

When the rate of incoming ARP packets exceeds the configured limit, the switch places the port in the error-disabled state. The port remains in that state until you intervene. You can use the **errdisable recovery** global configuration command to enable error disable recovery so that ports automatically emerge from this state after a specified timeout period.

For configuration information, see the "Configuring ARP Packet Rate Limiting" section on page 56-9.

Relative Priority of ARP ACLs and DHCP Snooping Entries

DAI uses the DHCP snooping binding database for the list of valid IP-to-MAC address bindings.

ARP ACLs take precedence over entries in the DHCP snooping binding database. The switch uses ACLs only if you configure them by using the **ip arp inspection filter** global configuration command. The switch first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the switch also denies the packet even if a valid binding exists in the database populated by DHCP snooping.

Logging of Dropped Packets

When the switch drops a packet, it places an entry in the log buffer and then generates system messages on a rate-controlled basis. After the message is generated, the switch clears the entry from the log buffer. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You use the **ip arp inspection log-buffer** global configuration command to configure the number of entries in the buffer and the number of entries needed in the specified interval to generate system messages. You specify the type of packets that are logged by using the **ip arp inspection vlan logging** global configuration command. For configuration information, see the "Configuring DAI Logging" section on page 56-12.

Default DAI Configuration

Table 56-1 shows the default DAI configuration.

Table 56-1 Default DAI Configuration

Feature	Default Setting
DAI	Disabled on all VLANs.
Interface trust state	All interfaces are untrusted.
Rate limit of incoming ARP packets	The rate is 15 pps on untrusted interfaces, assuming that the network is a Layer 2-switched network with a host connecting to as many as 15 new hosts per second. The rate is unlimited on all trusted interfaces. The burst interval is 1 second.
ARP ACLs for non-DHCP environments	No ARP ACLs are defined.
Validation checks	No checks are performed.
Log buffer	When DAI is enabled, all denied or dropped ARP packets are logged. The number of entries in the log is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second.
Per-VLAN logging	All denied or dropped ARP packets are logged.

DAI Configuration Guidelines and Restrictions

When configuring DAI, follow these guidelines and restrictions:

- DAI is an ingress security feature; it does not perform any egress checking.
- DAI is not effective for hosts connected to switches that do not support DAI or that do not have this feature enabled. Because man-in-the-middle attacks are limited to a single Layer 2 broadcast domain, separate the domain with DAI checks from the one with no checking. This action secures the ARP caches of hosts in the domain enabled for DAI.
- DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses. For configuration information, see Chapter 54, "Configuring DHCP Snooping."
- When DHCP snooping is disabled or in non-DHCP environments, use ARP ACLs to permit or to deny packets.
- DAI is supported on access ports, trunk ports, EtherChannel ports, and private VLAN ports.
- A physical port can join an EtherChannel port channel only when the trust state of the physical port and the channel port match.

  Conversely, when you change the trust state on the port channel, the switch configures a new trust state on all the physical ports that comprise the channel.
- The operating rate for the port channel is cumulative across all the physical ports within the channel. For example, if you configure the port channel with an ARP rate limit of 400 pps, all the interfaces combined on the channel receive an aggregate 400 pps. The rate of incoming ARP packets on EtherChannel ports is equal to the sum of the incoming rate of packets from all the channel members. Configure the rate limit for EtherChannel ports only after examining the rate of incoming ARP packets on the channel-port members.

  The rate of incoming packets on a physical port is checked against the port-channel configuration rather than the physical-ports configuration. The rate-limit configuration on a port channel is independent of the configuration on its physical ports.

  If the EtherChannel receives more ARP packets than the configured rate, the channel (including all physical ports) is placed in the error-disabled state.
- Make sure to limit the rate of ARP packets on incoming trunk ports. Configure trunk ports with higher rates to reflect their aggregation and to handle packets across multiple DAI-enabled VLANs. You also can use the **ip arp inspection limit none** interface configuration command to make the rate unlimited. A high rate-limit on one VLAN can cause a denial-of-service attack to other VLANs when the software places the port in the error-disabled state.

Configuring DAI

These sections describe how to configure DAI:

- Enabling Additional Validation, page 56-11
- Configuring DAI Logging, page 56-12
- Displaying DAI Information, page 56-14

Enabling DAI on VLANs

To enable DAI on VLANs, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`ip arp inspection vlan`** `{`*`vlan_ID`* `|` *`vlan_range`*`}`	Enables DAI on VLANs.
Step 3	`Router(config-if)#` **`do show ip arp inspection vlan`** `{`*`vlan_ID`* `|` *`vlan_range`*`}` **`| begin Vlan`**	Verifies the configuration.

You can enable DAI on a single VLAN or a range of VLANs:

- To enable a single VLAN, enter a single VLAN number.
- To enable a range of VLANs, enter a dash-separated pair of VLAN numbers.
- You can enter a comma-separated list of VLAN numbers and dash-separated pairs of VLAN numbers.

This example shows how to enable DAI on VLANs 10 through 12:

```
Router# configure terminal
Router(config)# ip arp inspection vlan 10-12
```

This example shows another way to enable DAI on VLANs 10 through 12:

```
Router# configure terminal
Router(config)# ip arp inspection vlan 10,11,12
```

This example shows how to enable DAI on VLANs 10 through 12 and VLAN 15:

```
Router# configure terminal
Router(config)# ip arp inspection vlan 10-12,15
```

This example shows how to verify the configuration:

```
Router(config)# do show ip arp inspection vlan 10-12,15 | begin Vlan
Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
   10     Enabled          Inactive
   11     Enabled          Inactive
   12     Enabled          Inactive
   15     Enabled          Inactive

 Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
   10     Deny             Deny
   11     Deny             Deny
   12     Deny             Deny
   15     Deny             Deny
```

Configuring the DAI Interface Trust State

The switch forwards ARP packets that it receives on a trusted interface, but does not check them.

On untrusted interfaces, the switch intercepts all ARP requests and responses. It verifies that the intercepted packets have valid IP-to-MAC address bindings before updating the local cache and before forwarding the packet to the appropriate destination. The switch drops invalid packets and logs them in the log buffer according to the logging configuration specified with the **ip arp inspection vlan logging** global configuration command. For more information, see the "Configuring DAI Logging" section on page 56-12.

To configure the DAI interface trust state, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface** {*type*[1] *slot/port* | **port-channel** *number*}	Specifies the interface connected to another switch, and enter interface configuration mode.
Step 3	Router(config-if)# **ip arp inspection trust**	Configures the connection between switches as trusted.
Step 4	Router(config-if)# **do show ip arp inspection interfaces**	Verifies the DAI configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure Fast Ethernet port 5/12 as trusted:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/12
Router(config-if)# ip arp inspection trust
Router(config-if)# do show ip arp inspection interfaces | include Int|--|5/12
Interface        Trust State     Rate (pps)    Burst Interval
 ---------------  -----------     ----------    --------------
 Fa5/12           Trusted               None               N/A
```

Applying ARP ACLs for DAI Filtering

Note See the command reference for information about the **arp access-list** command.

To apply an ARP ACL, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router# **ip arp inspection filter** *arp_acl_name* **vlan** {*vlan_ID* | *vlan_range*} [**static**]	Applies the ARP ACL to a VLAN.
Step 3	Router(config)# **do show ip arp inspection vlan** {*vlan_ID* | *vlan_range*}	Verifies your entries.

When applying ARP ACLs, note the following information:

- For *vlan_range*, you can specify a single VLAN or a range of VLANs:
  - To specify a single VLAN, enter a single VLAN number.
  - To specify a range of VLANs, enter a dash-separated pair of VLAN numbers.
  - You can enter a comma-separated list of VLAN numbers and dash-separated pairs of VLAN numbers.
- (Optional) Specify **static** to treat implicit denies in the ARP ACL as explicit denies and to drop packets that do not match any previous clauses in the ACL. DHCP bindings are not used.

  If you do not specify this keyword, it means that there is no explicit deny in the ACL that denies the packet, and DHCP bindings determine whether a packet is permitted or denied if the packet does not match any clauses in the ACL.
- ARP packets containing only IP-to-MAC address bindings are compared against the ACL. Packets are permitted only if the access list permits them.

This example shows how to apply an ARP ACL named example_arp_acl to VLANs 10 through 12 and VLAN 15:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection filter example_arp_acl vlan 10-12,15
Router(config)# do show ip arp inspection vlan 10-12,15 | begin Vlan
 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
   10     Enabled          Inactive    example_arp_acl   No
   11     Enabled          Inactive    example_arp_acl   No
   12     Enabled          Inactive    example_arp_acl   No
   15     Enabled          Inactive    example_arp_acl   No
Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
   10     Deny             Deny
   11     Deny             Deny
   12     Deny             Deny
   15     Deny             Deny
```

Configuring ARP Packet Rate Limiting

When DAI is enabled, the switch performs ARP packet validation checks, which makes the switch vulnerable to an ARP-packet denial-of-service attack. ARP packet rate limiting can prevent an ARP-packet denial-of-service attack.

To configure ARP packet rate limiting on a port, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface** {*type*[1] *slot/port* | **port-channel** *number*}	Selects the interface to be configured.
Step 3	Router(config-if)# **ip arp inspection limit** {**rate** *pps* [**burst interval** *seconds*] | **none**}	(Optional) Configures ARP packet rate limiting.
Step 4	Router(config-if)# **do show ip arp inspection interfaces**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring ARP packet rate limiting, note the following information:

- The default rate is 15 pps on untrusted interfaces and unlimited on trusted interfaces.
- For **rate** *pps*, specify an upper limit for the number of incoming packets processed per second. The range is 0 to 2048 pps.
- The **rate none** keywords specify that there is no upper limit for the rate of incoming ARP packets that can be processed.
- (Optional) For **burst interval** *seconds* (default is 1), specify the consecutive interval, in seconds, over which the interface is monitored for a high rate of ARP packets.The range is 1 to 15.
- When the rate of incoming ARP packets exceeds the configured limit, the switch places the port in the error-disabled state. The port remains in the error-disabled state until you enable error-disabled recovery, which allows the port to emerge from the error-disabled state after a specified timeout period.
- Unless you configure a rate-limiting value on an interface, changing the trust state of the interface also changes its rate-limiting value to the default value for the configured trust state. After you configure the rate-limiting value, the interface retains the rate-limiting value even when you change its trust state. If you enter the **no ip arp inspection limit** interface configuration command, the interface reverts to its default rate-limiting value.
- For configuration guidelines about limiting the rate of incoming ARP packets on trunk ports and EtherChannel ports, see the "DAI Configuration Guidelines and Restrictions" section on page 56-6.

This example shows how to configure ARP packet rate limiting on Fast Ethernet port 5/14:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/14
Router(config-if)# ip arp inspection limit rate 20 burst interval 2
Router(config-if)# do show ip arp inspection interfaces | include Int|--|5/14
 Interface        Trust State     Rate (pps)    Burst Interval
 ---------------  -----------     ----------    --------------
 Fa5/14           Untrusted               20                 2
```

Enabling DAI Error-Disabled Recovery

To enable DAI error-disabled recovery, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# errdisable recovery cause arp-inspection`	(Optional) Enables DAI error-disabled recovery.
Step 3	`Router(config)# do show errdisable recovery | include Reason|---|arp-`	Verifies the configuration.

This example shows how to enable DAI error disabled recovery:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# errdisable recovery cause arp-inspection
Router(config)# do show errdisable recovery | include Reason|---|arp-
ErrDisable Reason    Timer Status
-----------------    --------------
arp-inspection       Enabled
```

Enabling Additional Validation

DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can enable additional validation on the destination MAC address, the sender and target IP addresses, and the source MAC address.

To enable additional validation, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# ip arp inspection validate {[dst-mac] [ip] [src-mac]}`	(Optional) Enables additional validation.
Step 3	`Router(config)# do show ip arp inspection | include abled$`	Verifies the configuration.

The additional validations do the following:

- **dst-mac**—Checks the destination MAC address in the Ethernet header against the target MAC address in ARP body. This check is performed for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.
- **ip**—Checks the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses.
- **src-mac**—Checks the source MAC address in the Ethernet header against the sender MAC address in the ARP body. This check is performed on both ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

When enabling additional validation, note the following information:

- You must specify at least one of the keywords.
- Each **ip arp inspection validate** command overrides the configuration from any previous commands. If an **ip arp inspection validate** command enables **src-mac** and **dst-mac** validations, and a second **ip arp inspection validate** command enables IP validation only, the **src-mac** and **dst-mac** validations are disabled as a result of the second command.

This example shows how to enable **src-mac** additional validation:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection validate src-mac
Router(config)# do show ip arp inspection | include abled$
Source Mac Validation      : Enabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
```

This example shows how to enable **dst-mac** additional validation:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection validate dst-mac
Router(config)# do show ip arp inspection | include abled$
Source Mac Validation      : Disabled
Destination Mac Validation : Enabled
IP Address Validation      : Disabled
```

This example shows how to enable **ip** additional validation:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection validate ip
Router(config)# do show ip arp inspection | include abled$
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Enabled
```

This example shows how to enable **src-mac** and **dst-mac** additional validation:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection validate src-mac dst-mac
Router(config)# do show ip arp inspection | include abled$
Source Mac Validation      : Enabled
Destination Mac Validation : Enabled
IP Address Validation      : Disabled
```

This example shows how to enable **src-mac**, **dst-mac**, and **ip** additional validation:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection validate src-mac dst-mac ip
Router(config)# do show ip arp inspection | include abled$
Source Mac Validation      : Enabled
Destination Mac Validation : Enabled
IP Address Validation      : Enabled
```

Configuring DAI Logging

These sections describe DAI logging:

DAI Logging Overview

When DAI drops a packet, it places an entry in the log buffer and then generates system messages on a rate-controlled basis. After the message is generated, DAI clears the entry from the log buffer. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

A log-buffer entry can represent more than one packet. For example, if an interface receives many packets on the same VLAN with the same ARP parameters, DAI combines the packets as one entry in the log buffer and generates a single system message for the entry.

If the log buffer overflows, it means that a log event does not fit into the log buffer, and the display for the **show ip arp inspection log** privileged EXEC command is affected. Two dashes ("--") appear instead of data except for the packet count and the time. No other statistics are provided for the entry. If you see this entry in the display, increase the number of entries in the log buffer or increase the logging rate.

Configuring the DAI Logging Buffer Size

To configure the DAI logging buffer size, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`ip arp inspection log-buffer entries`** *`number`*	Configures the DAI logging buffer size (range is 0 to 1024).
Step 3	`Router(config)#` **`do show ip arp inspection log | include Size`**	Verifies the configuration.

This example shows how to configure the DAI logging buffer for 64 messages:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection log-buffer entries 64
Router(config)# do show ip arp inspection log | include Size
Total Log Buffer Size : 64
```

Configuring the DAI Logging System Messages

To configure the DAI logging system messages, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`ip arp inspection log-buffer logs`** *`number_of_messages`* **`interval`** *`length_in_seconds`*	Configures the DAI logging buffer.
Step 3	`Router(config)#` **`do show ip arp inspection log`**	Verifies the configuration.

When configuring the DAI logging system messages, note the following information:

- For **logs** *number_of_messages* (default is 5), the range is 0 to 1024. A 0 value means that the entry is placed in the log buffer, but a system message is not generated.
- For **interval** *length_in_seconds* (default is 1), the range is 0 to 86400 seconds (1 day). A 0 value means that a system message is immediately generated (and the log buffer is always empty). An interval setting of 0 overrides a log setting of 0.
- System messages are sent at the rate of *number_of_messages* per *length_in_seconds*.

This example shows how to configure DAI logging to send 12 messages every 2 seconds:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection log-buffer logs 12 interval 2
Router(config)# do show ip arp inspection log | include Syslog
Syslog rate : 12 entries per 2 seconds.
```

This example shows how to configure DAI logging to send 20 messages every 60 seconds.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection log-buffer logs 20 interval 60
Router(config)# do show ip arp inspection log | include Syslog
Syslog rate : 20 entries per 60 seconds.
```

Configuring DAI Log Filtering

To configure DAI log filtering, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# ip arp inspection vlan vlan_range logging {acl-match {matchlog | none} | dhcp-bindings {all | none | permit}}`	Configures log filtering for each VLAN.
Step 3	`Router(config)# do show running-config | include ip arp inspection vlan vlan_range`	Verifies the configuration.

When configuring the DAI log filtering, note the following information:

- By default, all denied packets are logged.
- For *vlan_range*, you can specify a single VLAN or a range of VLANs:
  - To specify a single VLAN, enter a single VLAN number.
  - To specify a range of VLANs, enter a dash-separated pair of VLAN numbers.
  - You can enter a comma-separated list of VLAN numbers and dash-separated pairs of VLAN numbers.
- **acl-match matchlog**—Logs packets based on the DAI ACL configuration. If you specify the **matchlog** keyword in this command and the **log** keyword in the **permit** or **deny** ARP access-list configuration command, ARP packets permitted or denied by the ACL are logged.
- **acl-match none**—Does not log packets that match ACLs.
- **dhcp-bindings all**—Logs all packets that match DHCP bindings.
- **dhcp-bindings none**—Does not log packets that match DHCP bindings.
- **dhcp-bindings permit**—Logs DHCP-binding permitted packets.

This example shows how to configure the DAI log filtering for VLAN 100 not to log packets that match ACLs:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection vlan 100 logging acl-match none
Router(config)# do show running-config | include ip arp inspection vlan 100
ip arp inspection vlan 100 logging acl-match none
```

Displaying DAI Information

To display DAI information, use the privileged EXEC commands described in Table 56-2.

Table 56-2 Commands for Displaying DAI Information

Command	Description
show arp access-list [*acl_name*]	Displays detailed information about ARP ACLs.

Table 56-2 Commands for Displaying DAI Information (continued)

Command	Description
show ip arp inspection interfaces [*interface_id*]	Displays the trust state and the rate limit of ARP packets for the specified interface or all interfaces.
show ip arp inspection vlan *vlan_range*	Displays the configuration and the operating state of DAI for the specified VLAN. If no VLANs are specified or if a range is specified, displays information only for VLANs with DAI enabled (active).

To clear or display DAI statistics, use the privileged EXEC commands in Table 56-3.

Table 56-3 Commands for Clearing or Displaying DAI Statistics

Command	Description
clear ip arp inspection statistics	Clears DAI statistics.
show ip arp inspection statistics [**vlan** *vlan_range*]	Displays statistics for forwarded, dropped, MAC validation failure, IP validation failure, ACL permitted and denied, and DHCP permitted and denied packets for the specified VLAN. If no VLANs are specified or if a range is specified, displays information only for VLANs with DAI enabled (active).

For the **show ip arp inspection statistics** command, the switch increments the number of forwarded packets for each ARP request and response packet on a trusted DAI port. The switch increments the number of ACL-permitted or DHCP-permitted packets for each packet that is denied by source MAC, destination MAC, or IP validation checks, and the switch increments the appropriate failure count.

To clear or display DAI logging information, use the privileged EXEC commands in Table 56-4:

Table 56-4 Commands for Clearing or Displaying DAI Logging Information

Command	Description
clear ip arp inspection log	Clears the DAI log buffer.
show ip arp inspection log	Displays the configuration and contents of the DAI log buffer.

DAI Configuration Samples

This section includes these samples:

- Sample One: Two Switches Support DAI, page 56-16
- Sample Two: One Switch Supports DAI, page 56-20

Sample One: Two Switches Support DAI

This procedure shows how to configure DAI when two switches support this feature. Host 1 is connected to Switch A, and Host 2 is connected to Switch B as shown in Figure 56-2 on page 56-4. Both switches are running DAI on VLAN 1 where the hosts are located. A DHCP server is connected to Switch A. Both hosts acquire their IP addresses from the same DHCP server. Switch A has the bindings for Host 1 and Host 2, and Switch B has the binding for Host 2. Switch A Fast Ethernet port 6/3 is connected to the Switch B Fast Ethernet port 3/3.

Note

- DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses. For configuration information, see Chapter 54, "Configuring DHCP Snooping."
- This configuration does not work if the DHCP server is moved from Switch A to a different location.
- To ensure that this configuration does not compromise security, configure Fast Ethernet port 6/3 on Switch A and Fast Ethernet port 3/3 on Switch B as trusted.

Configuring Switch A

To enable DAI and configure Fast Ethernet port 6/3 on Switch A as trusted, follow these steps:

Step 1 Verify the connection between switches Switch A and Switch B:

```
SwitchA# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SwitchB          Fas 6/3            177          R S I     WS-C6506  Fas 3/3
SwitchA#
```

Step 2 Enable DAI on VLAN 1 and verify the configuration:

```
SwitchA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)# ip arp inspection vlan 1
SwitchA(config)# end
SwitchA# show ip arp inspection vlan 1

Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
    1     Enabled          Active

 Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
    1     Deny             Deny
SwitchA#
```

Step 3 Configure Fast Ethernet port 6/3 as trusted:

```
SwitchA# configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)# interface fastethernet 6/3
SwitchA(config-if)# ip arp inspection trust
SwitchA(config-if)# end
SwitchA# show ip arp inspection interfaces fastethernet 6/3

Interface        Trust State     Rate (pps)
 ---------------  -----------     ----------
 Fa6/3            Trusted              None
SwitchA#
```

Step 4 Verify the bindings:

```
SwitchA# show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:02:00:02:00:02   1.1.1.2          4993        dhcp-snooping  1     FastEthernet6/4
SwitchA#
```

Step 5 Check the statistics before and after DAI processes any packets:

```
SwitchA# show ip arp inspection statistics vlan 1

Vlan      Forwarded        Dropped     DHCP Drops     ACL Drops
 ----      ---------        -------     ----------     ----------
    1            0              0             0             0

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
    1            0              0                    0

 Vlan   Dest MAC Failures   IP Validation Failures
 ----   -----------------   ----------------------
    1                0                        0
SwitchA#
```

If Host 1 then sends out two ARP requests with an IP address of 1.1.1.2 and a MAC address of 0002.0002.0002, both requests are permitted, as reflected in the following statistics:

```
SwitchA# show ip arp inspection statistics vlan 1

 Vlan      Forwarded        Dropped     DHCP Drops     ACL Drops
 ----      ---------        -------     ----------     ----------
    1            2              0             0             0

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
    1            2              0                    0

 Vlan   Dest MAC Failures   IP Validation Failures
 ----   -----------------   ----------------------
    1                0                        0
SwitchA#
```

If Host 1 then tries to send an ARP request with an IP address of 1.1.1.3, the packet is dropped and an error message is logged:

```
00:12:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 2 Invalid ARPs (Req) on Fa6/4, vlan
1.([0002.0002.0002/1.1.1.3/0000.0000.0000/0.0.0.0/02:42:35 UTC Tue Jul 10 2001])
SwitchA# show ip arp inspection statistics vlan 1
SwitchA#
```

The statistics will display as follows:

```
Vlan      Forwarded        Dropped     DHCP Drops     ACL Drops
```

```
 ----      ---------        -------     ----------     ----------
    1             2              2             2              0

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
    1             2              0                     0

 Vlan   Dest MAC Failures   IP Validation Failures
 ----   -----------------   ----------------------
    1                 0                         0
SwitchA#
```

Configuring Switch B

To enable DAI and configure Fast Ethernet port 3/3 on Switch B as trusted, follow these steps:

Step 1 Verify the connectivity:

```
SwitchA# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SwitchB          Fas 3/3            120         R S I     WS-C6506  Fas 6/3
SwitchB#
```

Step 2 Enable DAI on VLAN 1, and verify the configuration:

```
SwitchB# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchB(config)# ip arp inspection vlan 1
SwitchB(config)# end
SwitchB# show ip arp inspection vlan 1

Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
    1     Enabled          Active

 Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
    1     Deny             Deny
SwitchB#
```

Step 3 Configure Fast Ethernet port 3/3 as trusted:

```
SwitchB# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchB(config)# interface fastethernet 3/3
SwitchB(config-if)# ip arp inspection trust
SwitchB(config-if)# end
SwitchB# show ip arp inspection interfaces


Interface        Trust State     Rate (pps)
 ---------------  -----------     ----------
 Gi1/1            Untrusted              15
```

```
 Gi1/2            Untrusted              15
 Gi3/1            Untrusted              15
 Gi3/2            Untrusted              15
 Fa3/3            Trusted              None
 Fa3/4            Untrusted              15
 Fa3/5            Untrusted              15
 Fa3/6            Untrusted              15
 Fa3/7            Untrusted              15

<output truncated>
SwitchB#
```

Step 4 Verify the list of DHCP snooping bindings:

```
SwitchB# show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:01:00:01:00:01   1.1.1.1          4995        dhcp-snooping  1     FastEthernet3/4
SwitchB#
```

Step 5 Check the statistics before and after DAI processes any packets:

```
SwitchB# show ip arp inspection statistics vlan 1

Vlan      Forwarded        Dropped     DHCP Drops     ACL Drops
 ----      ---------        -------     ----------     ----------
    1              0              0              0              0

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
    1              0              0                     0

 Vlan   Dest MAC Failures   IP Validation Failures
 ----   -----------------   ----------------------
    1                  0                        0
SwitchB#
```

If Host 2 then sends out an ARP request with the IP address 1.1.1.1 and the MAC address 0001.0001.0001, the packet is forwarded and the statistics are updated appropriately:

```
SwitchB# show ip arp inspection statistics vlan 1

Vlan      Forwarded        Dropped     DHCP Drops     ACL Drops
 ----      ---------        -------     ----------     ----------
    1              1              0              0              0

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
    1              1              0                     0

 Vlan   Dest MAC Failures   IP Validation Failures
 ----   -----------------   ----------------------
    1                  0                        0
SwitchB#
```

If Host 2 attempts to send an ARP request with the IP address 1.1.1.2, DAI drops the request and logs a system message:

```
00:18:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa3/4, vlan
1.([0001.0001.0001/1.1.1.2/0000.0000.0000/0.0.0.0/01:53:21 UTC Fri May 23 2003])
SwitchB#
```

The statistics display as follows:

```
SwitchB# show ip arp inspection statistics vlan 1
```

```
Vlan      Forwarded        Dropped     DHCP Drops     ACL Drops
 ----      ---------        -------     ----------     ----------
    1              1              1              1              0

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
    1              1              0                     0

 Vlan   Dest MAC Failures   IP Validation Failures
 ----   -----------------   ----------------------
    1                  0                        0
SwitchB#
```

Sample Two: One Switch Supports DAI

This procedure shows how to configure DAI when Switch B shown in Figure 56-2 on page 56-4 does not support DAI or DHCP snooping.

If switch Switch B does not support DAI or DHCP snooping, configuring Fast Ethernet port 6/3 on Switch A as trusted creates a security hole because both Switch A and Host 1 could be attacked by either Switch B or Host 2.

To prevent this possibility, you must configure Fast Ethernet port 6/3 on Switch A as untrusted. To permit ARP packets from Host 2, you must set up an ARP ACL and apply it to VLAN 1. If the IP address of Host 2 is not static, which would make it impossible to apply the ACL configuration on Switch A, you must separate Switch A from Switch B at Layer 3 and use a router to route packets between them.

To set up an ARP ACL on switch Switch A, follow these steps:

Step 1 Configure the access list to permit the IP address 1.1.1.1 and the MAC address 0001.0001.0001, and verify the configuration:

```
SwitchA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)# arp access-list H2
SwitchA(config-arp-nacl)# permit ip host 1.1.1.1 mac host 1.1.1
SwitchA(config-arp-nacl)# end
SwitchA# show arp access-list
ARP access list H2
    permit ip host 1.1.1.1 mac host 0001.0001.0001
```

Step 2 Apply the ACL to VLAN 1, and verify the configuration:

```
SwitchA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)# ip arp inspection filter H2 vlan 1
SwitchA(config)# end
SwitchA#

SwitchA# show ip arp inspection vlan 1

Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
```

```
   1      Enabled          Active      H2                 No

 Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
   1      Deny             Deny
SwitchA#
```

Step 3 Configure Fast Ethernet port 6/3 as untrusted, and verify the configuration:

```
SwitchA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)# interface fastethernet 6/3
SwitchA(config-if)# no ip arp inspection trust
SwitchA(config-if)# end
Switch# show ip arp inspection interfaces fastethernet 6/3

 Interface        Trust State     Rate (pps)
 ---------------  -----------     ----------
 Fa6/3            Untrusted               15

Switch#
```

When Host 2 sends 5 ARP requests through Fast Ethernet port 6/3 on Switch A and a “get” is permitted by Switch A, the statistics are updated appropriately:

```
Switch# show ip arp inspection statistics vlan 1
Vlan      Forwarded        Dropped     DHCP Drops     ACL Drops
 ----      ---------        -------     ----------     ----------
   1              5              0              0              0
Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
   1              0              5                    0
Vlan   Dest MAC Failures   IP Validation Failures
 ----   -----------------   ----------------------
   1                  0                        0
Switch#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Traffic Storm Control

This chapter describes how to configure the traffic storm control feature in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. The traffic storm control feature prevents LAN ports from being disrupted by a broadcast, multicast, or unicast traffic storm on physical interfaces.

Traffic storm control (also called traffic suppression) monitors incoming traffic levels over a 1-second traffic storm control interval, and during the interval it compares the traffic level with the traffic storm control level that you configure. The traffic storm control level is a percentage of the total available bandwidth of the port. Each port has a single traffic storm control level that is used for all types of traffic (broadcast, multicast, and unicast).

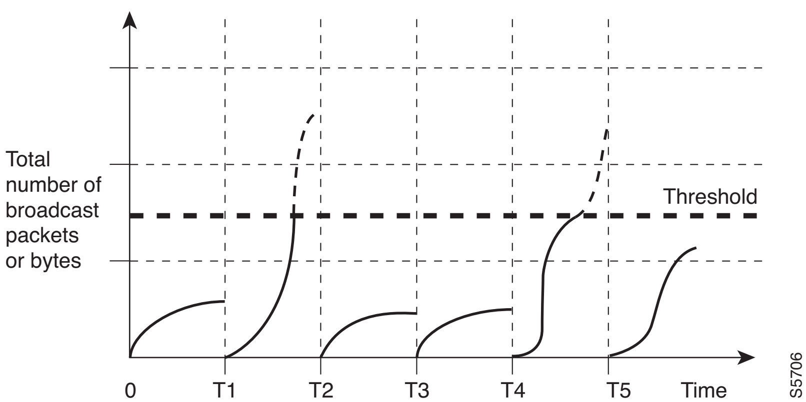
Traffic storm control monitors the level of each traffic type for which you enable traffic storm control in 1-second traffic storm control intervals.

In all releases, and by default in Release 12.2(33)SXJ and later releases, within an interval, when the ingress traffic for which traffic storm control is enabled reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the traffic storm control interval ends. Release 12.2(33)SXJ and later releases support these configurable traffic storm control optional actions:

- Shutdown—When a traffic storm occurs, traffic storm control puts the port into the error-disabled state. To reenable ports, use the error-disable detection and recovery feature or the **shutdown** and **no shutdown** commands.
- Trap—When a traffic storm occurs, traffic storm control generates an SNMP trap.

Figure 57-1 shows the broadcast traffic patterns on a LAN interface over a specific interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 57-1 Broadcast Suppression



The traffic storm control threshold numbers and the time interval combination make the traffic storm control algorithm work with different levels of granularity. A higher threshold allows more packets to pass through.

Traffic storm control is implemented in hardware. The traffic storm control circuitry monitors packets passing from a LAN interface to the switching bus. Using the Individual/Group bit in the packet destination address, the traffic storm control circuitry determines if the packet is unicast or broadcast, keeps track of the current count of packets within the 1-second interval and when the threshold is reached, traffic storm control filters out subsequent packets.

Because hardware traffic storm control uses a bandwidth-based method to measure traffic, the most significant implementation factor is setting the percentage of total available bandwidth that can be used by controlled traffic. Because packets do not arrive at uniform intervals, the 1-second interval during which controlled traffic activity is measured can affect the behavior of traffic storm control.

The following are examples of traffic storm control behavior:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast traffic until the end of the traffic storm control interval.
- If you enable broadcast and multicast traffic storm control, and the combined broadcast and multicast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.

- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.

Default Traffic Storm Control Configuration

Traffic storm control is disabled by default.

Configuration Guidelines and Restrictions

When configuring traffic storm control, follow these guidelines and restrictions:

- FlexWAN Fast Ethernet port adapters and all WAN modules supporting Ethernet SPAs do not support traffic storm control.
- The following LAN switching modules do not support traffic storm control:
  - WS-X6148E-GE-45AT
  - WS-X6148A-GE-45AF
  - WS-X6148A-GE-TX
  - WS-X6148-GE-45AF
  - WS-X6148-GE-TX
  - WS-X6148V-GE-TX
  - WS-X6548-GE-45AF
  - WS-X6548-GE-TX
  - WS-X6548V-GE-TX
- The switch supports multicast and unicast traffic storm control on Gigabit and 10-Gigabit Ethernet LAN ports. Most FastEthernet switching modules do not support multicast and unicast traffic storm control, with the exception of WS-X6148A-RJ-45 and the WS-X6148-SFP.
- The switch supports broadcast traffic storm control on all LAN ports except on those modules previously noted.
- Except for BPDUs, traffic storm control does not differentiate between control traffic and data traffic.
- When multicast suppression is enabled, traffic storm control suppresses BPDUs when the multicast suppression threshold is exceeded on these modules:
  - WS-X6748-SFP
  - WS-X6724-SFP
  - WS-X6748-GE-TX
  - WS-X6748-GE-TX
  - WS-X6704-10GE
  - WS-SUP32-GE-3B

- WS-SUP32-10GE-3B

When multicast suppression is enabled on the listed modules, do not configure traffic storm control on STP-protected ports that need to receive BPDUs.

Except on the listed modules, traffic storm control does not suppress BPDUs.

Configuring Traffic Storm Control

These sections describe how to configure traffic storm control:

- Enabling Traffic Storm Control, page 57-4
- Configuring the Traffic Storm Control Shutdown Mode, page 57-6
- Configuring Traffic Storm Control SNMP Traps, page 57-6

Enabling Traffic Storm Control

To enable traffic storm control on an interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`number`*`}}`	Selects an interface to configure.
Step 2	`Router(config-if)#` **`storm-control broadcast level`** *`level`*`[.`*`level`*`]`	Enables broadcast traffic storm control on the interface, configures the traffic storm control level, and applies the traffic storm control level to all traffic storm control modes enabled on the interface.
Step 3	`Router(config-if)#` **`storm-control multicast level`** *`level`*`[.`*`level`*`]` **Note** The **storm-control multicast** command is supported only on Gigabit and 10-Gigabit Ethernet interfaces.	Enables multicast traffic storm control on the interface, configures the traffic storm control level, and applies the traffic storm control level to all traffic storm control modes enabled on the interface.
Step 4	`Router(config-if)#` **`storm-control unicast level`** *`level`*`[.`*`level`*`]` **Note** The **storm-control unicast** command is supported only on Gigabit and 10-Gigabit Ethernet interfaces.	Enables unicast traffic storm control on the interface, configures the traffic storm control level, and applies the traffic storm control level to all traffic storm control modes enabled on the interface.
Step 5	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 6	`Router#` **`show running-config`** *`interface`*	Verifies the configuration.

1. *type* = **fastethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring the traffic storm control level, note the following information:

- You can configure traffic storm control on the port channel interface of an EtherChannel.
- Do not configure traffic storm control on ports that are members of an EtherChannel. Configuring traffic storm control on ports that are configured as members of an EtherChannel puts the ports into a suspended state.

- Specify the level as a percentage of the total interface bandwidth:
  - The level can be from 0 to 100.
  - The optional fraction of a level can be from 0 to 99.
  - 100 percent means no traffic storm control.
  - 0.0 percent suppresses all traffic.
- On these modules, these levels suppress all traffic:
  - WS-X6704-10GE: 0.33 percent or less
  - WS-X6724-SFP 10Mbps ports: 0.33 percent or less
  - WS-X6748-SFP 100Mbps ports: 0.03 percent or less
  - WS-X6748-GE-TX 100Mbps ports: 0.03 percent or less
  - WS-X6716-10G-3C, WS-X6716-10G-3CXL Oversubscription Mode: 0.29 percent or less
  - WS-X6716-10T-3C, WS-X6716-10T-3CXL Oversubscription Mode: 0.29 percent or less

Because of hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames making up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.

All routers in a VLAN see copies of all broadcast traffic. To avoid high RP CPU utilization caused by a high volume of broadcast traffic, the threshold typically is set to a very low value; for example, less than 1 percent on a Gigabit Ethernet port.

You can use the Top N feature to periodically measure the peak broadcast traffic levels of the selected ports. If you have a specific required broadcast traffic level (for example, from an application), you can use that requirement as the basis of the threshold.

Base the suppression threshold on your data, plus some additional capacity. For example, if the peak broadcast traffic that is acceptable for a port is 1 percent, a threshold of 1.5 percent might be appropriate. The faster the port speed, the less additional capacity is required.

Use the **show interfaces counters storm-control** command to monitor the effect of the values that you configure, and increase the configured threshold if the TotalSuppDiscards column shows nonzero values.

This example shows how to enable multicast traffic storm control on Gigabit Ethernet interface 3/16 and how to configure the traffic storm control level at 0.5 percent:

```
Router# configure terminal
Router(config)# interface gigabitethernet 3/16
Router(config-if)# storm-control multicast level 0.5
Router(config-if)# end
```

This example shows how the traffic storm control level configured for one mode affects all other modes that are already configured on the Gigabit Ethernet interface 4/10:

```
Router# show run inter gig4/10
Building configuration...

Current configuration : 176 bytes
!
Router# interface GigabitEthernet4/10
Router# switchport
Router# switchport mode access
Router# storm-control broadcast level 0.5
Router# storm-control multicast level 0.5
Router# spanning-tree portfast edge
Router# end
```

```
Router# configure terminal
Router(config)# interface gigabitethernet 4/10
Router(config-if)# storm-control unicast level 0.7
Router(config-if)# end

Router# show interfaces gig4/10 counters storm-control

Port      UcastSupp %     McastSupp %     BcastSupp %  TotalSuppDiscards
Gi4/10      00.70           00.70           00.70                0

Router#
```

Configuring the Traffic Storm Control Shutdown Mode

To configure the traffic storm control shutdown mode on an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {{*type*[1] *slot/port*} | {**port-channel** *number*}}	Selects an interface to configure.
Step 2	Router(config-if)# **storm-control action shutdown**	(Optional) Configures traffic storm control to error-disable ports when a traffic storm occurs. • Enter the **no storm-control action shutdown** command to revert to the default action (drop). • Use the error disable detection and recovery feature, or the **shutdown** and **no shutdown** commands to reenable ports.
Step 3	Router(config-if)# **end**	Exits configuration mode.
Step 4	Router# **show running-config** *interface*	Verifies the configuration.

1. *type* = **fastethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure the traffic storm control shutdown mode on Gigabit Ethernet interface 3/16:

```
Router# configure terminal
Router(config)# interface gigabitethernet 3/16
Router(config-if)# storm-control action shutdown
Router(config-if)# end
```

Configuring Traffic Storm Control SNMP Traps

To configure traffic storm control SNMP traps, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {{*type*[1] *slot/port*} | {**port-channel** *number*}}	Selects an interface to configure.
Step 2	Router(config-if)# **storm-control action trap**	Configures traffic storm control to generate an SNMP trap when a storm is detected on the port.
Step 3	Router(config-if)# **exit**	Exits interface configuration mode.

	Command	Purpose
Step 4	`Router(config)# snmp-server enable traps storm-control trap-rate value`	Configures the maximum number of storm-control traps sent per minute. The range is 0 to 1000; the default is 0 (no limit is imposed; a trap is sent at every occurrence).
Step 5	`Router(config)# end`	Exits configuration mode.
Step 6	`Router# show running-config interface`	Verifies the interface configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure Gigabit Ethernet interface 3/16 to send an SNMP trap when a traffic storm is detected on the port and how to revert traffic storm control trap rate limiting to the default value:

```
Router# configure terminal
Router(config)# interface gigabitethernet 3/16
Router(config-if)# storm-control action trap
Router(config-if)# exit
Router(config)# snmp-server enable traps storm-control trap-rate 0
Router(config)# end
```

Displaying Traffic Storm Control Settings

To display traffic storm control information, use the commands described in Table 57-1.

Table 57-1 Commands for Displaying Traffic Storm Control Status and Configuration

Command	Purpose
`Router# show interfaces [{type1 slot/port}	{port-channel number}] switchport`	Displays the administrative and operational status of all Layer 2 LAN ports or the specified Layer 2 LAN port.
`Router# show interfaces [{type1 slot/port}	{port-channel number}] counters storm-control`  `Router# show interfaces counters storm-control [module slot_number]`	Displays the total number of packets discarded for all three traffic storm control modes, on all interfaces or on the specified interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note The **show interfaces** [{*interface_type slot/port*} | {**port-channel** *number*}] **counters** command does not display the discard count. You must the **storm-control** keyword to display the discard count.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Unknown Unicast and Multicast Flood Control

This chapter describes how to configure the unknown unicast flood blocking (UUFB), unknown multicast flood blocking (UMFB), and unknown unicast flood rate-limiting (UUFRL) features in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Understanding Unknown Traffic Flood Control

By default, unknown unicast and multicast traffic is flooded to all Layer 2 ports in a VLAN. You can use the UUFB, UMFB, and UUFRL features to prevent or limit this traffic.

The UUFB and UMFB features block unknown unicast and multicast traffic flooding at a specific port, only permitting egress traffic with MAC addresses that are known to exist on the port. The UUFB and UMFB features are supported on all ports that are configured with the **switchport** command, including private VLAN (PVLAN) ports.

The UUFRL feature applies a rate limit globally to unknown unicast traffic on all VLANs.

Note Enter the **switchport block multicast** command only on ports where all unknown multicast flooded traffic needs to be completely blocked. UMFB disrupts protocols that make use of local subnetwork multicast control groups in the 224.0.0.0/24 range, for example:

- ARP
- IPv6 neighbor discovery (IPv6 ND)
- Network Time Protocol (NTP)

Do not enter this command on nonreceiver (router) ports or host ports that rely on dynamic ARP. Use IGMP snooping or other rate-limiting options to restrict, rather than completely block, unknown multicast flooded traffic.

Configuring UUFB or UMFB

To configure UUFB or UFMB, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface** {{*type*[1] *slot/port*} | {**port-channel** *number*}}	Selects the interface to configure.
Step 3	Router(config-if)# **switchport**	Configures the port for Layer 2 switching.
Step 4	Router(config-if)# **switchport block** {**unicast** | **multicast**}	Enables unknown unicast or multicast flood blocking on the port.
Step 5	Router(config-if)# **do show interfaces** [*type*[1] *slot/port*] **switchport | include Unknown**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure UUFB on Fast Ethernet port 5/12 and how to verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport
Router(config-if)# switchport block unicast
Router(config-if)# do show interface fastethernet 5/12 switchport | include Unknown
Unknown unicast blocked: enabled
```

Configuring UUFRL

Note The UUFRL feature is available only with the Supervisor Engine 720-10GE.

To configure UUFRL, perform this task:

	Command	Purpose
Step 1	`Router# `**`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)# `**`mls rate-limit layer2 unknown`** *`rate-in-pps`* `[`*`burst-size`*`]`	Enables UUFRL and sets the maximum packet rate. (Optional) Specify a burst size limit.
Step 3	`Router(config)# `**`exit`**	Exits configuration mode.

When you configure UUFRL, note the following information:

- When unknown unicast flood rate-limiting (UUFRL) is enabled, per-VLAN learning must be enabled on all the Layer 3 routed ports, otherwise, any unicast flooded packet coming into a routed port will also be rate-limited by UUFRL.
- For the *rate-in-pps* value:
  - The range is 10 through 1,000,000 (entered as 1000000).
  - There is no default value.
  - Values lower than 1,000 (entered as 1000) should offer sufficient protection.
- For the *burst-size* value:
  - The range is 1 through 255.
  - The default is 10.
  - The default value should provide sufficient protection.

This example shows how to configure UUFRL with a rate limit of 1000 pps with a burst of 20 packets:

```
Router# configure terminal
Router(config)# mls rate-limit layer2 unknown 1000 20
Router(config)# exit
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Network Admission Control

This chapter describes how to configure Network Admission Control (NAC) in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- The *Cisco IOS Security Command Reference*, Release 12.3 at this URL:

  http://www.cisco.com/en/US/products/ps6017/prod_command_reference_list.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains these sections:

Understanding NAC

These sections describe NAC:

NAC Overview

NAC is part of the Cisco Self-Defending Network Initiative that helps you identify, prevent, and adapt to security threats in your network. Because of the increased threat and impact of worms and viruses to networked businesses, NAC allows you to check and validate the antivirus status of endpoints or clients before granting network access.

Release 12.2(18)SXF2 and later releases supports NAC Layer 2 IP validation. Release 12.2(33)SXH and rebuilds support NAC Layer 3 IP validation. Release 12.2(33)SXI and later releases do not support NAC Layer 3 IP validation.

NAC Layer 2 IP (also known as LAN Port IP) operates on Layer 2 ports on edge switches. NAC Layer 2 IP validation has different methods for validation initiation, message exchange, and policy enforcement from the NAC Layer 2 IEEE 802.1x. LAN Port IP does not require IEEE 802.1x support on the host PCs. For additional information about IEEE 802.1x, see Chapter 60, "Configuring IEEE 802.1X Port-Based Authentication."

NAC Layer 3 IP (also known as NAC Gateway IP) operates on Layer 3 interfaces on distribution layer switches. An advantage of NAC Layer 3 IP is that access layer switches do not require any changes to use the NAC feature.

Note The NAC feature applies access controls only to IPv4 traffic. NAC does not restrict Layer 2-bridged traffic.

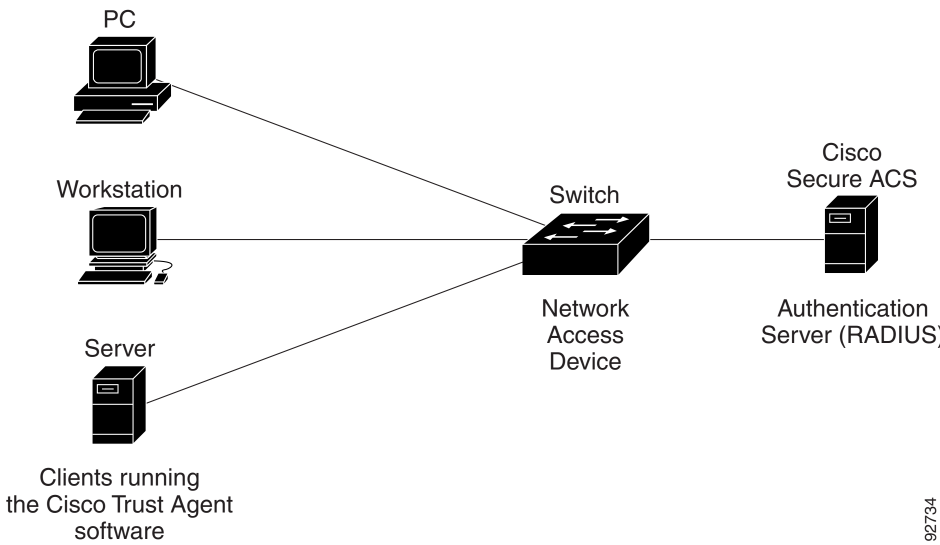
NAC provides *posture validation* for routed traffic. Posture validation reduces the exposure of a virus to the network. This feature allows network access based on the antivirus credentials of the network device that is requesting network access. These credentials may be antivirus software, a virus definitions file, or a particular virus scan engine version. Based on the antivirus credentials of the host, the requesting device is allowed access to the network or is restricted from network access.

If the client host fails the credential validation, then partial access to the network can be allowed by using the *remediation* feature. The remediation process redirects HTTP traffic from the client host to a web page URL that provides access to the latest antivirus files. The URL used by the remediation process resolves to a remediation server address defined as a part of the network access policy. The remediation server is where the latest antivirus files are located. These antivirus files can be downloaded or upgraded from this location.

NAC Device Roles

The devices in the network have specific roles when you use NAC as shown in Figure 59-1.

Figure 59-1 Posture Validation Devices



The following devices that support NAC on the network perform these roles:

- Endpoint system or client—This is a device (host) on the network such as a PC, workstation, or server. The host, which is running the Cisco Trust Agent (CTA) software, requests access to the LAN and switch services and responds to requests from the switch. This endpoint system is a potential source of virus infections, and its antivirus status needs to be validated before the host is granted network access.
  - For NAC Layer 2 IP, the device is connected to an access port through a direct connection, an IP phone, or a wireless access point.
  - For NAC Layer 3 IP, the device is one or more Layer 3 hops away from the switch.

  The CTA software is also referred to as the *posture agent* or the *antivirus client*.
- Switch—This is a network access device, which provides validation services and policy enforcement.
  - Edge switch—This is the network access device that provides NAC Layer 2 IP validation services and policy enforcement at the network edge and controls the physical access to the network based on the access policy of the client.
  - Distribution switch—This is the NAC gateway, which provides validation services and policy enforcement at the Layer 3 network edge and controls access to the Layer 3 network based on the access policy of the client.

  The encapsulation information in the EAP messages can be based on the User Datagram Protocol (UDP). When using UDP, the switch uses EAP over UDP (EAPoUDP) frames, which are also referred to as EoU frames.

  The switch relays Extensible Authentication Protocol (EAP) messages between the endpoints and the authentication server.

- Authentication server—This device performs the actual validation of the client. The authentication server validates the antivirus status of the client, determines the access policy, and notifies the switch whether the client is authorized to access the LAN and switch services. Because the switch acts as the proxy, the EAP message exchange between the switch and authentication server is transparent to the switch.

  The switch supports the Cisco Secure Access Control Server (ACS) Version 4.0 or later with RADIUS, authentication, authorization, and accounting (AAA), and EAP extensions.

  The authentication server is also referred to as the *posture server.*

AAA Down Policy

The AAA down policy is a method of allowing a host to remain connected to the network if the AAA server is not available. Typical deployments of NAC use Cisco Secure ACS to validate the client posture and to pass policies back to the Network Access Device (NAD). If the AAA server cannot be reached when the posture validation occurs, instead of rejecting the user (that is, not providing the access to the network), an administrator can configure a default AAA down policy that can be applied to the host.

This policy is advantageous for the following reasons:

- While AAA is unavailable, the host will still have connectivity to the network, although it may be restricted.
- When the AAA server is again available, a user can be revalidated, and the user's policies can be downloaded from the ACS.

Note When the AAA server is down, the AAA down policy is applied only if there is no existing policy associated with the host. Typically, during revalidation when the AAA server goes down, the policies being used for the host are retained.

When the AAA policy is applied, the session state is maintained as AAA DOWN.

NAC IP Validation

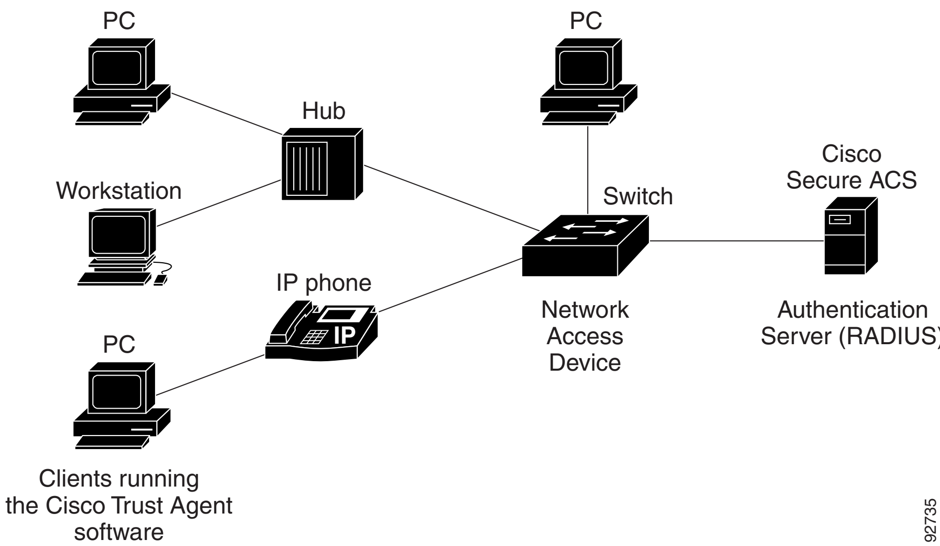
The following sections describe NAC IP validation:

NAC Layer 2 IP Validation

You can use NAC Layer 2 IP on the access port of an edge switch to which an endpoint system or client is connected. The device (host or client) can be a PC, a workstation, or a server that is connected to the access port through a direct connection, an IP phone, or a wireless access point, as shown in Figure 59-2.

When NAC Layer 2 IP is enabled, EAPoUDP only works with IPv4 traffic. The switch checks the antivirus status of the endpoint devices or clients and enforces access control policies.

Figure 59-2 Network Using NAC Layer 2 IP



NAC Layer 2 IP supports the posture validation of multiple hosts on the same Layer 2 port, as shown in Figure 59-2.

When you enable NAC Layer 2 IP validation on a Layer 2 port to which hosts are connected, the switch can use DHCP snooping and Address Resolution Protocol (ARP) snooping to identify connected hosts. The switch initiates posture validation after receiving an ARP packet or creating a DHCP snooping binding entry. When you enable NAC Layer 2 IP validation, ARP snooping is the default method to detect connected hosts. If you want the switch to detect hosts when a DHCP snooping binding entry is created, you must enable DHCP snooping.

NAC Layer 3 IP Validation

The gateway IP feature supports two types of posture validation (PV) triggers:

- Intercept ACLs
- ARP-based triggers (SVI interfaces only)

Intercept ACLs are configured to intercept inbound traffic from the clients and initiate PV if the traffic matches the ACL.

If the IP admission rule configured on an interface does not have an associated intercept ACL, the gateway uses ARP or DHCP events to trigger posture validation on the interface.

When the gateway learns a new client in its ARP cache, it initiates PV with the client. Using the ARP-based triggers reduces ACL TCAM usage, but it is only valid for clients that are one Layer 3 hop away from the gateway. Therefore, you can configure only SVIs (and not router ports) to have IP admission rules without an intercept ACL.

Posture Validation

If only dynamic ARP inspection is enabled on the access VLAN assigned to a Layer 2 port, posture validation is initiated when ARP packets pass the dynamic ARP inspection validation checks. However, if DHCP snooping and dynamic ARP inspection are enabled, when you create a DHCP snooping binding entry, posture validation is initiated through DHCP.

When posture validation is initiated, the switch creates an entry in the session table to track the posture validation status of the host and follows this process to determine the NAC policy:

1. If the host is in the exception list, the switch applies the user-configured NAC policy to the host.
2. If EoU bypass is enabled, the switch sends a nonresponsive-host request to the Cisco Secure ACS and applies the access policy from the server to the host. The switch inserts a RADIUS AV pair to the request to specify that the request is for a nonresponsive host.
3. If EoU bypass is disabled, the switch sends an EAPoUDP hello packet to the host, requesting the host antivirus condition. If no response is received from the host after the specified number of attempts, the switch classifies the host as clientless, and the host is considered to be a nonresponsive host. The switch sends a nonresponsive-host request to the Cisco Secure ACS and applies the access policy from the server to the host.

Note If a DHCP snooping binding entry for a client is deleted, the switch removes the client entry in the session table, and the client is no longer authenticated.

Exception Lists

An exception list has local profile and policy configurations. Use the identity profile to statically authorize or validate devices based on the IP address, MAC address, or device type. An identity profile is associated with a local policy that specifies the access control attributes.

You can bypass posture validation of specific hosts by specifying those hosts in an exception list and applying a user-configured policy to the hosts. After the entry is added to the EAPoUDP session table, the switch compares the host information to the exception list. If the host is in the exception list, the switch applies the configured NAC policy to the host. The switch also updates the EAPoUDP session table with the validation status of the client as POSTURE ESTAB.

EoU Bypass

The switch can use the EoU bypass feature to speed up posture validation of hosts that are not using the CTA. If EoU bypass is enabled, the switch does not contact the host to request the antivirus condition. Instead, the switch sends a request to the Cisco Secure ACS that includes the IP address, MAC address, service type, and EAPoUDP session ID of the host. The Cisco Secure ACS makes the access control decision and sends the policy to the switch.

If EoU bypass is enabled and the host is nonresponsive, the switch sends a nonresponsive-host request to the Cisco Secure ACS and applies the access policy from the server to the host.

If EoU bypass is enabled and the host uses CTA, the switch also sends a nonresponsive-host request to the Cisco Secure ACS and applies the access policy from the server to the host.

EAPoUDP Sessions

If the EoU bypass is disabled, the switch sends an EAPoUDP packet to initiate posture validation. While posture validation occurs, the switch enforces the default access policy. After the switch sends an EAPoUDP message to the host and the host responds to the antivirus condition request, the switch forwards the EAPoUDP response to the Cisco Secure ACS. If no response is received from the host after the specified number of attempts, the switch classifies the host as nonresponsive. After the ACS validates the credentials, the authentication server returns an Access-Accept message with the posture token and the policy attributes to the switch. The switch updates the EAPoUDP session table and enforces the access limitations, which provides segmentation and quarantine of poorly postured clients, or by denying network access.

There are two types of policies that apply during posture validation:

- Host Policy—The host policy consists of an ACL that enforces the access limitations as determined by the outcome of posture validation.
- URL Redirect Policy—The URL redirect policy provides a method to redirect all HTTP or HTTPS traffic to a remediation server that allows a noncompliant host to perform the necessary upgrade actions to become compliant.

The operation of the URL-redirect deny ACEs (typically to bypass the redirection of the HTTP traffic destined to remediation servers) is that the traffic to these ACEs is forwarded in hardware without applying the default interface and the downloaded host policies.

If this traffic (that is, the traffic that matches the deny URL redirect ACEs) is required to be filtered, you need to define a VLAN ACL on the Layer 2 access VLAN.

The URL redirect policy consists of the following:

- A URL that points to the remediation server.
- An ACL on the switch that causes all HTTP or HTTPS packets from the host other than those destined to the remediation server address to be captured and redirected to the switch software for the necessary HTTP redirection.

The ACL name for the host policy, the redirect URL, and the URL redirect ACL are conveyed using RADIUS Attribute-Value objects.

Cisco Secure ACS and AV Pairs

When NAC IP validation is enabled, the Cisco Secure ACS provides NAC AAA services by using RADIUS. Cisco Secure ACS gets information about the antivirus status of the endpoint system and validates the antivirus condition of the endpoint.

You can set these Attribute-Value (AV) pairs on the Cisco Secure ACS by using the RADIUS *cisco-av-pair* vendor-specific attributes (VSAs):

- CiscoSecure-Defined-ACL—Specifies the names of the downloadable ACLs on the Cisco Secure ACS. The switch gets the ACL name through the CiscoSecure-Defined-ACL AV pair in this format:

  #ACL#-IP-*name*-*number*

  name is the ACL name and *number* is the version number, such as 3f783768.

  The Auth-Proxy posture code checks if the access control entries (ACEs) of the specified downloadable ACL were previously downloaded. If they were not, the Auth-Proxy posture code sends an AAA request with the downloadable ACL name as the username so that the ACEs are downloaded. The downloadable ACL is then created as a named ACL on the switch. This ACL has ACEs with a source address of any and does not have an implicit deny statement at the end. When the downloadable ACL is applied to an interface after posture validation is complete, the source address is changed from any to the host source IP address. The ACEs are prepended to the downloadable ACL applied to the switch interface to which the endpoint device is connected. If traffic matches the CiscoSecure-Defined-ACL ACEs, the appropriate NAC actions are taken.

- url-redirect and url-redirect-acl—Specifies the local URL policy on the switch. The switches use these cisco-av-pair VSAs as follows:
  - url-redirect = <HTTP or HTTPS URL>
  - url-redirect-acl = ACL name or number

  These AV pairs enable the switch to intercept an HTTP or HTTPS request from the endpoint device and forward the client web browser to the specified redirect address from which the latest antivirus files can be downloaded. The url-redirect AV pair on the Cisco Secure ACS contains the URL to which the web browser will be redirected. The url-redirect-acl AV pair contains the name or number of an ACL that specifies the HTTP or HTTPS traffic to be redirected. The ACL must be defined on the switch. Traffic that matches a permit entry in the redirect ACL will be redirected.

  These AV pairs may be sent if the host's posture is not healthy.

Note You can redirect the URL for either HTTP or HTTPS traffic but not for both at the same time. This situation occurs because the Cisco IOS software HTTP server can either listen to the HTTP port or to the HTTPS port but cannot listen to both at the same time.

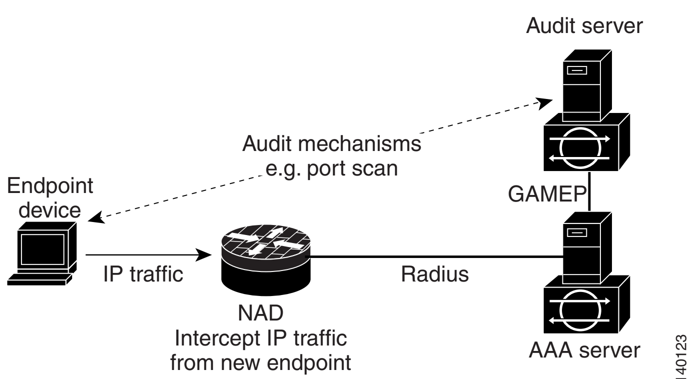
For more information about AV pairs that are supported by Cisco IOS software, see the ACS configuration and command reference documentation about the software releases running on the AAA clients.

Audit Servers

End devices that do not run Cisco Trust Agent (CTA) will not be able to provide credentials when challenged by Network Access Devices. These devices are described as *agentless* or *nonresponsive*. The NAC architecture has been extended to incorporate audit servers. An audit server is a third-party server that can probe, scan, and determine security compliance of a host without the need for presence of Cisco trust agent on the host. The result of the audit server examination can influence the access servers to make host-specific network access policy decisions instead of enforcing a common restrictive policy for all nonresponsive hosts. You can build more robust host audit and examination functionality by integrating any third-party audit operations into the NAC architecture.

Figure 59-3 shows how audit servers fit into the typical topology.

Figure 59-3 NAC Device Roles



The architecture assumes that the audit server can be reached so that the host can communicate with it. When a host (endpoint device) makes network access through the NAD configured for posture validation, the network access device eventually requests the AAA server (Cisco Secure ACS) for an access policy to be enforced for the host. The AAA server can be configured to trigger a scan of the host with an external audit server. The audit server scan occurs asynchronously and can take several seconds to complete. During the time of the audit server scan, the AAA server conveys a minimal restrictive security policy to NAD for enforcement along with a short poll timer (session-timeout). The NAD polls the AAA server at the specified timer interval until the result is available from the audit server. After the AAA server receives the audit result, it computes an access policy based on the audit result and is sent down to NAD for enforcement on its next request.

ACLs

If you configure NAC IP validation on an interface, you must also configure a default security ACL on the same interface. The default ACL is applied to IP traffic for hosts that have not completed posture validation.

If the default ACL is configured on the switch and the Cisco Secure ACS sends a host access policy to the switch, the switch applies the policy to traffic from the host connected to a Layer 2 port. If the policy applies to the traffic, the switch forwards the traffic. If the policy does not apply, the switch applies the default ACL. If there is no default ACL configured, the traffic is permitted.

If the Cisco Secure ACS sends the switch a downloadable ACL that specifies a redirect URL as a policy-map action, this ACL takes precedence over the default ACL already configured on the Layer 2 port. The redirect URL ACL policy also takes precedence over the policy already configured on the host. If the default port ACL is not configured on the switch, the switch can still apply the downloadable ACL from the Cisco Secure ACS.

NAC Timers

The switch supports these timers:

- Hold Timer, page 59-9
- Idle Timer, page 59-10
- Retransmission Timer, page 59-11
- Revalidation Timer, page 59-11
- Status-Query Timer, page 59-11

Hold Timer

The hold timer prevents a new EAPoUDP session from immediately starting after the previous attempt to validate the session fails. This timer is used only when the Cisco Secure ACS sends a Accept-Reject message to the switch.

The default value of the hold timer is 180 seconds (3 minutes).

An EAPoUDP session might not be validated because the posture validation of the host fails, a session timer expires, or the switch or Cisco Secure ACS receives invalid messages. If the switch or authentication server continuously receives invalid messages, a malicious user might be trying to cause a denial-of-service attack.

Idle Timer

The idle timer controls how long the switch waits for an ARP packet from the postured host or a refreshed entry in the IP device tracking table to verify that the host is still connected. The idle timer works with a list of known hosts to track hosts that have initiated posture validation and the IP device tracking table.

The idle timer is reset when the switch receives an ARP packet or when an entry in the IP device tracking table is refreshed. If the idle timer expires, the switch ends the EAPoUDP session on the host, and the host is no longer validated.

The default value of the idle timer is calculated as the probe interval times the number of probe retries. By default, the idle timer default is 90 seconds which is the probe interval of 30 seconds times the number of probe retries of 3.

The switch maintains a list of known hosts to track hosts that have initiated posture validation. When the switch receives an ARP packet, it resets the aging timers for the list and the idle timer. If the aging time of the list expires, the switch sends an ARP probe to verify that the host is present. If the host is present, it sends a response to the switch. The switch updates the entry in the list of known hosts. The switch then resets the aging timers for the list and the idle timer. If the switch receives no response, the switch ends the session with the Cisco Secure ACS, and the host is no longer validated.

The switch uses the IP device tracking table to detect and manage hosts connected to the switch. The switch also uses ARP or DHCP snooping to detect hosts. By default, the IP device tracking feature is disabled on a switch.

You must enable the IP device tracking feature to use NAC IP validation.

When IP device tracking is enabled, and a host is detected, the switch adds an entry to the IP device tracking table that includes this information:

- IP and MAC address of the host
- Interface on which the switch detected the host
- Host state that is set to ACTIVE when the host is detected

If NAC Layer 2 or Layer 3 IP validation is enabled on an interface, adding an entry to the IP device tracking table initiates posture validation.

For the IP device tracking table, you can configure the number of times that the switch sends ARP probes for an entry before removing an entry from the table and you can also configure the number of seconds that the switch waits before resending the ARP probe. If the switch uses the default settings of the IP device tracking table, the switch sends ARP probes every 30 seconds for all the entries. When the host responds to the probe, the host state is refreshed and remains active. The switch can send up to three additional ARP probes at 30-second intervals if the switch does not get a response. After the maximum number of ARP probes are sent, the switch removes the host entry from the table. The switch ends the EAPoUDP session for the host if a session was set up.

Using the IP device tracking ensures that hosts are detected in a timely manner, despite the limitations of using DHCP. If a link goes down, the IP device tracking entries associated with the interface are not removed, and the state of entries is changed to inactive. The switch does not limit the number of active entries in the IP device tracking table but limits the number of inactive entries. When the table reaches the table size limit, the switch removes the inactive entries. If the table does not have inactive entries, the number of entries in the IP device tracking table increases. When a host becomes inactive, the switch ends the host session.

The table size limit is 2048.

After an interface link is restored, the switch sends ARP probes for the entry associated with the interface. The switch ages out entries for hosts that do not respond to ARP probes. The switch changes the state of hosts that respond to an active host and initiates posture validation.

Retransmission Timer

The retransmission timer controls the amount of time that the switch waits for a response from the client before resending a request during posture validation. Setting the timer value too low might cause unnecessary transmissions, and setting the timer value too high might cause poor response times.

The default value of the retransmission timer is 3 seconds.

Revalidation Timer

The revalidation timer controls the amount of time that a NAC policy is applied to a client that used EAPoUDP messages during posture validation. The timer starts after the initial posture validation is complete. The timer resets when the host is revalidated. The default value of the revalidation timer is 36000 seconds (10 hours).

You can specify the revalidation timer value on the switch by using the **eou timeout revalidation** *seconds* global configuration command. You can also specify the revalidation timer value on an interface by using the **eou timeout revalidation** *seconds* interface configuration command.

Note The revalidation timer can be configured locally on the switch or it can be downloaded from the control server.

The revalidation timer operation is based on Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute[29]) in the Access-Accept message from the Cisco Secure ACS running AAA. If the switch gets the Session-Timeout value, this value overrides the revalidation timer value on the switch.

If the revalidation timer expires, the switch action depends on one of these values of the Termination-Action attribute:

- If the value of the Termination-Action RADIUS attribute is the default, the session ends.
- If the switch gets a value for the Termination-Action attribute other than the default, the EAPoUDP session and the current access policy remain in effect during posture revalidation.
- If the value of the Termination-Action attribute is RADIUS, the switch revalidates the client.
- If the packet from the server does not include the Termination-Action attribute, the EAPoUDP session ends.

Status-Query Timer

The status-query timer controls the amount of time the switch waits before verifying that the previously validated client is present and that its posture has not changed. Only clients that were authenticated with EAPoUDP messages use this timer, which starts after the client is initially validated. The default value of the status-query timer is 300 seconds (5 minutes).

The timer resets when the host is reauthenticated. When the timer expires, the switch checks the host posture validation by sending a Status-Query message to the host. If the host sends a message to the switch that the posture has changed, the switch revalidates the posture of the host.

NAC and Switchover

When RPR mode redundancy is configured, a switchover causes the loss of all information about currently postured hosts. All sessions will be revalidated. Users will be unvalidated and may see service disruption.

With Release 12.2(33)SXH and later releases, if you are using SSO mode redundancy, you can enter the **ip admission ha** command to enable host session table synchronization to the standby supervisor engine. When the high availability feature is enabled, established postured hosts do not need to be revalidated when a switchover occurs, because they will not see any disruption from the switchover. Posture sessions that were not yet established prior to the SSO will need to be revalidated after the switchover.

Configuring NAC

The following sections describe how to configure NAC:

Default NAC Configuration

By default, NAC IP validation is disabled.

NAC IP Guidelines, Limitations, and Restrictions

- The NAC feature applies access controls only to IPv4 traffic. NAC does not restrict Layer 2 bridged traffic.
- IPv6 traffic does not trigger posture validation and NAC IP does not apply access policies to IPv6 traffic.
- Default ACLs must permit EAPoUDP traffic for NAC IP to function.
- DHCP traffic must be permitted in the interface default ACL and the host policy for DHCP snooping to function.
- If you want to forward HTTP and HTTPS requests from an endpoint device to a specific URL, you must enable the HTTP server feature. The url-redirect-acl AV pair should be defined as the URL ACL name. This ACL should contain a **deny tcp any** *remediation server address* **eq www** command followed by the permit ACEs for the HTTP traffic that is being redirected.

NAC Layer 2 IP Guidelines, Limitations, and Restrictions

When configuring NAC Layer 2 IP validation, follow these guidelines, limitations, and restrictions:

- You must configure Layer 3 routes from the switch to the host for the Layer 2 IP to operate correctly.
- Layer 2 IP is not allowed if the parent VLAN of the port has VACL capture configured.
- LAN Port IP (LPIP) ARP traffic redirected to the CPU cannot be spanned using the SPAN feature.
- NAC Layer 2 IP validation is not supported on trunk ports, tunnel ports, EtherChannel members, or routed ports. The Catalyst 6500 series switches support Layer 2 IP on EtherChannels.
- When NAC Layer 2 IP validation is enabled, you must configure an ACL on the Layer 2 port to which hosts are connected.
- NAC Layer 2 IP is not supported if the Layer 2 port is part of a private VLAN.
- NAC Layer 2 IP ARP traffic redirected to the CPU cannot be spanned using the SPAN feature.
- A denial-of-service attack might occur if the switch receives many ARP packets with different source IP addresses. To avoid this problem, you must configure the IP admission MLS rate-limiting feature using the **mls rate-limit layer2 ip-admission** command.
- If DAI is also enabled on the parent VLAN of the Layer 2 port, the IP admission rate limiting for ARP packets directed to the CPU is ineffective. In this situation, ARP inspection rate limiting is functional. ARP inspection rate limiting is performed in software and IP admission rate limiting is performed in hardware.
- When NAC Layer 2 IP and NAC Layer 2 IEEE 802.1x are enabled on the same access port, IEEE 802.1x authentication takes precedence. The posture of the host to which the port is connected might already have been validated, and the switch would have applied the access limitations based on IEEE 802.1x.
- DHCP snooping must be enabled if the switch wants to use DHCP lease grants to identify connected hosts. DHCP packets are permitted in DHCP environments in both the default interface and the downloaded host policy.
- If you want the end stations to send DNS requests before posture validation occurs, you must configure the named downloadable ACL on the Layer 2 port with ACEs permitting DNS packets.
- If NAC Layer 2 IP validation is configured on a Layer 2 port that belongs to a voice VLAN, the switch does not validate the posture of the IP phone. Make sure that the IP phone is on the exception list.
- If NAC Layer 2 IP validation is enabled, the NAC Layer 2 IP configuration takes precedence over VLAN ACLs and router ACLs that are configured on ingress interfaces. For example, when a VLAN ACL and a router ACL are configured, the operation applies the policies serially in the order of the LPIP policy to VLAN ACL to router ACL. The next policy is applied only when the traffic passes through the previous policy check. Any policy in the serial order denying the traffic causes the traffic to be denied. The downloaded LPIP host policy always overrides the default interface policy.
- If dynamic ARP inspection is enabled on the ingress VLAN, the switch initiates posture validation only after the ARP packets are validated.
- The traffic sent to the URL-redirect deny ACEs is forwarded in hardware without applying the default interface and the downloaded host policies. If this traffic (that is, the traffic matching the deny URL-redirect ACEs) requires filtering, you should define a VLAN ACL on the Layer 2 access VLAN. This configuration allows you to bypass the redirection of the HTTP traffic destined for the remediation servers.

NAC Layer 3 IP Guidelines, Limitations, and Restrictions

When configuring NAC Layer 3 IP validation, follow these guidelines, limitations, and restrictions:

- Only Release 12.2(33)SXH and rebuilds support NAC Layer 3 IP validation. Release 12.2(33)SXI and later releases do not support NAC Layer 3 IP validation.
- NAC Gateway feature is supported on Supervisor Engine 720 and Supervisor Engine 32-8GE.
- For ARP-based trigger, the GWIP detects host presence using the ARP probing mechanism.
- When you enable NAC Gateway IP validation on an interface, you must also configure a default Cisco IOS ACL on the interface.
- The traffic sent to the URL-redirect deny ACEs is forwarded in hardware without applying the default interface and the downloaded host policies. If this traffic (that is, the traffic matching the deny URL-redirect ACEs) requires filtering, you should define a Cisco IOS ACL on the interface. This configuration allows you to bypass the redirection of the HTTP traffic destined for the remediation servers.
- If IEEE 802.1x authentication in single-host mode and NAC Layer 2 IP validation are configured on a Layer 2 port, and IEEE 802.1x authentication of the connected hosts fails, the switch does not initiate posture validation when it receives DHCP or ARP packets from the host.

  If IEEE 802.1x authentication is configured on the port, the port cannot send or receive traffic other than EAPOL frames until the client is successfully authenticated.

Configuring NAC IP Validation

To configure NAC Layer 2 IP validation, beginning in privileged EXEC mode, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`ip admission name`** *`rule_name`* **`eapoudp`**	Creates and configures an IP NAC rule by specifying the rule name. To remove the IP NAC rule on the switch, use the **no ip admission name** *rule-name* **eapoudp** global configuration command.

Step 3	`Router(config)#` **`mls rate-limit layer 2 ip ip-admission`** *`pps`* `(`*`burst`*`)`
	`or`
	`Router(config)#` **`mls rate-limit unicast ip features`** *`pps`* `(`*`burst`*`)`
		For a Layer 2 port, enables the rate limiting of the IP admission traffic to the CPU.
		For a Layer 3 port, enables the rate limiting of the IP admission traffic to the CPU.

	Command	Purpose
Step 4	`Router(config)#` **access-list** *access_list_number* {**deny** | **permit**} *source* [*source_wildcard*] [**log**]	Defines an ACL by using a source address and wildcard. The *access_list_number* value is a decimal number from 1 to 99 or 1300 to 1999. Enter **deny** or **permit** to specify whether to deny or permit access if conditions are matched. The *source* value is the source address of the network or host from which the packet is being sent specified as follows: • The 32-bit quantity in dotted-decimal format. • The keyword **any** as an abbreviation for *source* and *source_wildcard* of 0.0.0.0 255.255.255.255. You do not need to enter a *source_wildcard*. • The keyword **host** as an abbreviation for source and source-wildcard of *source* 0.0.0.0. (Optional) The *source_wildcard* applies wildcard bits to the source. (Optional) Enter **log** to cause an informational logging message about the packet that matches the entry to be sent to the console.
Step 5	`Router(config)#` **interface** *interface_id*	Enters interface configuration mode.
Step 6	`Router(config)#` **ip access-group** {*access_list_number* | *name*} **in**	Controls access to the specified interface.
Step 7	`Router(config)#` **ip admission name** *rule_name*	Applies the specified IP NAC rule to the interface. To remove the IP NAC rule that was applied to a specific interface, use the **no ip admission name** *rule-name* interface configuration command.
Step 8	`Router(config)#` **exit**	Returns to global configuration mode.
Step 9	`Router(config)#` **aaa new-model**	Enables AAA.
Step 10	`Router(config)#` **aaa authentication eou default group radius**	Sets authentication methods for EAPoUDP. To remove the EAPoUDP authentication methods, use the **no aaa authentication eou default** global configuration command.
Step 11	`Router(config)#` **ip device tracking**	Enables the IP device tracking table. To disable the IP device tracking table, use the **no device tracking** global configuration command.

	Command	Purpose
Step 12	`Router(config)# ip device tracking probe {count count | interval interval}`	(Optional) Configures these parameters for the IP device tracking table: • **count** *count*—Sets the number of times that the switch sends the ARP probe. The range is from 1 to 5. The default is 3. • **interval** *interval*—Sets the number of seconds that the switch waits for a response before resending the ARP probe. The range is from 30 to 300 seconds. The default is 30 seconds.
Step 13	`Router(config)# radius-server host {hostname | ip_address} key string`	(Optional) Configures the RADIUS server parameters. For the *hostname* or *ip_address* value, specify the hostname or IP address of the remote RADIUS server. For the **key** *string* value, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server. **Note** Always configure the key as the last item in the **radius-server host** command syntax because leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon. If you want to use multiple RADIUS servers, reenter this command.
Step 14	`Router(config)# radius-server attribute 8 include-in-access-req`	If the switch is connected to nonresponsive hosts, configures the switch to send the Framed-IP-Address RADIUS attribute (Attribute[8]) in access-request or accounting-request packets.
Step 15	`Router(config)# radius-server vsa send authentication`	Configures the network access server to recognize and use vendor-specific attributes.
Step 16	`Router(config)# ip device tracking [probe {count count | interval interval}]`	(Optional) Configures these IP device tracking table parameters: • **probe count** *count*—Sets the number of times that the switch sends the ARP probe for an entry before removing an entry from the IP device tracking table. The range is from 1 to 5. The default is 3. • **probe interval** *interval*—Sets the number of seconds that the switch waits before resending the ARP probe. The range is from 30 to 300 seconds. The default is 30 seconds.
Step 17	`Router(config)# eou logging`	(Optional) Enables EAPoUDP system logging events.
Step 18	`Router(config)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 19	`Router# show ip admission {[cache] [configuration] [eapoudp]}`	Displays the NAC configuration or network admission cache entries.
Step 20	`Router# show ip device tracking {all	interface interface_id	ip ip_address	mac mac_address}`	Displays information about the entries in the IP device tracking table.
Step 21	`Router# show ip access lists interface interface`	Displays the downloaded host policies in the Cisco IOS software configuration.
Step 22	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

When configuring NAC IP validation, note the following guidelines:

- To remove the IP NAC rule on the switch, use the **no ip admission name** *rule_name* **eapoudp** global configuration command. To remove the IP NAC rule that was applied to a specific interface, use the **no ip admission** *admission_name* interface configuration command.
- To remove the EAPoUDP authentication methods, use the **no aaa authentication eou default** global configuration command. To configure the auth-proxy posture code to not obtain security associations from the AAA server, use the **no aaa authorization auth-proxy default** global configuration command.
- To disable the IP device tracking table and return the parameters for the table to the default values, use the **no device tracking** and the **no device tracking probe** {**count** | **interval**} global configuration commands.
- To configure the switch to not send the Framed-IP-Address attribute, use the **no radius-server attribute 8 include-in-access-req** global configuration command.
- To disable the logging of EAPoUDP system events, use the **no eou logging** global configuration command.
- To clear all NAC client device entries on the switch or on the specified interface, use the **clear eou** privileged EXEC command. To clear entries in the IP device tracking table, use the **clear ip device tracking** privileged EXEC command.
- If IEEE 802.1x authentication in single-host mode and NAC Layer 2 IP validation are configured on a Layer 2 port and IEEE 802.1x authentication of the connected hosts fails, the switch does not initiate posture validation when it receives DHCP or ARP packets from the host.

  If IEEE 802.1x authentication is configured on the port, the port cannot send or receive traffic other than EAPOL frames until the client is successfully authenticated.

This example shows how to configure NAC Layer 2 IP validation on a switch interface:

```
Router# configure terminal
Router(config)# ip admission name nac eapoudp
Router(config)# access-list 5 permit any any
Router(config)# interface gigabitethernet 2/0/1
Router(config-if)# ip access-group 5 in
Router(config-if)# ip admission nac
Router(config-if)# exit
Router(config)# aaa new-model
Router(config)# aaa authentication eou default group radius
Router(config)# radius-server host admin key rad123
Router(config)# radius-server vsa send authentication
Router(config)# ip device tracking probe count 2
Router(config)# eou logging
Router(config)# end
```

Configuring EAPoUDP

To configure the EAPoUDP, beginning in privileged EXEC mode, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **eou allow** {**clientless** | **ip-station-id**} **eou default** **eou logging** **eou max-retry** *number* **eou port** *port_number* **eou ratelimit** *number* **eou timeout** {**aaa** *seconds* | **hold-period** *seconds* | **retransmit** *seconds* | **revalidation** *seconds* | **status-query** *seconds*} **eou revalidate**	Specifies EAPoUDP values. For more information about the **allow**, **default**, **logging**, **max-retry**, **port**, **rate-limit**, **revalidate**, and **timeout** keywords, see the command reference for this release and the *Network Admission Control* feature module.
Step 3	Router(config)# **interface** *interface_id*	Enters interface configuration mode.
Step 4	Router(config)# **eou default** **eou max-retry** *number* **eou timeout** {**aaa** *seconds* | **hold-period** *seconds* | **retransmit** *seconds* | **revalidation** *seconds* | **status-query** *seconds*} **eou revalidate**	Enables and configures the EAPoUDP association for the specified interface. For more information about the **default**, **max-retry**, **revalidate**, and **timeout** keywords, see the command reference for this release and the *Network Admission Control* feature module.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	Router# **show eou** {**all** | **authentication** {**clientless** | **eap** | **static**} | **interface** *interface_id* | **ip** *ip_address* | **mac** *mac_address* | **posturetoken** *name*}	Displays information about the EAPoUDP configuration or session cache entries.
Step 7	Router# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return to the global default EAPoUDP values, use the **no** forms of the **eou** global configuration commands. To disable the EAPoUDP associations, use the **no** forms of the **eou** interface configuration commands.

Configuring Identity Profiles and Policies

To configure the identity profile and policy beginning in privileged EXEC mode, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **identity policy** *policy_name*	Creates an identity policy, and enters identity-policy configuration mode.
Step 3	Router(config-identity-policy)# **access-group** *access_group*	Defines network access attributes for the identity policy.
Step 4	Router(config)# **identity profile eapoudp**	Creates an identity profile, and enters identity-profile configuration mode.

	Command	Purpose
Step 5	`Router(config-identity-prof)#` **`device`** `{`**`authorize`** `|` **`not-authorize`**`}` `{`**`ip-address`** *`ip_address`* `|` **`mac-address`** *`mac_address`* `|` **`type cisco ip phone`**`}` `[`**`policy`** *`policy_name`*`]`	Authorizes the specified IP device, and applies the specified policy to the device.
Step 6	`Router(config)#` **`exit`**	Exits from identity-profile configuration mode, and returns to global configuration mode.
Step 7	`Router#` **`end`**	Returns to privileged EXEC mode.
Step 8	`Router#` **`show running-config`**	Verifies your entries.
Step 9	`Router#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To remove the identity policy from the switch, use the **no identity-policy** *policy_name* global configuration command. To remove the identity profile, use the **no identity profile eapoudp** global configuration command. To not authorize the specified IP device and remove the specified policy from the device, use the **no device** {**authorize** | **not-authorize**} {**ip-address** *ip_address* | **mac-address** *mac_address* | **type cisco ip phone**} [**policy** *policy_name*] interface configuration command.

This example shows how to configure the identity profile and policy:

```
Router# configure terminal
Router(config)# identity policy policy1
Router(config-identity-policy)# access-group group1
Router(config)# identity profile eapoudp
Router(config-identity-prof)# device authorize ip address 10.10.142.25 policy policy1
Router(config-identity-prof)# exit
Router(config)# end
```

Configuring NAC High Availability

To configure IP admission high availability, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`ip admission ha`**	Enables IP admission high availability.
Step 3	`Router(config)#` **`ip admission ha update`** *`update_interval`*	Defines how often the active supervisor engine sends synchronization updates to the standby. The interval has a range of 30 to 60 seconds.
Step 4	`Router#` **`show ip admission ha stats`**	Displays the statistics related to session synchronization.
Step 5	`Router#` **`clear ip admission ha stats`**	(Optional) Clears the statistics related to session synchronization.

Note You cannot enable the IP admission high availability feature if there are active Webauth or Posture sessions.

To disable IP admission high availability from the switch, use the **no ip admission ha** configuration command.

This example shows how to configure IP admission high availability:

```
Router# configure terminal
Router(config)# ip admission ha
Router(config)# ip admission ha update 50
Router(config)# end
Router(config)# clear ip admission ha stats
Router(config-identity-prof)# show ip admission ha stats
```

Configuring a NAC AAA Down Policy

To configure NAC AAA down policy, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)#` **`ip admission name`** *`rule-name`* **`eapoudp event timeout aaa policy identity`** *`identity_policy_name`*	Creates a NAC rule and associates an identity policy to be applied to sessions, when the AAA server is unreachable. To remove the rule on the switch, use the **no ip admission name** *rule-name* **eapoudp event timeout aaa policy identity** global configuration command.
Step 3	`Router(config)#` **`access-list`** *`access-list-number`* {**`deny`** | **`permit`**} *`source`* [*`source-wildcard`*] [**`log`**]	Defines the default port ACL by using a source address and wildcard. The *access-list-number* is a decimal number from 1 to 99 or 1300 to 1999. Enter **deny** or **permit** to specify whether to deny or permit access if conditions are matched. The *source* is the source address of the network or host from which the packet is being sent specified as follows: • The 32-bit quantity in dotted-decimal format. • The keyword **any** as an abbreviation for *source* and *source-wildcard* value of 0.0.0.0 255.255.255.255. You do not need to enter a *source-wildcard* value. • The keyword **host** as an abbreviation for source and source-wildcard of *source* 0.0.0.0. (Optional) Applies the *source-wildcard* wildcard bits to the source. (Optional) Enters **log** to cause an informational logging message about the packet that matches the entry to be sent to the console.
Step 4	`Router(config-if)#` **`interface`** *`interface-id`*	Enters interface configuration mode.
Step 5	`Router(config-if)#` **`ip access-group`** {*`access-list-number`* | *`name`*} **`in`**	Controls access to the specified interface.
Step 6	`Router(config-if)#` **`ip admission`** *`rule-name`*	Applies the specified IP NAC rule to the interface. To remove the IP NAC rule that was applied to a specific interface, use the **no ip admission** *rule-name* interface configuration command.
Step 7	`Router(config)#` **`exit`**	Returns to global configuration mode.
Step 8	`Router(config)#` **`aaa new-model`**	Enables AAA.

	Command	Purpose
Step 9	`Router(config)#` **`aaa authentication eou default group radius`**	Sets authentication methods for EAPoUDP. To remove the EAPoUDP authentication methods, use the **no aaa authentication eou default** global configuration command.
Step 10	`Router(config)#` **`aaa authorization network default local`**	Sets the authorization method to local. To remove the authorization method, use **no aaa authorization network default local** command.
Step 11	`Router(config)#` **`ip device tracking`**	Enables the IP device tracking table. To disable the IP device tracking table, use the **no ip device tracking** global configuration commands.
Step 12	`Router(config)#` **`ip device tracking`** `[`**`probe`** `{`**`count`** *`count`* `|` **`interval`** *`interval`*`}]`	(Optional) Configures these parameters for the IP device tracking table: • **count** *count*—Sets the number of times that the switch sends the ARP probe. The range is from 1 to 5. The default is 3. • **interval** *interval*—Sets the number of seconds that the switch waits for a response before resending the ARP probe. The range is from 30 to 300 seconds. The default is 30 seconds.
Step 13	`Router(config)#` **`radius-server host`** `{`*`hostname`* `|` *`ip-address`*`}` **`test username`** *`username`* **`idle-time 1 key`** *`string`*	(Optional) Configures the RADIUS server parameters. For the *hostname* or *ip-address,* specify the hostname or IP address of the remote RADIUS server. For the **key** *string* value, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server. **Note** Always configure the key as the last item in the **radius-server host** command syntax because leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon. The **test username** value parameter is used for configuring the dummy username that tests whether the AAA server is active or not. The **idle-time** parameter is used to set how often the server should be tested to determine its operational status. If there is no traffic to the RADIUS server, the NAD sends dummy radius packets to the RADIUS server based on the idle-time. If you want to use multiple RADIUS servers, reenter this command.
Step 14	`Router(config)#` **`radius-server attribute 8 include-in-access-req`**	(Optional) Configures the switch to send the Framed-IP-Address RADIUS attribute (Attribute[8]) in access-request or accounting-request packets if the switch is connected to nonresponsive hosts. To configure the switch to not send the Framed-IP-Address attribute, use the **no radius-server attribute 8 include-in-access-req** global configuration command.
Step 15	`Router(config)#` **`radius-server vsa send authentication`**	Configures the network access server to recognize and use vendor-specific attributes.

	Command	Purpose
Step 16	Router(config)# **radius-server dead-criteria** {**tries** | **time}** *value*	Forces one or both of the criteria (used to mark a RADIUS server as dead) to be the indicated constant.
Step 17	Router(config)# **eou logging**	(Optional) Enables EAPoUDP system logging events. To disable the logging of EAPoUDP system events, use the **no eou logging** global configuration command.
Step 18	Router(config)# **end**	Returns to privileged EXEC mode.
Step 19	Router# **show ip admission** {[**cache**] [**configuration**] [**eapoudp**]}	Displays the NAC configuration or network admission cache entries.
Step 20	Router# **show ip device tracking** {**all** | **interface** *interface-id* | **ip** *ip-address* | **mac** *mac-address*}	Displays information about the entries in the IP device tracking table.
Step 21	Router(# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The following example illustrates how to apply a AAA down policy:

```
Router# config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip admission name AAA_DOWN eapoudp event timeout aaa policy identity
global_policy
Router(config)# aaa new-model
Router(config)# aaa authorization network default local
Router(config)# aaa authentication eou default group radius
Router(config)# identity policy global_policy
Router(config-identity-policy)# ac
Router(config-identity-policy)# access-group global_acl
Router(config)# ip access-list extended global_acl
Router(config-ext-nacl)# permit ip any any
Router(config-ext-nacl)# exit
Router(config)# radius-server host 40.0.0.4 test username administrator idle-time 1 key
cisco
Router(config)# radius-server dead-criteria tries 3
Router(config)# radius-server vsa send authentication
Router(config)# radius-server attribute 8 include-in-access-req
Router(config)# int fastEthernet 2/13
Router(config-if)# ip admission AAA_DOWN
Router(config-if)# exit
Router# show ip admission configuration

Show running output
-------------------
aaa new-model
aaa authentication eou default group radius
aaa authorization network default local

ip admission name AAA_DOWN eapoudp event timeout aaa policy identity global_policy

identity policy global_policy
 access-group global_acl

interface FastEthernet2/13
 switchport
 switchport access vlan 222
 switchport mode access
 no ip address
 ip access-group 115 in
 ip admission AAA_DOWN
```

```
!
ip access-list extended global_acl
 permit ip any any

radius-server dead-criteria tries 3
radius-server attribute 8 include-in-access-req
radius-server host 40.0.0.4 auth-port 1645 acct-port 1646 test username administrator
idle-time 1 key cisco
radius-server vsa send authentication

Router# show ip admission configuration
Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication global init state time is 2 minutes
Auth-proxy name AAA_DOWN
   eapoudp list not specified auth-cache-time 60 minutes
   Identity policy name global_policy for AAA fail policy
```

Monitoring and Maintaining NAC

You can perform the tasks in these sections to monitor and maintain NAC:

- Clearing Table Entries, page 59-23
- Displaying NAC Information, page 59-23

Clearing Table Entries

To clear client entries in the EAPoUDP session table, use the **clear eou** privileged EXEC command. After the entries are removed, they are created only after the switch receives an ARP packet from the host or after it creates a DHCP binding entry for the host.

To clear entries in the IP device tracking table on the switch, use the **clear ip device tracking** privileged EXEC command.

Displaying NAC Information

To display NAC information, perform one of the following tasks:

Command	Purpose
Router# **show dot1x** [**all** | **interface** *interface_id* | **statistics interface** *interface_id*]	Displays IEEE 802.1x statistics, administrative status, and operational status.
Router# **show eou** {**all** | **authentication** {**clientless** | **eap** | **static**} | **interface** *interface_id* | **ip** *ip_address* | **mac** *mac_address* | **posturetoken** *name*}	Displays information about the EAPoUDP configuration or session cache entries.
Router# **show ip admission** {[**cache**] [**configuration**] [**eapoudp**]}	Displays the NAC configuration or network admission cache entries.
Router# **show ip device tracking** {**all** | **interface** *interface_id* | **ip** *ip_address* | **mac** *mac_address*}	Displays information about the entries in the IP device tracking table.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring IEEE 802.1X Port-Based Authentication

This chapter describes how to configure IEEE 802.1X port-based authentication in Cisco IOS Release 12.2SX to prevent unauthorized devices (clients) from gaining access to the network.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

This chapter consists of these sections:

Understanding 802.1X Port-Based Authentication

The IEEE 802.1X standard defines a client and server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a switch port and assigns the port to a VLAN before making available any services offered by the switch or the LAN.

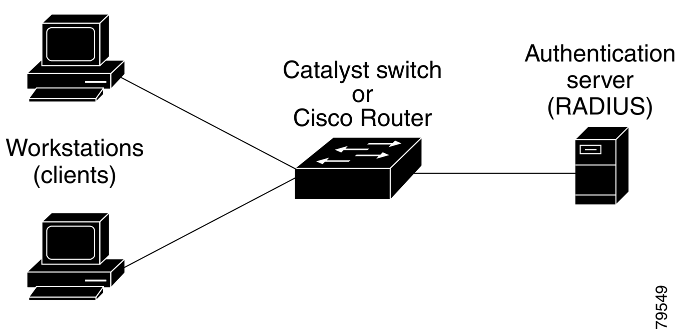
Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

These sections describe the role of 802.1X port-based authentication as a part of a system of authentication, authorization, and accounting (AAA):

Understanding 802.1X Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles as shown in Figure 60-1.

Figure 60-1 802.1X Device Roles



The specific roles shown in Figure 60-1 are as follows:

- *Client*—The device (workstation) that requests access to the LAN and switch services and responds to requests from the switch.The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating system. (The client is the *supplicant* in the IEEE 802.1X specification.)

Note To resolve Windows XP network connectivity and 802.1X port-based authentication issues, read the Microsoft Knowledge Base article at this URL: http://support.microsoft.com/kb/q303597/

- *Authentication server*—Performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch whether or not the client is authorized to access the LAN and switch services. Because the switch acts as the proxy, the authentication service is transparent to the client. The Remote Authentication Dial-In User Service (RADIUS) security system with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server (ACS), version 3.0. RADIUS uses a client-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.
- *Switch* (also called the *authenticator* and *back-end authenticator*)—With Release 12.2(33)SXH and later releases, controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client. The switch includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

  When the switch receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is reencapsulated in the RADIUS format. The EAP frames are not modified or examined during encapsulation, and the authentication server must support EAP within the native frame format. When the switch receives frames from the authentication server, the server's frame header is removed, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client.

Understanding the Port-based Authentication Process

When 802.1X port-based authentication is enabled, these events occur:

- If the client supports 802.1X-compliant client software and the client's identity is valid, the 802.1X authentication succeeds and the switch grants the client access to the network.
- If 802.1X authentication times out while waiting for an EAPOL message exchange, the switch can use a fallback authentication method, such as MAC authentication bypass (MAB) or web-based authentication (webauth), if either or both are enabled:
  - If MAC authentication bypass is enabled, the switch relays the client's MAC address to the AAA server for authorization. If the client's MAC address is valid, the authorization succeeds and the switch grants the client access to the network.
  - If web-based authentication is enabled, the switch sends an HTTP login page to the client. The switch relays the client's username and password to the AAA server for authorization. If the login succeeds, the switch grants the client access to the network.

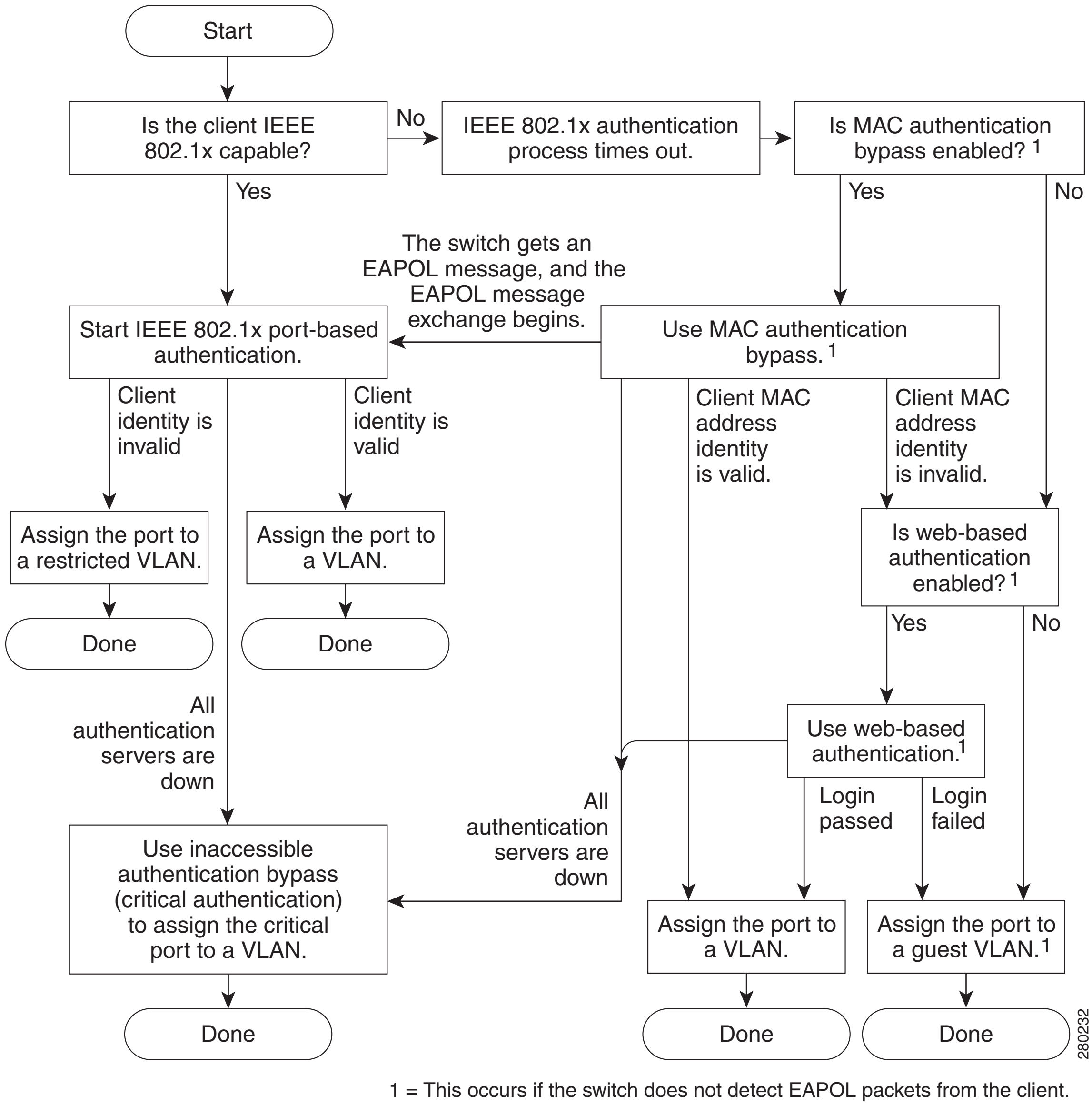
Note The default order for authentication methods is 802.1X, and then MAB, then web-based authentication. You can change the order, and you can disable any of these methods.

- If fallback authentication methods are not enabled or are not successful, and if a guest VLAN is configured, the switch assigns the client to a guest VLAN that provides limited services.
- If the switch receives an invalid identity from an 802.1X-capable client and a restricted VLAN is specified, the switch can assign the client to a restricted VLAN that provides limited services.
- If the RADIUS authentication server is unavailable (down) and inaccessible authentication bypass is enabled, the switch grants the client access to the network by putting the port in the critical-authentication state in the user-specified critical VLAN. Release 12.2(33)SXJ1 and later releases support configuration of critical voice and data VLANs.

Note Inaccessible authentication bypass is also referred to as critical authentication or the AAA fail policy.

Figure 60-2 shows the authentication process.

Figure 60-2 Authentication Flowchart



The switch reauthenticates a client when one of these situations occurs:

- Periodic reauthentication is enabled, and the reauthentication timer expires.

  You can configure the reauthentication timer to use a switch-specific value or to be based on values from the RADIUS server.

  After 802.1X authentication using a RADIUS server is configured, the switch uses timers based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]).

  The Session-Timeout RADIUS attribute (Attribute[27]) specifies the time after which reauthentication occurs.

The Termination-Action RADIUS attribute (Attribute [29]) specifies the action to take during reauthentication. The actions are Initialize and ReAuthenticate. When the Initialize action is set (the attribute value is DEFAULT), the 802.1X session ends, and connectivity is lost during reauthentication. When the ReAuthenticate action is set (the attribute value is RADIUS-Request), the session is not affected during reauthentication.

- You manually reauthenticate the client by entering the **dot1x re-authenticate interface** *type slot/port* privileged EXEC command (Cisco IOS Release 12.2(33)SXH and earlier releases).

Authentication Initiation and Message Exchange

The switch or the client can initiate authentication. If you enable authentication on a port by using the **dot1x pae authenticator** and **authentication port-control auto** interface configuration commands (**dot1x port-control auto** command in Cisco IOS Release 12.2(33)SXH and earlier releases), the switch must initiate authentication when it determines that the port link state transitions from down to up. The switch then sends an EAP-request/identity frame to the client to request its identity (typically, the switch sends an initial identity/request frame followed by one or more requests for authentication information). When the client receives the frame, it responds with an EAP-response/identity frame.

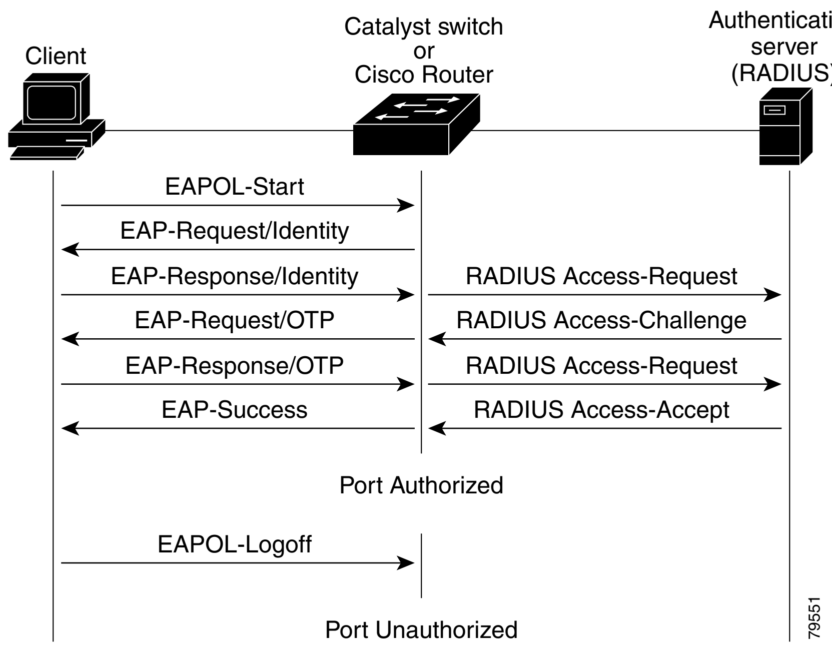
If the client does not receive an EAP-request/identity frame from the switch during bootup, the client can initiate authentication by sending an EAPOL-start frame, which prompts the switch to request the client's identity.

Note If 802.1X is not enabled or supported on the network access device, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client transmits frames as if the port is in the authorized state. A port in the authorized state effectively means that the client has been successfully authenticated. For more information, see the "Ports in Authorized and Unauthorized States" section on page 60-8.

When the client supplies its identity, the switch begins its role as the intermediary, passing EAP frames between the client and the authentication server until authentication succeeds or fails. If the authentication succeeds, the port becomes authorized. If the authentication fails, authentication can be retried, the port might be assigned to a VLAN that provides limited services, or network access is not granted. For more information, see the "Ports in Authorized and Unauthorized States" section on page 60-8.

The specific exchange of EAP frames depends on the authentication method being used. Figure 60-3 shows a message exchange initiated by the client using the One-Time-Password (OTP) authentication method with a RADIUS server.

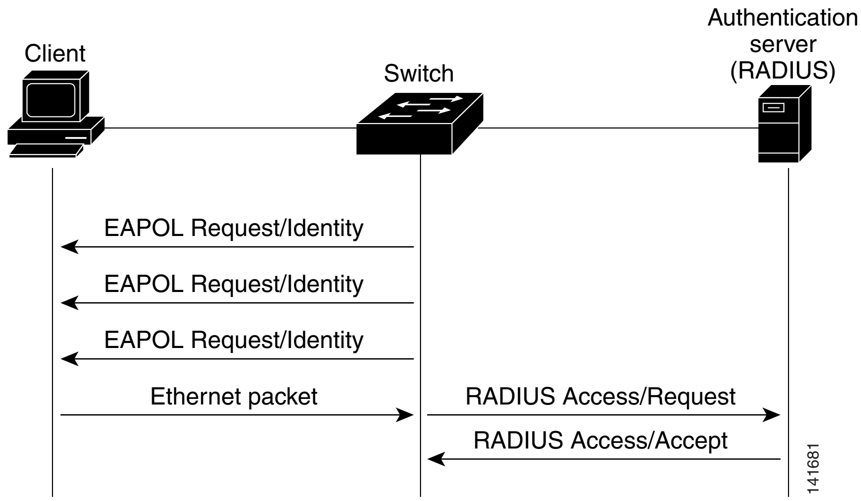
Figure 60-3 Message Exchange



If 802.1X authentication times out while waiting for an EAPOL message exchange, and MAC authentication bypass is enabled, the switch can authorize the client when the switch detects an Ethernet packet from the client. The switch uses the MAC address of the client as its identity and includes this information in the RADIUS-access/request frame that is sent to the RADIUS server. After the server sends the switch the RADIUS-access/accept frame (authorization is successful), the port becomes authorized. If MAB authorization fails and a guest VLAN is specified, the switch assigns the port to the guest VLAN. If the switch detects an EAPOL packet while waiting for an Ethernet packet, the switch stops the MAC authentication bypass process and starts 802.1X authentication.

Figure 60-4 shows the message exchange during MAC authentication bypass.

Figure 60-4 Message Exchange During MAC Authentication Bypass



Ports in Authorized and Unauthorized States

The switch port state determines whether or not the client is granted access to the network. The port starts in the *unauthorized* state. While in this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a client is successfully authenticated, the port transitions to the *authorized* state, allowing all traffic for the client to flow normally.

If a client that does not support 802.1X authentication connects to an unauthorized 802.1X port, the switch requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

You control the port authorization state by using the **authentication port-control** interface configuration command (**dot1x port-control auto** command in Cisco IOS Release 12.2(33)SXH and earlier releases) and these keywords:

- **force-authorized**—Disables 802.1X port-based authentication and causes the port to transition to the authorized state without any authentication exchange required. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This is the default setting.
- **force-unauthorized—**Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The switch cannot provide authentication services to the client through the interface.
- **auto**—Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received. The switch requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each client attempting to access the network is uniquely identified by the switch by using the client's MAC address.

If the client is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated client are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the switch can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and network access is not granted.

When a client logs off, it sends an EAPOL-logoff message, causing the switch port to transition to the unauthorized state.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

802.1X Host Modes

The 802.1X port's host mode determines whether more than one client can be authenticated on the port and how authentication will be enforced. You can configure an 802.1X port to use any of the four host modes described in the following sections. In addition, each mode may be modified to allow pre-authentication open access.

- Single-Host Mode, page 60-9
- Multiple-Hosts Mode, page 60-9
- Multidomain Authentication Mode, page 60-10
- Multiauthentication Mode, page 60-10
- Pre-Authentication Open Access, page 60-11

Single-Host Mode

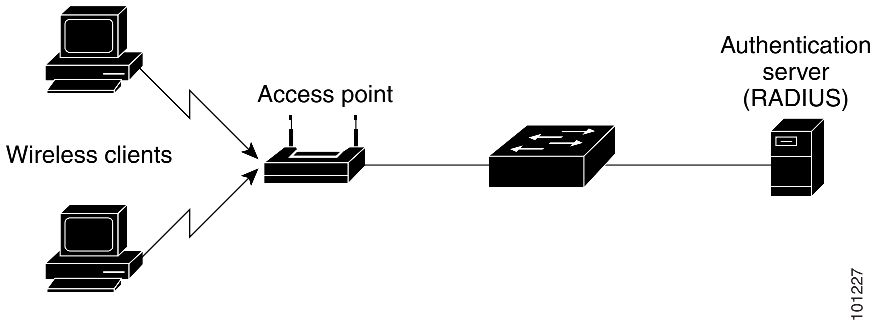
In single-host mode (see Figure 60-1 on page 60-2), only one client can be connected to the 802.1X-enabled port. The switch detects the client by sending an EAPOL frame when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

Multiple-Hosts Mode

In multiple-hosts mode, you can attach multiple hosts to a single 802.1X-enabled port. Figure 60-5 shows 802.1X port-based authentication in a wireless LAN. In this mode, only one of the attached clients must be authorized for all clients to be granted network access. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the switch denies network access to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and it also acts as a client to the switch.

With the multiple-hosts mode enabled, you can use 802.1X authentication to authenticate the port and you can use port security to manage network access for all MAC addresses, including the client's MAC address.

Figure 60-5 Multiple Host Mode Example

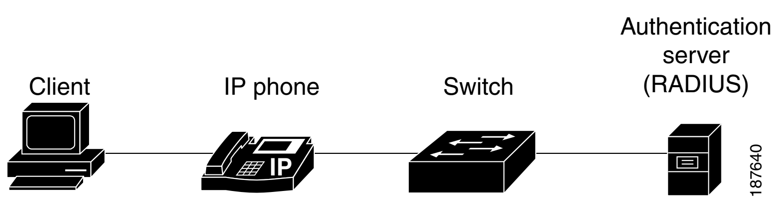


Multidomain Authentication Mode

Supported in Cisco IOS Release 12.2(33)SXI and later releases, multidomain authentication (MDA) mode allows an IP phone (Cisco or third-party) and a single host behind the IP phone to authenticate independently, using 802.1X, MAC authentication bypass (MAB), or (for the host only) web-based authentication. In this application, multidomain refers to two domains, data and voice, and only two MAC addresses are allowed per port. The switch can place the host in the data VLAN and the IP phone in the voice VLAN, though they appear on the same switch port. The data VLAN assignment can be obtained from the vendor-specific attributes (VSAs) received from the authentication, authorization, and accounting (AAA) server during authentication.

Figure 60-6 shows a typical MDA application with a single host behind an IP phone connected to the 802.1X-enabled port. Because the client is not directly connected to the switch, the switch cannot detect a loss of port link if the client is disconnected. To prevent the possibility of another device using the established authentication of the disconnected client, later Cisco IP phones send a Cisco Discovery Protocol (CDP) host presence type length value (TLV) to notify the switch of changes in the attached client's port link state.

Figure 60-6 Multidomain Authentication Mode Example



Multiauthentication Mode

Available in Cisco IOS Release 12.2(33)SXI and later releases, multiauthentication (multiauth) mode allows one 802.1X/MAB client on the voice VLAN and multiple authenticated 802.1X/MAB/webauth clients on the data VLAN. When a hub or access point is connected to an 802.1X port (as shown in Figure 60-5), multiauth mode provides enhanced security over the multiple-hosts mode by requiring authentication of each connected client. For non-802.1X devices, MAB or web-based authentication can be used as the fallback method for individual host authentications, which allows different hosts to be authenticated through different methods on a single port.

Multiauth also supports MDA functionality on the voice VLAN by assigning authenticated devices to either a data or voice VLAN depending on the data that the VSAs received from the authentication server.

Release 12.2(33)SXJ and later releases support the assignment of a RADIUS server-supplied VLAN in multiauth mode, by using the existing commands and when these conditions occur:

- The host is the first host authorized on the port, and the RADIUS server supplies VLAN information.
- Subsequent hosts are authorized with a VLAN that matches the operational VLAN.
- A host is authorized on the port with no VLAN assignment, and subsequent hosts either have no VLAN assignment, or their VLAN information matches the operational VLAN.
- The first host authorized on the port has a group VLAN assignment, and subsequent hosts either have no VLAN assignment, or their group VLAN matches the group VLAN on the port. Subsequent hosts must use the same VLAN from the VLAN group as the first host. If a VLAN list is used, all hosts are subject to the conditions specified in the VLAN list.
- After a VLAN is assigned to a host on the port, subsequent hosts must have matching VLAN information or be denied access to the port.
- The behavior of the critical-auth VLAN is not changed for multiauth mode. When a host tries to authenticate and the server is not reachable, all authorized hosts are reinitialized in the configured VLAN.

Note

- Only one voice VLAN is supported on a multiauth port.
- You cannot configure a guest VLAN or an auth-fail VLAN in multiauth mode.

Pre-Authentication Open Access

With Cisco IOS Release 12.2(33)SXI and later releases, any of the four host modes may be additionally configured to allow a device to gain network access before authentication. This pre-authentication open access is useful in an application such as the Pre-boot eXecution Environment (PXE), where a device must access the network to download a bootable image containing an authentication client.

Pre-authentication open access is enabled by entering the **authentication open** command after host mode configuration, and acts as an extension to the configured host mode. For example, if pre-authentication open access is enabled with single-host mode, then the port will allow only one MAC address. When pre-authentication open access is enabled, initial traffic on the port is restricted only by whatever other access restriction, independent of 802.1X, is configured on the port. If no access restriction other than 802.1X is configured on the port, then a client device will have full access on the configured VLAN.

Understanding 802.1X Authentication with DHCP Snooping

With Cisco IOS Release 12.2(33)SXH and later releases, when the Dynamic Host Configuration Protocol (DHCP) snooping option-82 with data insertion feature is enabled, the switch can insert a client’s 802.1X authenticated user identity information into the DHCP discovery process, allowing the DHCP server to assign IP addresses from different IP address pools to different classes of end users. This feature allows you to secure the IP addresses given to the end users for accounting purposes and to allow services based on Layer 3 criteria.

After a successful 802.1X authentication between a supplicant and the RADIUS server, the switch puts the port in the forwarding state and stores the attributes that it receives from the RADIUS server. While performing DHCP snooping, the switch acts as a DHCP relay agent, receiving DHCP messages and regenerating those messages for transmission on another interface. When a client, after 802.1X authentication, sends a DHCP discovery message, the switch receives the packet. The switch adds to the packet a RADIUS attributes suboption section containing the stored RADIUS attributes of the client. The switch then submits the discovery broadcast again. The DHCP server receives the modified DHCP discovery packet and can, if configured to do so, use the authenticated user identity information when creating the IP address lease. The mapping of user-to-IP address can be on a one-to-one, one-to-many, or many-to-many basis. The one-to-many mapping allows the same user to authenticate through the 802.1X hosts on multiple ports.

The switch will automatically insert the authenticated user identity information when 802.1X authentication and DHCP snooping option-82 with data insertion features are enabled. To configure DHCP snooping option-82 with data insertion, see the "DHCP Snooping Option-82 Data Insertion" section on page 54-3.

For information about the data inserted in the RADIUS attributes suboption, see RFC 4014, "Remote Authentication Dial-In User Service (RADIUS) Attributes Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Information Option."

Understanding 802.1X Accounting

The IEEE 802.1X standard defines how users are authorized and authenticated for network access but does not keep track of network usage. IEEE 802.1X accounting is disabled by default. With Release 12.2(33)SXH and later releases, you can enable 802.1X accounting to monitor the following activities on 802.1X-enabled ports:

- User successfully authenticates.
- User logs off.
- Link-down occurs.
- Reauthentication successfully occurs.
- Reauthentication fails.

The switch does not log IEEE 802.1X accounting information. Instead, it sends this information to the RADIUS server, which must be configured to log accounting messages.

The information sent to the RADIUS server is represented in the form of 802.1X Accounting Attribute-Value (AV) pairs. These AV pairs provide data for different applications. (For example, a billing application might require information that is in the Acct-Input-Octets or the Acct-Output-Octets attributes of a RADIUS packet.)

AV pairs are automatically sent by a switch that is configured for 802.1X accounting. Three types of RADIUS accounting packets are sent by a switch:

- START–Sent when a new user session starts.
- INTERIM–Sent during an existing session for updates.
- STOP–Sent when a session terminates.

Table 60-1 lists the AV pairs and indicates when they are sent are sent by the switch.

Table 60-1 Accounting AV Pairs

Attribute Number	AV Pair Name	START	INTERIM	STOP
Attribute[1]	User-Name	Always	Always	Always
Attribute[4]	NAS-IP-Address	Always	Always	Always
Attribute[5]	NAS-Port	Always	Always	Always
Attribute[8]	Framed-IP-Address	Never	Sometimes[1]	Sometimes[1]
Attribute[25]	Class	Always	Always	Always
Attribute[26]	Vendor-Specific[2]	—	—	—
Attribute[30]	Called-Station-ID	Always	Always	Always
Attribute[31]	Calling-Station-ID	Always	Always	Always
Attribute[40]	Acct-Status-Type	Always	Always	Always
Attribute[41]	Acct-Delay-Time	Always	Always	Always
Attribute[42]	Acct-Input-Octets	Never	Never	Always
Attribute[43]	Acct-Output-Octets	Never	Never	Always
Attribute[44]	Acct-Session-ID	Always	Always	Always
Attribute[45]	Acct-Authentic	Always	Always	Always
Attribute[46]	Acct-Session-Time	Never	Never	Always
Attribute[49]	Acct-Terminate-Cause	Never	Never	Always
Attribute[61]	NAS-Port-Type	Always	Always	Always

1. The Framed-IP-Address AV pair is sent only if a valid DHCP binding exists for the host in the DHCP snooping bindings table.
2. Vendor-specific attributes (VSAs) are used by other 802.1X features.

You can view the AV pairs that are being sent by the switch by entering the **debug radius accounting** privileged EXEC command. For more information about this command, see the *Cisco IOS Debug Command Reference, Release 12.2* at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/debug/command/reference/122debug.html

For more information about AV pairs, see RFC 3580, “IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines.”

Understanding 802.1X Authentication with VLAN Assignment

After successful 802.1X authentication of a port, the RADIUS server sends the VLAN assignment to configure the port. The RADIUS server database maintains the username-to-VLAN mappings, assigning the VLAN based on the username of the client connected to the port. You can use this feature to limit network access for certain users.

When configured on the switch and the RADIUS server, 802.1X authentication with VLAN assignment has these characteristics:

- If 802.1X authentication is enabled on a port, and if all information from the RADIUS server is valid, the port is placed in the RADIUS server-assigned VLAN after authentication.
- If the multiple-hosts mode is enabled on an 802.1X port, all hosts on the port are placed in the same RADIUS server-assigned VLAN as the first authenticated host.
- If the multiauth mode is enabled on an 802.1X port, the VLAN assignment will be ignored.
- If no VLAN number is supplied by the RADIUS server, the port is configured in its access VLAN after successful authentication. An access VLAN is a VLAN assigned to an access port. All packets sent from or received on this port belong to this VLAN.
- If 802.1X authentication is enabled but the VLAN information from the RADIUS server is not valid, the port returns to the unauthorized state and remains in the configured access VLAN. This prevents ports from appearing unexpectedly in an inappropriate VLAN because of a configuration error.

  Configuration errors could include specifying a VLAN for a routed port, a malformed VLAN ID, a nonexistent or internal (routed port) VLAN ID, or an attempted assignment to a voice VLAN ID.
- If 802.1X authentication is disabled on the port, the port is returned to the configured access VLAN.

When the port is in the force-authorized, force-unauthorized, unauthorized, or shutdown state, the port is put into the configured access VLAN.

If an 802.1X port is authenticated and put in the RADIUS server-assigned VLAN, any change to the port access VLAN configuration does not take effect.

The 802.1X authentication with VLAN assignment feature is not supported on trunk ports, dynamic ports, or with dynamic-access port assignment through a VLAN Membership Policy Server (VMPS).

To configure VLAN assignment, perform this task:

Step 1 Enable AAA authorization by using the **network** keyword to allow interface configuration from the RADIUS server.

Step 2 Enable 802.1X authentication.

Step 3 The VLAN assignment feature is automatically enabled when you configure 802.1X authentication on an access port.

Step 4 Assign vendor-specific tunnel attributes in the RADIUS server. The RADIUS server must return these attributes to the switch:

- [64] Tunnel-Type = VLAN
- [65] Tunnel-Medium-Type = 802
- [81] Tunnel-Private-Group-ID = VLAN name or VLAN ID

Attribute [64] must contain the value *VLAN* (type 13). Attribute [65] must contain the value *802* (type 6). Attribute [81] specifies the *VLAN name* or *VLAN ID* assigned to the 802.1X-authenticated user.

Understanding Multiple VLANs and VLAN User Distribution with VLAN Assignment

In Cisco IOS Release 12.2(33)SXI1 and later releases, the RADIUS-supplied VLAN assignment can provide load balancing by distributing 802.1X-authenticated users among multiple VLANs.

In earlier releases, the RADIUS server can supply a single VLAN name or ID for the assignment of an authenticating user. In Cisco IOS Release 12.2(33)SXI1 and later releases, the RADIUS server can supply multiple VLAN names and IDs or the name of a VLAN group that contains multiple VLANs. Use either of the following two methods to load balance the users between the different VLANs:

- Configure the RADIUS server to send more than one VLAN ID or VLAN name as part of the response to the authenticating user. The 802.1X VLAN user group feature tracks the users in a particular VLAN and achieves load balancing by placing newly authenticated users in the least populated VLAN of the RADIUS-supplied VLAN IDs.

  Perform the steps shown in the "Understanding 802.1X Authentication with VLAN Assignment" section on page 60-13 with the following exception:

  Attribute [81] Tunnel-Private-Group-ID specifies multiple VLAN names or VLAN IDs

- Define a VLAN group that contains multiple VLANs. Configure the RADIUS server to supply the VLAN group name instead of a VLAN ID as part of the response to the authenticating user. If the supplied VLAN group name is found among the VLAN group names that you have defined, the newly authenticated user is placed in the least populated VLAN within the VLAN group.

  Perform the steps shown in the "Understanding 802.1X Authentication with VLAN Assignment" section on page 60-13 with the following exception:

  Attribute [81] Tunnel-Private-Group-ID specifies a defined VLAN group name

For more information, see the "Configuring VLAN User Distribution" section on page 60-49.

Understanding 802.1X Authentication with Guest VLAN

With Release 12.2(33)SXH and later releases, you can configure a guest VLAN for each 802.1X port on the switch to provide limited services to non-802.1X-compliant clients, such as for downloading the 802.1X client software. These clients might be upgrading their system for 802.1X authentication, and some hosts, such as Windows 98 systems, might not be 802.1X-capable.

When you enable a guest VLAN on an 802.1X port, the switch assigns clients to a guest VLAN when the switch does not receive a response to its EAP request/identity frame or when EAPOL packets are not sent by the client and no fallback authentication methods are enabled.

In addition, the switch maintains the EAPOL packet history. If an EAPOL packet is detected on the interface during the lifetime of the link, the switch determines that the device connected to that interface is an 802.1X-capable supplicant, and the interface will not change to the guest VLAN state. The EAPOL packet history is cleared if the interface link status goes down.

Use the **dot1x guest-vlan supplicant** global configuration command to allow an interface to change to the guest VLAN state regardless of the EAPOL packet history. That is, a host that is not 802.1X-capable will be assigned to the guest VLAN even if a previous host on that interface was 802.1X-capable.

Note If an EAPOL packet is detected after the interface has changed to the guest VLAN, the interface reverts to an unauthorized state, and 802.1X authentication restarts.

Any number of 802.1X-incapable clients are allowed access when the port is moved to the guest VLAN. If an 802.1X-capable client joins the same port on which the guest VLAN is configured, the port is put into the unauthorized state in the user-configured access VLAN, and authentication is restarted.

When operating as an 802.1X guest VLAN, a port functions in multiple-hosts mode regardless of the configured host mode of the port.

You can configure any active VLAN except an RSPAN VLAN, a private primary PVLAN, or a voice VLAN as an 802.1X guest VLAN. The guest VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

The switch supports MAC authentication bypass in Release 12.2(33)SXH and later releases. When MAC authentication bypass is enabled on an 802.1X port, the switch can authorize clients based on the client MAC address when 802.1X authentication times out while waiting for an EAPOL message exchange. After detecting a client on an 802.1X port, the switch waits for an Ethernet packet from the client. The switch sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the switch grants the client access to the network. If authorization fails, the switch assigns the port to the guest VLAN if one is specified.

For more information, see the "Understanding 802.1X Authentication with MAC Authentication Bypass" section on page 60-23 and the "Configuring a Guest VLAN" section on page 60-49.

Understanding 802.1X Authentication with Restricted VLAN

You can configure a restricted VLAN (also referred to as an *authentication failed VLAN*) for each 802.1X port on a switch to provide limited services to clients that failed authentication and cannot access the guest VLAN. These clients are 802.1X-compliant and cannot access another VLAN because they fail the authentication process. A restricted VLAN allows users without valid credentials in an authentication server (typically, visitors to an enterprise) to access a limited set of services. The administrator can control the services available to the restricted VLAN.

Note You can configure a VLAN to be both the guest VLAN and the restricted VLAN if you want to provide the same services to both types of users.

Without this feature, the client attempts and fails authentication indefinitely, and the port remains in the spanning-tree blocking state. With this feature, you can configure the port to be in the restricted VLAN after a specified number of authentication attempts.

The authenticator counts the failed authentication attempts for the client. The failed attempt count increments when the RADIUS server replies with either an Access-Reject EAP failure or an empty response without an EAP packet. When this count exceeds the configured maximum number of authentication attempts, the port moves to the restricted VLAN, the failed attempt counter resets, and subsequent EAPOL-start messages from the failed client are ignored.

Users who fail authentication remain in the restricted VLAN until the next switch-initiated reauthentication attempt. A port in the restricted VLAN tries to reauthenticate at configured intervals (the default is 60 seconds). If reauthentication fails, the port remains in the restricted VLAN. If reauthentication is successful, the port moves either to the configured VLAN or to a VLAN sent by the RADIUS server. You can disable reauthentication. If you do this, the only way to restart the authentication process is for the port to receive a link down or EAP logoff event. We recommend that you keep reauthentication enabled if a client might connect through a hub. When a client disconnects from the hub, the port might not receive the link down or EAP logoff event.

When operating as an 802.1X restricted VLAN, a port functions in single-host mode regardless of the configured host mode of the port. Only the client that failed authentication is allowed access on the port. An exception is that a port configured in MDA mode can still authenticate a voice supplicant from the restricted VLAN.

You can configure any active VLAN except an RSPAN VLAN or a voice VLAN as an 802.1X restricted VLAN. The restricted VLAN feature is not supported on routed or trunk ports; it is supported only on access ports.

This feature works with port security. As soon as the port is authorized, a MAC address is provided to port security. If port security does not permit the MAC address or if the maximum secure address count is reached, the port becomes unauthorized and error disabled.

Other port security features such as dynamic ARP Inspection, DHCP snooping, and IP source guard can be configured independently on a restricted VLAN.

For more information, see the "Configuring a Restricted VLAN" section on page 60-51.

Understanding 802.1X Authentication with Inaccessible Authentication Bypass

With Release 12.2(33)SXH and later releases, when the switch cannot reach the configured RADIUS servers and hosts cannot be authenticated, you can configure the switch to allow network access to the hosts connected to critical ports. A critical port is enabled for the inaccessible authentication bypass feature, also referred to as critical authentication or the AAA fail policy.

When this feature is enabled, the switch checks the status of the configured RADIUS servers whenever the switch tries to authenticate a host connected to a critical port. If a server is available, the switch can authenticate the host. However, if all the RADIUS servers are unavailable, the switch grants network access to the host and puts the port in the critical-authentication state, which is a special case of the authentication state.

The behavior of the inaccessible authentication bypass feature depends on the authorization state of the port:

- If the port is unauthorized when a host connected to a critical port tries to authenticate and all servers are unavailable, the switch puts the port in the critical-authentication state in the user-specified critical VLAN.
- If the port is already authorized and reauthentication occurs, the switch puts the critical port in the critical-authentication state in the current VLAN, which might be the one previously assigned by the RADIUS server.
- If the RADIUS server becomes unavailable during an authentication exchange, the current exchanges times out, and the switch puts the critical port in the critical-authentication state during the next authentication attempt.

When a RADIUS server that can authenticate the host is available, all critical ports in the critical-authentication state are automatically reauthenticated.

Inaccessible authentication bypass interacts with these features:

- Guest VLAN—Inaccessible authentication bypass is compatible with guest VLAN. When a guest VLAN is enabled on 8021.x port, the features interact as follows:
  - If at least one RADIUS server is available, the switch assigns a client to a guest VLAN when the switch does not receive a response to its EAP request/identity frame or when EAPOL packets are not sent by the client.
  - If all the RADIUS servers are not available and the client is connected to a critical port, the switch authenticates the client and puts the critical port in the critical-authentication state in the user-specified critical VLAN.
  - If all the RADIUS servers are not available and the client is not connected to a critical port, the switch might not assign clients to the guest VLAN if one is configured.
  - If all the RADIUS servers are not available and if a client is connected to a critical port and was previously assigned to a guest VLAN, the switch keeps the port in the guest VLAN.
- Restricted VLAN—If the port is already authorized in a restricted VLAN and the RADIUS servers are unavailable, the switch puts the critical port in the critical-authentication state in the restricted VLAN.
- 802.1X accounting—Accounting is not affected if the RADIUS servers are unavailable.
- Private VLAN—You can configure inaccessible authentication bypass on a private VLAN host port. The access VLAN must be a secondary private VLAN.
- Voice VLAN—Inaccessible authentication bypass is compatible with voice VLAN, but the RADIUS-configured or user-specified access VLAN and the voice VLAN must be different.
- Remote Switched Port Analyzer (RSPAN)—Do not configure an RSPAN VLAN as the RADIUS-configured or user-specified access VLAN for inaccessible authentication bypass.

Understanding 802.1X Authentication with Voice VLAN Ports

A Multi-VLAN Access Port (MVAP) is a port that belong to two VLANs. A voice VLAN port is an MVAP that allows separating a port's voice traffic and data traffic on different VLANs. A voice VLAN port is associated with two VLAN identifiers:

- Voice VLAN identifier (VVID) to carry voice traffic to and from the IP phone. The VVID is used to configure the IP phone connected to the port.
- Port VLAN identifier (PVID) to carry the data traffic to and from the workstation connected to the switch through the IP phone. The PVID is the native VLAN of the port.

In releases earlier than Release 12.2(33)SXH, a switch in single-host mode accepted traffic from a single host, and voice traffic was not allowed. In multiple-hosts mode, the switch did not accept voice traffic until the client was authenticated on the primary VLAN, which makes the IP phone inoperable until the user logged in.

With Release 12.2(33)SXH and later releases, the IP phone uses the VVID for its voice traffic, regardless of the authorization state of the port. This allows the phone to work independently of 802.1X authentication.

In single-host mode, only the IP phone is allowed on the voice VLAN. In multiple-hosts mode, additional clients can send traffic on the voice VLAN after a supplicant is authenticated on the PVID. When multiple-hosts mode is enabled, the supplicant authentication affects both the PVID and the VVID.

In order to recognize an IP phone, the switch will allow CDP traffic on a port regardless of the authorization state of the port. A voice VLAN port becomes active when there is a link, and the device MAC address appears after the first CDP message from the IP phone. Cisco IP phones do not relay CDP messages from other devices. As a result, if several IP phones are connected in series, the switch recognizes only the one directly connected to it. When 802.1X authentication is enabled on a voice VLAN port, the switch drops packets from unrecognized IP phones more than one hop away.

When 802.1X authentication is enabled on a port, you cannot configure a port VLAN that is equal to a voice VLAN.

Note If you enable 802.1X authentication on an access port on which a voice VLAN is configured and to which a Cisco IP phone is connected, the Cisco IP phone loses connectivity to the switch for up to 30 seconds.

For voice VLAN configuration information, see Chapter 15, “Configuring Cisco IP Phone Support.”

Understanding 802.1X Authentication Critical Voice VLAN Support

With normal network connectivity, when an IP phone successfully authenticates on a port, the authentication server puts the phone into the voice domain. If the authentication server becomes unreachable, IP phones cannot authenticate. In multidomain authentication (MDA) mode or multiauthentication mode, you can configure the critical voice VLAN support feature to put phone traffic into the configured voice VLAN of the port (see the “Enabling Critical Voice VLAN Support” section on page 60-56).

Understanding 802.1X Authentication with Port Security

With Release 12.2(33)SXH and later releases, you can configure an 802.1X port with port security in either single-host or multiple-hosts mode. (You also must configure port security on the port by using the **switchport port-security** interface configuration command.) When you enable port security and 802.1X authentication on a port, 802.1X authentication authenticates the port, and port security manages network access for all MAC addresses, including that of the client. You can then limit the number or group of clients that can access the network through an 802.1X port.

These are some examples of the interaction between 802.1X authentication and port security on the switch:

- When a client is authenticated, and the port security table is not full, the client MAC address is added to the port security list of secure hosts. The port then proceeds to come up normally.

  When a client is authenticated and manually configured for port security, it is guaranteed an entry in the secure host table.

  A security violation occurs if the client is authenticated, but the port security table is full. This can happen if the maximum number of secure hosts has been statically configured or if the client ages out of the secure host table. If the client address is aged, its place in the secure host table can be taken by another host.

  If a security violation is caused by any host, the port becomes error-disabled and immediately shuts down.

  The port security violation modes determine the action for security violations. For more information, see the “Configuring the Port Security Violation Mode on a Port” section on page 62-6.

- When you manually remove an 802.1X client address from the port security table by using the **no switchport port-security mac-address** *mac-address* interface configuration command, you should reauthenticate the 802.1X client by using the **dot1x re-authenticate interface** *type slot/port* privileged EXEC command.
- When an 802.1X client logs off, the port changes to an unauthenticated state, and all dynamic entries in the secure host table are cleared, including the entry for the client. Normal authentication then takes place.
- If the port is administratively shut down, the port becomes unauthenticated, and all dynamic entries are removed from the secure host table.
- Port security and a voice VLAN can be configured simultaneously on an 802.1X port that is in either single-host or multiple-hosts mode. Port security applies to both the voice VLAN identifier (VVID) and the port VLAN identifier (PVID).

For more information about enabling port security on your switch, see the "Configuring Port Security" section on page 62-5.

Understanding 802.1X Authentication with ACL Assignments and Redirect URLs

With Cisco IOS Release 12.2(33)SXI and later releases, per-host policies such as ACLs and redirect URLs can be downloaded to the switch from the authentication server (AS) in a RADIUS Access-Accept packet at the end of an 802.1X, MAB, or web-based authentication exchange.

Per-host policies are activated during authentication as follows:

- Downloadable ACLs (DACLs) are defined in the Cisco Secure ACS and downloaded from the ACS to the switch using VSAs.
- Filter-ID ACLs are defined on the switch, and only the ACL name is downloaded from the AS to the switch using the RADIUS Filter-ID attribute. Filter-ID ACLs are supported in Cisco IOS Release 12.2(33)SXI2 and later releases.
- A redirection URL and an ACL name are downloaded from the ACS to the switch using VSAs. The redirection ACL is defined on the switch.

For information about configuring per-host policies, see the "Configuring the Switch for DACLs or Redirect URLs" section on page 60-60.

Downloadable ACLs Using the Cisco Secure ACS

Following a successful host authentication, the Cisco Secure ACS can use a VSA to download an ACL to the switch. The switch combines the DACL with the default ACL on the port to which the host has connected. Because the DACL definition resides on the authentication server, this feature allows for centralized policy management.

Two methods are provided in the Cisco Secure ACS for configuring DACLs:

- Downloadable IP ACL

  Downloading of the DACL is enabled by selecting Assign IP ACL in the ACS configuration, and the DACL is defined in the Downloadable IP ACL Content menu of the ACS. There is no restriction on the size of the DACL.
- Per-user ACL

In Cisco IOS Release 12.2(33)SXI2 and later releases, the ACS can use the CiscoSecure-Defined-ACL [009\001 cisco-av-pair] VSAs to deliver the DACL. Because the entire DACL is delivered in a single RADIUS packet, the maximum size is limited by the 4096-byte maximum size for a RADIUS packet. The DACL must be defined on the ACS using the following format:

protocol:inacl#*sequence_number*=*ace*

as shown in this example:

```
ip:inacl#10=permit ip any 67.2.2.0 0.0.0.255
```

These guidelines apply when using DACLs:

- The source address for all ACEs must be defined as ANY.
- When the 802.1X host mode of the port is MDA or multiauth, the DACL will be modified to use the authenticated host's IP address as the source address. When the host mode is either single-host or multiple-host, the source address will be configured as ANY, and the downloaded ACLs or redirects will apply to all devices on the port.
- If no DACLs are provided during the authentication of a host, the static default ACL configured on the port will be applied to the host. On a voice VLAN port, only the static default ACL of the port will be applied to the phone.

Filter-ID ACLs

In Cisco IOS Release 12.2(33)SXI2 and later releases, following a successful host authentication, the authentication server can use the RADIUS Filter-ID attribute (Attribute[11]) rather than a VSA to deliver only the name of an extended ACL to the switch in the following format:

acl_name.in

The suffix ".in" indicates that the ACL should be applied in the inbound direction.

In this method, the ACL must be already defined on the switch. The switch matches the Filter-ID attribute value to a locally configured ACL that has the same name or number as the Filter-ID (for example, Filter-ID=101.in will match the extended numbered ACL 101, and Filter-ID= guest.in will match the extended named ACL "guest"). The specified ACL is then applied to the port. Because the ACL definition resides on the switch, this feature allows for local variation in a policy.

These guidelines apply when using Filter-ID ACLs:

- The guidelines for using DACLs also apply to Filter-ID ACLs.
- The Filter-ID attribute may be a number (100 to 199, or 2000 to 2699) or a name.

Redirect URLs

Following a successful host authentication, the Cisco Secure ACS can use a VSA to download information to the switch for intercepting and redirecting HTTP or HTTPS requests from the authenticated host. The ACS downloads a redirection ACL and URL. When an HTTP or HTTPS request from the host matches the downloaded ACL, the host's web browser is redirected to the downloaded redirection URL.

The ACS uses these cisco-av-pair VSAs to configure the redirection:

- url-redirect-acl

This AV pair contains the name or number of an ACL that specifies the HTTP or HTTPS traffic to be redirected. The ACL must be defined on the switch, and the source address must be defined as ANY. Traffic that matches a permit entry in the redirect ACL will be redirected.

- url-redirect

  This AV pair contains the HTTP or HTTPS URL to which the web browser will be redirected.

Static Sharing of ACLs

When a number of interfaces have the same PACL and VLAN-based features, you can use the **mls acl tcam share-acl** global configuration command to enable the static sharing feature. With static sharing, only one copy of the PACL and inherited VLAN-based feature ACLs is stored in the TCAM for all ports using the same ACL set, freeing TCAM space for more ACLs. With static sharing enabled, the switch will automatically evaluate all configured or enabled interfaces for static sharing when any of these events occur:

- When the **mls acl tcam share-acl** command is entered.
- When an interface is configured.
- When a state change occurs on an interface.

When enabling static sharing, consider the following guidelines and restrictions:

- Static sharing is not supported for interfaces enabled with IPv6.
- Static sharing is not supported with PFC3A-based supervisor engines or earlier, or in systems running in PFC3A mode or lower.
- Static sharing is supported only on switch ports in access mode with NAC or 802.1X DACL features configured.
- Static sharing is not supported on switch ports enabled with QOS, with the exception of VLAN-based QoS.
- When 802.1X is used with DACL, we recommend entering the **platform hardware acl dynamic setup static** command to avoid triggering a static sharing evaluation when the port is dynamically configured by the authentication server response. The static sharing evaluation may adversely affect the port/host linkup time.
- 802.1X interfaces with fallback authentication as active cannot form a static sharing group with interfaces on which fallback is not enabled or is not active.

Understanding 802.1X Authentication with Port Descriptors

With Release 12.2(33)SXI and later releases, you can associate descriptive text with an 802.1X client's authentication information by configuring the Cisco vendor-specific attribute (VSA) **aaa:supplicant-name** on the RADIUS server. During a successful 802.1X authentication of the client on the port, the switch will receive the descriptive information from the RADIUS server as part of the

Access-Accept packet and will display the information when the **show interface users** command is entered for the port. If the port is in a mode supporting multiple authenticated hosts, identity information for all the authenticated hosts will be displayed with the port description.

Understanding 802.1X Authentication with MAC Authentication Bypass

With Release 12.2(33)SXH and later releases, you can configure the switch to authorize clients based on the client MAC address (see Figure 60-4 on page 60-8) by using the MAC authentication bypass feature. For example, you can enable this feature on 802.1X ports connected to devices such as printers.

If 802.1X authentication times out while waiting for an EAPOL response from the client, the switch tries to authorize the client by using MAC authentication bypass.

When the MAC authentication bypass feature is enabled on an 802.1X port, the switch uses the MAC address as the client identity. The authentication server has a database of client MAC addresses that are allowed network access. After detecting a client on an 802.1X port, the switch waits for an Ethernet packet from the client. The switch sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the switch grants the client access to the network. If authorization fails, the switch assigns the port to the guest VLAN if one is configured.

If an EAPOL packet is detected on the interface during the lifetime of the link, the switch determines that the device connected to that interface is an 802.1X-capable supplicant and uses 802.1X authentication (not MAC authentication bypass) to authorize the interface. EAPOL history is cleared if the interface link status goes down.

If the switch already authorized a port by using MAC authentication bypass and detects an 802.1X supplicant, the switch does not unauthorize the client connected to the port. When reauthentication occurs, the switch uses 802.1X authentication as the preferred reauthentication process if the previous session ended because the Termination-Action RADIUS attribute value is DEFAULT.

Clients that were authorized with MAC authentication bypass can be reauthenticated. The reauthentication process is the same as that for clients that were authenticated with 802.1X. During reauthentication, the port remains in the previously assigned VLAN. If reauthentication is successful, the switch keeps the port in the same VLAN. If reauthentication fails, the switch assigns the port to the guest VLAN, if one is configured.

If reauthentication is based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]) and if the Termination-Action RADIUS attribute (Attribute [29]) action is Initialize, (the attribute value is DEFAULT), the MAC authentication bypass session ends, and connectivity is lost during reauthentication. If MAC authentication bypass is enabled and the 802.1X authentication times out, the switch uses the MAC authentication bypass feature to initiate reauthorization. For more information about these AV pairs, see RFC 3580, "IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines."

MAC authentication bypass interacts with the features:

- 802.1X authentication—You can enable MAC authentication bypass only if 802.1X authentication is enabled on the port.
- Guest VLAN—If a client has an invalid MAC address identity, the switch assigns the client to a guest VLAN if one is configured.
- Restricted VLAN—This feature is not supported when the client connected to an 802.lx port is authenticated with MAC authentication bypass.
- Port security—See the "Understanding 802.1X Authentication with Port Security" section on page 60-19.

- Voice VLAN—See the "Understanding 802.1X Authentication with Voice VLAN Ports" section on page 60-18.
- VLAN Membership Policy Server (VMPS)— 802.1X and VMPS are mutually exclusive.
- Private VLAN—You can assign a client to a private VLAN.
- Network admission control (NAC) Layer 2 IP validation—This feature takes effect after an 802.1X port is authenticated with MAC authentication bypass, including hosts in the exception list.

Understanding Network Admission Control Layer 2 IEEE 802.1X Validation

Cisco IOS Release 12.2(33)SXH and later releases support Network Admission Control (NAC) Layer 2 IEEE 802.1X validation, which checks the antivirus condition or *posture* of endpoint systems or clients before granting the devices network access. NAC Layer 2 IEEE 802.1X validation performs policy enforcement by assigning the authenticated port into a specified VLAN, which provides segmentation and quarantine of poorly postured hosts at Layer 2.

Configuring NAC Layer 2 IEEE 802.1X validation is similar to configuring 802.1X port-based authentication except that you must configure a posture token on the RADIUS server. You can view the NAC posture token, which shows the posture of the client, by using the **show dot1x** privileged EXEC command. For information about configuring NAC Layer 2 IEEE 802.1X validation, see the "Configuring NAC Layer 2 IEEE 802.1X Validation" section on page 60-58.

For more information about NAC, see the *Network Admission Control Software Configuration Guide*.

NAC Agentless Audit Support

With Cisco IOS Release 12.2(33)SXI and later releases, MAB support is added for the Cisco NAC Audit Architecture, which uses an external audit server to check the antivirus posture of clients that do not run a Cisco Trust Agent (CTA) and cannot respond to NAC queries. To audit and report an agentless client's antivirus posture, the NAC audit server must possess the client's IP address and a unique session identifier for the client's connection to the switch. To support the NAC audit architecture for agentless clients, the switch must snoop the client's IP address, create and assign a unique session identifier for the agentless client, and pass this information to the RADIUS server for sharing with the NAC audit server.

Because MAB operates at Layer 2, the MAB authenticator does not normally know the IP address of the supplicant, and the supplicant might not have an IP address when it first contacts the authenticator. A supplicant that requires a DHCP-assigned IP address must be allowed access to a DHCP server before authentication. You must enable ARP and DHCP snooping on the switch to allow the MAB authenticator to learn the IP address of the supplicant. To allow the IP address and unique session identifier information to be shared with the NAC audit server, you must enable the sending of certain RADIUS attributes. See the "Configuring NAC Agentless Audit Support" section on page 60-59.

The client IP address and unique session identifier are shared in RADIUS Access-Requests and Access-Accepts using the following RADIUS *cisco-av-pair* vendor-specific attributes (VSAs):

- Cisco-AVPair="identity-request=*ip-address*"

  ip-address is the client IP address obtained by the switch through ARP or DHCP snooping.

- Cisco-AVPair="audit-session-id=*audit session id string*"

  audit session id string is a UTF-8 encoding of a unique 96-bit identifier derived by the switch from the network access server (NAS) IP address, a session count, and the session start timestamp.

Understanding RADIUS Change of Authorization

With Cisco IOS Release 12.2(33)SXI4 and later releases, the switch can accept and execute unsolicited Change of Authorization (CoA) messages from the authentication server (AS). CoA is an extension to the RADIUS protocol that allows the AS to make dynamic and unsolicited changes to the authorization information of an active session hosted by a network access device, such as a switch. For more information about CoA, see RFC 5176.

The Catalyst 6500 series switch supports per-session and per-policy CoA commands relating to 802.1X, MAB, and web-based authentication sessions.

Per-Session CoA

Using per-session CoA commands, the AS can cause the switch to terminate a session or to force a reauthentication of the session. To terminate a session, the AS can instruct the switch to perform one of the following actions:

- End the session—The AS sends a CoA Disconnect-Request (see RFC 5176), causing the switch to delete all state information about the session.
- Shut down the port—The AS sends the following VSA to force an administrative shutdown of the port:

  Cisco-AVPair="subscriber:command=*disable-host-port*"
- Bounce the port—The AS sends the following VSA to force the switch link to be taken down, then up again:

  Cisco-AVPair="subscriber:command=*bounce-host-port*"

By default, the switch accepts and executes per-session CoA commands, but you can configure the switch to ignore CoA shutdown or bounce commands directed at specific ports.

The AS sends the following VSA to force a reauthentication of the session:

Cisco-AVPair="subscriber:command=*re-authenticate*"

Per-Policy CoA

Using per-policy CoA commands, the AS can instruct the switch to update the contents of a DACL or a Filter-ID ACL, and apply the updated policy information to all sessions that currently have the affected ACL applied.

Understanding 802.1X Authentication with Wake-on-LAN

With Release 12.2(33)SXH and later releases, the 802.1X authentication with wake-on-LAN (WoL) feature allows dormant PCs to be powered up when the switch receives a specific Ethernet frame, known as the *magic packet*. You can use this feature in environments where administrators need to connect to systems that have been powered down.

When a host that uses WoL is attached through an 802.1X port and the host powers off, the 802.1X port becomes unauthorized. The port can only receive and send EAPOL packets, and WoL magic packets cannot reach the host. When the PC is powered off, it is not authorized, and the switch port is not opened.

When the switch uses 802.1X authentication with WoL, the switch forwards traffic to unauthorized 802.1X ports, including magic packets. While the port is unauthorized, the switch continues to block ingress traffic other than EAPOL packets. The host can receive packets but cannot send packets to other devices in the network.

Note If PortFast is not enabled on the port, the port is forced to the bidirectional state.

When you configure a port as unidirectional by using the **authentication control-direction in** interface configuration command (**dot1x control-direction in** command in Cisco IOS Release 12.2(33)SXH and earlier releases), the port changes to the spanning-tree forwarding state. The port can send packets to the host but cannot receive packets from the host.

When you configure a port as bidirectional by using the **authentication control-direction both** interface configuration command (**dot1x control-direction both** command in Cisco IOS Release 12.2(33)SXH and earlier releases), the port is access-controlled in both directions. The port does not receive packets from or send packets to the host.

Understanding MAC Move

Release 12.2(33)SXI4 and later releases support the Mac move feature. When a MAC address is authenticated on one switch port, that address is not allowed on another authentication manager-enabled port of the switch. If the switch detects that same MAC address on another authentication manager-enabled port, the address is not allowed.

There are situations where a MAC address might need to move from one port to another on the same switch. For example, when there is another device (for example a hub or an IP phone) between an authenticated host and a switch port, you might want to disconnect the host from the device and connect it directly to another port on the same switch.

You can globally enable MAC move so the device is reauthenticated on the new port. When a host moves to a second port, the session on the first port is deleted, and the host is reauthenticated on the new port.

When a MAC address moves from one port to another, the switch terminates the authenticated session on the original port and initiates a new authentication sequence on the new port. Port security behavior remains the same when you configure MAC move.

Note

- MAC move is supported in all host modes. (The authenticated host can move to any port on the switch, no matter which host mode is enabled on the that port.)
- MAC move is supported with port security.
- The MAC move feature applies to both voice and data hosts.
- In open authentication mode, a MAC address is immediately moved from the original port to the new port, with no requirement for authorization on the new port.

For more information see the “Enabling MAC Move” section on page 60-71.

Understanding MAC Replace

Release 12.2(33)SXI4 and later releases support the Mac replace feature. The MAC replace feature can be configured to address the violation that occurs when a host attempts to connect to a port where another host was previously authenticated.

Note

- The Mac replace feature is not supported on ports in multiauth mode, because violations are not triggered in that mode.
- The Mac replace feature is not supported on ports in multiple host mode, because in that mode, only the first host requires authentication.

If you configure the **authentication violation** interface configuration command with the **replace** keyword, the authentication process on a port in multi-domain mode is:

- A new MAC address is received on a port with an existing authenticated MAC address.
- The authentication manager replaces the MAC address of the current data host on the port with the new MAC address.
- The authentication manager initiates the authentication process for the new MAC address.
- If the authentication manager determines that the new host is a voice host, the original voice host is removed.

If a port is in open authentication mode, any new MAC address is immediately added to the MAC address table.

For more information see the "Enabling MAC Replace" section on page 60-71.

Understanding 802.1x Supplicant and Authenticator Switches with Network Edge Access Topology (NEAT)

Release 12.2(33)SXJ and later releases support the Network Edge Access Topology (NEAT) feature. NEAT extends identity to areas outside the wiring closet (such as conference rooms). This allows any type of device to authenticate on the port.

- 802.1x switch supplicant: You can configure a switch to act as a supplicant to another switch by using the 802.1x supplicant feature. This configuration is helpful in a scenario, where, for example, a switch is outside a wiring closet and is connected to an upstream switch through a trunk port. A switch configured with the 802.1x switch supplicant feature authenticates with the upstream switch for secure connectivity.

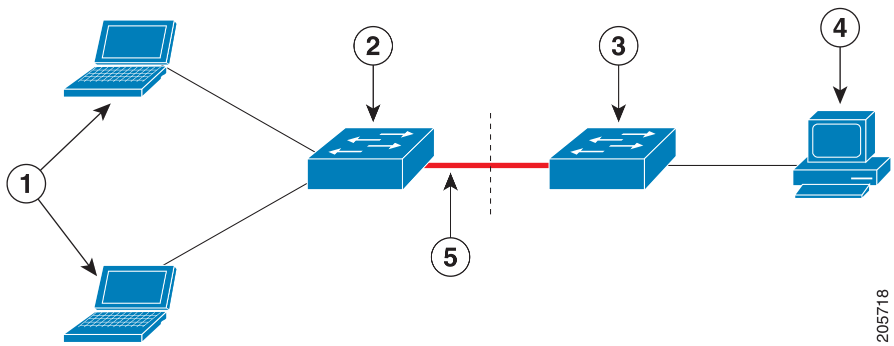
  Once the supplicant switch authenticates successfully the port mode changes from access to trunk.
- If the access VLAN is configured on the authenticator switch, it becomes the native VLAN for the trunk port after successful authentication.

You can enable MDA or multiauth mode on the authenticator switch interface that connects to one more supplicant switches. On the authenticator switch interface, multihost mode is not supported and in MDA mode voice client is not supported.

Use the **dot1x supplicant force-multicast** global configuration command on the supplicant switch for Network Edge Access Topology (NEAT) to work in all host modes.

- Host Authorization: Ensures that only traffic from authorized hosts (connecting to the switch with supplicant) is allowed on the network. The switches use Client Information Signalling Protocol (CISP) to send the MAC addresses connecting to the supplicant switch to the authenticator switch, as shown in Figure 60-7.
- Auto enablement: Automatically enables trunk configuration on the authenticator switch, allowing user traffic from multiple VLANs coming from supplicant switches. Configure the cisco-av-pair as *device-traffic-class=switch* at the ACS. (You can configure this under the *group* or the *user* settings.)

Figure 60-7 Authenticator and Supplicant Switch using CISP



1	Workstations (clients)	**2**	Supplicant switch (outside wiring closet)
3	Authenticator switch	**4**	Access control server (ACS)
5	Trunk port		

When configuring NEAT and CISP, follow these guidelines and restrictions:

- You can configure NEAT ports with the same configurations as the other authentication ports. When the supplicant switch authenticates, the port mode is changed from *access* to *trunk* based on the switch vendor-specific attributes (VSAs). (device-traffic-class=switch).
- The VSA changes the authenticator switch port mode from access to trunk and enables 802.1x trunk encapsulation and the access VLAN if any would be converted to a native trunk VLAN. VSA does not change any of the port configurations on the supplicant

For more information, see the "Configuring an Authenticator and a Supplicant Switch with NEAT" section on page 60-64.

802.1X Authentication Feature Configuration Guidelines

This section has configuration guidelines for these features:

- 802.1X Authentication, page 60-29
- 802.1X Host Mode, page 60-30
- VLAN Assignment, Guest VLAN, Restricted VLAN, and Inaccessible Authentication Bypass, page 60-31
- MAC Authentication Bypass, page 60-32
- Web-Based Authentication, page 60-33

802.1X Authentication

When configuring 802.1X authentication, note the following guidelines:

- In releases where CSCtg01609 is not resolved, on ports with the **authentication port-control auto** command or the **dot1x pae supplicant** command configured, you cannot successfully enter the **no switchport** command. In releases where CSCtg01609 is resolved, on ports with any **authentication**, **dot1x**, or **mab** command configured, you cannot successfully enter the **no switchport** command.

  Note Enter the **default interface** *type slot/port* command to revert to the default port configuration.

- When 802.1X authentication is enabled, ports are authenticated before any other Layer 2 or Layer 3 features are enabled.
- If you try to change the mode of an 802.1X-enabled port (for example, from access to trunk), an error message appears, and the port mode is not changed.
- With Cisco IOS Release 12.2(33)SXH and later releases, you can configure port security and 802.1X port-based authentication on the same port. We do not recommend configuring both together.
- If the VLAN to which an 802.1X-enabled port is assigned changes, this change is transparent and does not affect the switch. For example, this change occurs if a port is assigned to a RADIUS server-assigned VLAN and is then assigned to a different VLAN after reauthentication.

  If the VLAN to which an 802.1X port is assigned to shut down, disabled, or removed, the port becomes unauthorized. For example, the port is unauthorized after the access VLAN to which a port is assigned shuts down or is removed.
- The 802.1X protocol is supported on Layer 2 static-access ports, voice VLAN ports, and Layer 3 routed ports, but it is not supported on these port types:
  - Trunk ports:

    With Release 12.2(33)SXJ and later releases, you can enter the commands to enable 802.1X authentication on a trunk port or change the mode of an 802.1X-enabled port to trunk, but 802.1X authentication works only on trunk ports configured to support a switch supplicant (SSw). Configure 802.1X authentication on trunk ports only to support NEAT (CSCtx16322).

    With releases earlier than Release 12.2(33)SXJ, if you try to enable 802.1X authentication on a trunk port, an error message appears, and 802.1X authentication is not enabled. If you try to change the mode of an 802.1X-enabled port to trunk, an error message appears, and the port mode is not changed.

- Dynamic ports—A port in dynamic mode can negotiate with its neighbor to become a trunk port. If you try to enable 802.1X authentication on a dynamic port, an error message appears, and 802.1X authentication is not enabled. If you try to change the mode of an 802.1X-enabled port to dynamic, an error message appears, and the port mode is not changed.
- Dynamic-access ports—If you try to enable 802.1X authentication on a dynamic-access (VLAN Query Protocol [VQP]) port, an error message appears, and 802.1X authentication is not enabled. If you try to change an 802.1X-enabled port to dynamic VLAN assignment, an error message appears, and the VLAN configuration is not changed.
- EtherChannel port—Do not configure a port that is an active or a not-yet-active member of an EtherChannel as an 802.1X port. If you try to enable 802.1X authentication on an EtherChannel port, an error message appears, and 802.1X authentication is not enabled.

Note In software releases earlier than Release 12.2(33)SXH, if 802.1X authentication is enabled on a not-yet active port of an EtherChannel, the port does not join the EtherChannel.

- Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) destination ports—You can enable 802.1X authentication on a port that is a SPAN or RSPAN destination port. However, 802.1X authentication is disabled until the port is removed as a SPAN or RSPAN destination port. You can enable 802.1X authentication on a SPAN or RSPAN source port.

Note In software releases earlier than Release 12.2(33)SXH, 802.1X authentication is not supported on voice VLAN ports.

- Before globally enabling 802.1X authentication on a switch by entering the **dot1x system-auth-control** global configuration command, remove the EtherChannel configuration from the interfaces on which 802.1X authentication and EtherChannel are configured.
- Because all traffic from unauthenticated hosts is forwarded to the switch processor, we recommend that you apply rate limiting to this traffic.

802.1X Host Mode

When configuring any host mode, note the following guidelines:

- In most cases when the host mode is changed on a port, any existing 802.1X authentications on that port are deleted. Exceptions are when changing from the single-host mode to any other mode, and when changing from multidomain mode to multiauth mode. In these two cases, existing 802.1X authentications are retained.
- If you enter the **authentication open** interface configuration command in Cisco IOS Release 12.2(33)SXI and later releases, any new MAC address detected on the port will be allowed unrestricted Layer 2 access to the network even before any authentication has succeeded. If you use this command, you should use static default ACLs to restrict Layer 3 traffic. For additional details, see the "Pre-Authentication Open Access" section on page 60-11.

When configuring multiple-hosts mode, note the following guideline:

- If the multiple-hosts port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), all attached clients are denied access to the network.

When configuring MDA host mode, note the following guideline:

- A third-party IP phone's MAC address will initially be assigned to the data VLAN. When tagged voice packets are observed, the device will be removed from the data VLAN and placed on the voice VLAN.

When configuring multiauth host mode, note the following guidelines:

- If one client on a multiauth port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received from that client), the authorization status of the other attached clients is not changed.
- RADIUS-assigned VLANs are not supported on multiauth ports, which can have only one data VLAN. If the authentication server sends VLAN-related attributes, the authentication will succeed but the VLAN assignment will be ignored.
- Although multiple hosts are allowed on the data VLAN, only one host is allowed on the voice VLAN. When one IP phone has been authenticated, further IP phones on the same port will be denied authentication.
- A multiauth port does not support a guest VLAN, authentication-fail VLAN, or with releases earlier than Release 12.2(33)SXJ1, a critical VLAN.

VLAN Assignment, Guest VLAN, Restricted VLAN, and Inaccessible Authentication Bypass

When configuring VLAN assignment, guest VLAN, restricted VLAN, and inaccessible authentication bypass, note the following guidelines:

- When 802.1X authentication is enabled on a port, you cannot configure a port VLAN that is equal to a voice VLAN.
- The 802.1X authentication with VLAN assignment feature is not supported on trunk ports, dynamic ports, or with dynamic-access port assignment through a VMPS.
- You can configure any VLAN except an RSPAN VLAN, a private primary PVLAN, or a voice VLAN as an 802.1X guest VLAN. The guest VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.
- After you configure a guest VLAN for an 802.1X port to which a DHCP client is connected, you might need to get a host IP address from a DHCP server. You can change the settings for restarting the 802.1X authentication process on the switch before the DHCP process on the client times out and tries to get a host IP address from the DHCP server. Decrease the settings for the 802.1X authentication process (**dot1x timeout quiet-period** and **dot1x timeout tx-period** interface configuration commands). The amount to decrease the settings depends on the connected 802.1X client type.
- When configuring the 802.1X VLAN user distribution feature, follow these guidelines:
  - A maximum of 100 VLAN groups can be configured, and a maximum of 4094 VLANs can be mapped to a VLAN group.
  - A VLAN can be mapped to more than one VLAN group.
  - A guest VLAN, a critical VLAN, or a restricted VLAN can be mapped to a VLAN group.
  - A VLAN group name cannot be specified as a guest VLAN, a critical VLAN, or a restricted VLAN.

- You can modify a VLAN group by adding or removing a VLAN, but at least one VLAN must be mapped to the VLAN group. If you remove the last VLAN from the VLAN group, the VLAN group is deleted.
- Removing an existing VLAN from the VLAN group name does not revoke the authentication status of the ports in the VLAN, but the mappings are removed from the existing VLAN group.
- Deleting an existing VLAN group name does not revoke the authentication status of the ports in any VLAN within the group, but the VLAN mappings to the VLAN group are removed.

- When configuring the inaccessible authentication bypass feature, follow these guidelines:
  - The inaccessible authentication bypass feature is supported on 802.1X ports in single-host mode, multiple-hosts mode, and MDA mode.
  - If the client is running Windows XP and the port to which the client is connected is in the critical-authentication state, Windows XP might report that the interface is not authenticated.
  - If the Windows XP client is configured for DHCP and has an IP address from the DHCP server, receiving an EAP-Success message on a critical port might not reinitiate the DHCP configuration process.
  - You can configure the inaccessible authentication bypass feature and the critical VLAN on an 802.1X port. If the switch tries to reauthenticate a critical port in a critical VLAN and all the RADIUS servers are unavailable, the switch changes the port state to the critical authentication state and the port remains in the critical VLAN.
  - You can configure the inaccessible bypass feature and port security on the same port.
- You can configure any VLAN except an RSPAN VLAN or a voice VLAN as an 802.1X restricted VLAN. The restricted VLAN feature is not supported on internal VLANs (routed ports) or trunk ports; it is supported only on access ports.

MAC Authentication Bypass

When configuring MAC authentication bypass, note the following guidelines:

- Unless otherwise stated, the MAC authentication bypass guidelines are the same as the 802.1X authentication guidelines. For more information, see the "802.1X Authentication" section on page 60-29.
- If you disable MAC authentication bypass from a port after the port has been authorized with its MAC address, the session will be removed.
- When MAC authentication bypass with EAP has been enabled on an interface, it is not disabled by a subsequent **default interface** command on the interface.
- If the port is in the unauthorized state and the client MAC address is not the authentication-server database, the port remains in the unauthorized state. However, if the client MAC address is added to the database, the switch can use MAC authentication bypass to reauthorize the port.
- If the port is in the authorized state, the port remains in this state until reauthorization occurs.
- To use MAC authentication bypass on a routed port, make sure that MAC address learning is enabled on the port.
- In Release 12.2(33)SXH and later releases, you can optionally configure a timeout period for hosts that are connected by MAC authentication bypass but are inactive. The range is 1 to 65535 seconds, but should be set to a value less than the reauthentication timeout. You must enable port security before configuring a timeout value. For more information, see the "Configuring Port Security" section on page 62-5.

Web-Based Authentication

When configuring web-based authentication, note the following guidelines:

- Fallback to web-based authentication is configured on switch ports in access mode. Ports in trunk mode are not supported.
- Fallback to web-based authentication is not supported on EtherChannels or EtherChannel members.
- Although fallback to web-based authentication is an interface-specific configuration, the web-based authentication fallback behavior is defined in a global fallback profile. If the global fallback configuration changes, the new profile will not be used until the next instance of authentication fallback.

For detailed information on configuring web-based authentication, see Chapter 61, "Configuring Web-Based Authentication."

Configuring 802.1X Port-Based Authentication

These sections describe how to configure 802.1X port-based authentication:

Default 802.1X Port-Based Authentication Configuration

Table 60-2 shows the default 802.1X configuration.

Table 60-2 Default 802.1X Configuration

Feature	Default Setting
Switch 802.1X enable state	Disabled.
Per-port 802.1X enable state	Disabled (force-authorized). The port sends and receives normal traffic without 802.1X-based authentication of the client.
AAA	Disabled.
RADIUS server • IP address • UDP authentication port • Key	 • None specified. • 1812. • None specified.
Host mode	Single-host mode.
Control direction	Bidirectional control.
Periodic reauthentication	Disabled.
Number of seconds between reauthentication attempts	3600 seconds.
Reauthentication number	2 times (number of times that the switch restarts the authentication process before the port changes to the unauthorized state).
Quiet period	60 seconds (number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client).
Retransmission time	30 seconds (number of seconds that the switch should wait for a response to an EAP request/identity frame from the client before retransmitting the request).

Table 60-2 Default 802.1X Configuration (continued)

Feature	Default Setting
Maximum retransmission number	2 times (number of times that the switch will send an EAP-request/identity frame before restarting the authentication process).
Client timeout period	30 seconds (when relaying a request from the authentication server to the client, the amount of time the switch waits for a response before retransmitting the request to the client).
Authentication server timeout period	30 seconds (when relaying a response from the client to the authentication server, the amount of time the switch waits for a reply before retransmitting the response to the server).
Inactivity timeout	Disabled.
Guest VLAN	None specified.
Inaccessible authentication bypass	Disabled.
Restricted VLAN	None specified.
Authenticator (switch) mode	None specified.
MAC authentication bypass	Disabled. **Note** When MAC authentication bypass with EAP has been enabled on an interface, it is not disabled by a subsequent **default interface** command on the interface.

Enabling 802.1X Authentication

To enable 802.1X port-based authentication, you must enable AAA and specify the authentication method list.

A method list describes the sequence and authentication methods to be queried to authenticate a user. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle, the authentication process stops, and no other authentication methods are attempted.

To allow VLAN assignment, you must enable AAA authorization to configure the switch for all network-related service requests.

The 802.1X AAA process is as follows:

1. A user connects to a port on the switch.
2. Authentication is performed.
3. VLAN assignment is enabled, as appropriate, based on the RADIUS server configuration.
4. The switch sends a start message to an accounting server.
5. Reauthentication is performed, as necessary.

6. The switch sends an interim accounting update to the accounting server that is based on the result of reauthentication.
7. The user disconnects from the port.
8. The switch sends a stop message to the accounting server.

To configure 802.1X port-based authentication, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`aaa new-model`**	Enables AAA.
Step 2	`Router(config)#` **`aaa authentication dot1x`** `{`**`default`**`}` *`method1`* `[`*`method2...`*`]`	Creates an 802.1X port-based authentication method list. To create a default list that is used when a named list is *not* specified in the **aaa authentication** command, use the **default** keyword followed by the method that is to be used in default situations. The default method list is automatically applied to all ports. For *method1*, enter the **group radius** keywords to use the list of all RADIUS servers for authentication. Though other keywords are visible in the command-line help string, only the **group radius** keywords are supported.
Step 3	`Router(config)#` **`dot1x system-auth-control`**	Globally enables 802.1X port-based authentication.
Step 4	`Router(config)#` **`aaa authorization network {default} group radius`**	(Optional) Configures the switch to use user-RADIUS authorization for all network-related service requests such as VLAN assignment.
Step 5	`Router(config)#` **`radius-server host`** *`ip-address`*	Specifies the IP address of the RADIUS server.
Step 6	`Router(config)#` **`radius-server key`** *`string`*	Specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 7	`Router(config)#` **`mls acl tcam static-share`**	(Optional) Enables static sharing, which allows more efficient use of the TCAM when a number of interfaces have the same PACL and VLAN-based features.
Step 8	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 9	`Router(config-if)#` **`switchport mode access`**	Sets the port to access mode only if you configured the RADIUS server in previous steps.
Step 10	**Cisco IOS Release 12.2(33)SXI or later releases:** `Router(config-if)#` **`authentication port-control auto`** **Releases earlier than Release 12.2(33)SXI:** `Router(config-if)#` **`dot1x port-control auto`**	Enables port-based authentication on the interface. The **no** form of the command disables port-based authentication on the interface. For feature interaction information, see the "802.1X Authentication Feature Configuration Guidelines" section on page 60-29.
Step 11	`Router(config-if)#` **`dot1x pae authenticator`**	Enables 802.1X authentication on the interface.

	Command	Purpose
Step 12	`Router(config)# end`	Returns to privileged EXEC mode.
Step 13	`Router# show dot1x all`	Verifies your entries. Check the Status column in the 802.1X Port Summary section of the display. An *enabled* status means the port-control value is set either to **auto** or to **force-unauthorized**.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable AAA and 802.1X on Fast Ethernet port 5/1:

```
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# dot1x system-auth-control
Router(config)# mls acl tcam static-share
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show dot1x all

Sysauthcontrol              Enabled
Dot1x Protocol Version            2

Dot1x Info for GigabitEthernet1/7
-----------------------------------
PAE                        = AUTHENTICATOR
PortControl                = AUTO
ControlDirection           = Both
HostMode                   = SINGLE_HOST
QuietPeriod                = 60
ServerTimeout              = 30
SuppTimeout                = 30
ReAuthMax                  = 2
MaxReq                     = 2
TxPeriod                   = 30
```

Configuring Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by any of the following:

- Host name
- Host IP address
- Host name and specific UDP port numbers
- IP address and specific UDP port numbers

The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service (for example, authentication) the second host entry configured acts as the failover backup to the first one. The RADIUS host entries are tried in the order that they were configured.

To configure the RADIUS server parameters, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip radius source-interface interface_name`	Specifies that the RADIUS packets have the IP address of the indicated interface.
Step 2	`Router(config)# radius-server host {hostname | ip_address}`	Configures the RADIUS server host name or IP address on the switch. If you want to use multiple RADIUS servers, reenter this command.
Step 3	`Router(config)# radius-server key string`	Configures the authorization and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.

When you configure the RADIUS server parameters, note the following information:

- For *hostname* or *ip_address,* specify the host name or IP address of the remote RADIUS server.
- Specify the **key** *string* on a separate command line.
- For **key** *string*, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server.
- When you specify the **key** *string*, spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon.
- You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius-server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and the **radius-server key** global configuration commands. For more information, see the *Cisco IOS Security Configuration Guide, Release 12.2,* publication at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/fsecur_c.html

  and the *Cisco IOS Security Command Reference*, Release 12.2, publication at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html

Note You also need to configure some settings on the RADIUS server. These settings include the IP address of the switch and the key string to be shared by both the server and the switch. For more information, see the RADIUS server documentation.

This example shows how to configure the RADIUS server parameters on the switch:

```
Router(config)# ip radius source-interface Vlan80
Router(config)# radius-server host 172.120.39.46
Router(config)# radius-server key rad123
```

Configuring 802.1X Authenticator Host Mode

An 802.1X-enabled port can grant access to a single client or multiple clients as described in the "802.1X Host Modes" section on page 60-9.

To configure the host mode of an 802.1X-authorized port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control auto`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control auto`
		Enables port-based authentication on the interface.
		The **no** form of the command disables port-based authentication on the interface.
		For feature interaction information, see the "802.1X Authentication Feature Configuration Guidelines" section on page 60-29.

Step 3	`Router(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the interface.
Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:** `Router(config-if)# authentication host-mode single-host`	Allows a single authenticated host (client) on an authorized port.
	`Router(config-if)# authentication host-mode multi-host`	Allows multiple clients on an authorized port when one client is authenticated.
	`Router(config-if)# authentication host-mode multi-domain`	Allows a single IP phone and a single data client to independently authenticate on an authorized port.
	`Router(config-if)# authentication host-mode multi-auth`	Allows a single IP phone and multiple data clients to independently authenticate on an authorized port.
	Releases earlier than Release 12.2(33)SXI: `Router(config-if)# dot1x host-mode {single-host | multi-host}`	
Step 5	`Router(config-if)# authentication open`	(Optional) With Cisco IOS Release 12.2(33)SXI or later releases, enables pre-authentication open access.
Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show dot1x interface type[1] slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable 802.1X on Fast Ethernet interface 5/1 and to allow multiple hosts:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# authentication host-mode multi-host
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x host-mode multi-host
```

Enabling Fallback Authentication

On a port in multiauth mode, either or both of MAB and web-based authentication can be configured as fallback authentication methods for non-802.1X hosts (those that do not respond to EAPOL). You can configure the order and priority of the authentication methods.

For detailed configuration information for MAB, see the "Configuring MAC Authentication Bypass" section on page 60-57.

For detailed configuration information for web-based authentication, see Chapter 61, "Configuring Web-Based Authentication."

In Cisco IOS Release 12.2(33)SXI and later releases, to enable fallback authentication, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip admission name`** *`rule-name`* **`proxy http`**	Configures an authentication rule for web-based authentication.
Step 2	`Router(config)#` **`fallback profile`** *`profile-name`*	Creates a fallback profile for web-based authentication.
Step 3	`Router(config-fallback-profile)#` **`ip access-group`** *`rule-name`* **`in`**	Specifies the default ACL to apply to network traffic before web-based authentication.
Step 4	`Router(config-fallback-profile)#` **`ip admission name`** *`rule-name`*	Associates an IP admission rule with the profile, and specifies that a client connecting by web-based authentication uses this rule.
Step 5	`Router(config-fallback-profile)#` **`exit`**	Returns to global configuration mode.
Step 6	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Specifies the port to be configured, and enters interface configuration mode.
Step 7	`Router(config-if)#` **`authentication port-control auto`**	Enables authentication on the port.
Step 8	`Router(config-if)#` **`authentication order`** *`method1`* [*`method2`*] [*`method3`*]	(Optional) Specifies the fallback order of authentication methods to be used. The three values of *method*, in the default order, are **dot1x**, **mab**, and **webauth**. The specified order also determines the relative priority of the methods for reathentication, from highest to lowest.
Step 9	`Router(config-if)#` **`authentication priority`** *`method1`* [*`method2`*] [*`method3`*]	(Optional) Overrides the relative priority of authentication methods to be used. The three values of *method*, in the default order of priority, are **dot1x**, **mab**, and **webauth**.
Step 10	`Router(config-if)#` **`authentication event fail action next-method`**	Specifies that the next configured authentication method will be used if authentication fails.
Step 11	`Router(config-if)#` **`mab`** [**`eap`**]	Enables MAC authentication bypass. The optional **eap** keyword specifies that the EAP extension is used during RADIUS authentication.
Step 12	`Router(config-if)#` **`authentication fallback`** *`profile-name`*	Enables web-based authentication using the specified profile.

	Command	Purpose
Step 13	`Router(config-if)# authentication violation [shutdown	restrict]`	(Optional) Configures the disposition of the port if a security violation occurs. The default action is to shut down the port. If the **restrict** keyword is configured, the port will not be shutdown, but trap entries will be installed for the violating MAC address, and traffic from that MAC address will be dropped.
Step 14	`Router(config-if)# authentication timer inactivity {seconds	server}`	(Optional) Configures the inactivity timeout value for MAB and 802.1X. By default, inactivity aging is disabled for a port. • *seconds*—Specifies inactivity timeout period. The range is from 1 to 65535 seconds. • server—Specifies that the inactivity timeout period value will be obtained from the authentication server.
Step 15	`Router(config-if)# authentication timer restart seconds`	(Optional) Specifies a period after which the authentication process will restart in an attempt to authenticate an unauthorized port. • *seconds*—Specifies the restart period. The range is from 1 to 65535 seconds.
Step 16	`Router(config-if)# exit`	Returns to global configuration mode.
Step 17	`Router(config)# ip device tracking`	Enables the IP device tracking table, which is required for web-based authentication.
Step 18	`Router(config)# ip device tracking [probe {count count	delay delay_interval	interval interval}]`	(Optional) Configures these parameters for the IP device tracking table: • *count*—Number of times that the switch sends the ARP probe. The range is 1 to 5. The default is 3. • *delay_interval* (implemented in releases where CSCtn27420 is resolved)—Number of seconds that the switch delays sending an ARP probe, triggered by link-up and ARP probe generation by the tracked device. The range is 1 to 120 seconds. The default is 0 seconds. • *interval*—Number of seconds that the switch waits for a response before resending the ARP probe. The range is 30 to 300 seconds. The default is 30 seconds.
Step 19	`Router(config)# exit`	Returns to privileged EXEC mode.
Step 20	`Router# show dot1x interface type[1] slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable 802.1X fallback to MAB, and then to enable web-based authentication, on an 802.1X-enabled port:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# ip admission name rule1 proxy http
Router(config)# fallback profile fallback1
Router(config-fallback-profile)# ip access-group default-policy in
Router(config-fallback-profile)# ip admission rule1
```

```
Router(config-fallback-profile)# exit
Router(config)# interface gigabit1/1
Router(config-if)# switchport mode access
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# authentication order dot1x mab webauth
Router(config-if)# mab eap
Router(config-if)# authentication fallback fallback1
Router(config-if)# exit
Router(config)# ip device tracking
Router(config)# exit
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# ip admission name rule1 proxy http
Router(config)# fallback profile fallback1
Router(config-fallback-profile)# ip access-group default-policy in
Router(config-fallback-profile)# ip admission rule1
Router(config-fallback-profile)# exit
Router(config)# interface gigabit1/1
Router(config-if)# switchport mode access
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x fallback fallback1
Router(config-if)# exit
Router(config)# ip device tracking
Router(config)# exit
```

Enabling Periodic Reauthentication

With Release 12.2(33)SXH and later releases, you can enable periodic 802.1X client reauthentication and specify how often it occurs. You can specify the reauthentication period manually or you can use the session-timeout period specified by the RADIUS server. If you enable reauthentication without specifying a time period, the number of seconds between reauthentication attempts is 3600.

To enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication periodic`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x reauthentication`
		Enables periodic reauthentication of the client, which is disabled by default.

	Command	Purpose

Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication timer reauthenticate [seconds | server]`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x timeout reauth-period [seconds | server]`
		Specifies the number of seconds between reauthentication attempts using these keywords:
		• *seconds*—Sets the number of seconds from 1 to 65535; the default is 3600 seconds.
		• **server**—Sets the number of seconds based on the value of the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]).
		This command affects the operation of the switch only if periodic reauthentication is enabled.

Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Router# show dot1x interface type slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication periodic
Router(config-if)# authentication timer reauthenticate 4000
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x reauthentication
Router(config-if)# dot1x timeout reauth-period 4000
```

Manually Reauthenticating the Client Connected to a Port

Note Reauthentication does not disturb the status of an already authorized port.

To manually reauthenticate the client connected to a port, perform this task:

	Command	Purpose
Step 1	`Router# dot1x re-authenticate interface type[1] slot/port`	Manually reauthenticates the client connected to a port.
Step 2	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to manually reauthenticate the client connected to Fast Ethernet port 5/1:

```
Router# dot1x re-authenticate interface fastethernet 5/1
```

Initializing Authentication for the Client Connected to a Port

Note Initializing authentication disables any existing authentication before authenticating the client connected to the port.

To initialize the authentication for the client connected to a port, perform this task:

	Command	Purpose
Step 1	`Router#` **`dot1x initialize interface`** *`type`*[1] *`slot/port`*	Initializes the authentication for the client connected to a port.
Step 2	`Router#` **`show dot1x all`**	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to initialize the authentication for the client connected to Fast Ethernet port 5/1:

```
Router# dot1x initialize interface fastethernet 5/1
```

Removing 802.1X Client Information

To completely delete all existing supplicants from an interface or from all the interfaces on the switch, perform this task:

	Command	Purpose
Step 1	`Router#` **`clear dot1x interface`** *`type`*[1] *`slot/port`*	Removes 802.1X client information for the client connected to the specified port.
	`Router#` **`clear dot1x all`**	Removes 802.1X client information for all clients connected to all ports.
Step 2	`Router#` **`show dot1x all`**	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to remove 802.1X client information for the client connected to Fast Ethernet port 5/1:

```
Router# clear dot1x interface fastethernet 5/1
```

Clearing Authentication Sessions

To clear all or selected authentication sessions, perform this task:

Command	Purpose
Router# **clear authentication sessions** [**handle** *handle*] [**interface** *interface*] [**mac** *mac*] [**method** *method*]	Clears current authentication sessions. With no options specified, all current active sessions will be cleared. The keywords can be added and combined to clear specific sessions or subset of sessions.

This example shows how to clear all MAB authentication sessions connected to Fast Ethernet port 5/1:

```
Router# clear authentication sessions interface fastethernet 5/1 method mab
```

Changing 802.1X Timeouts

You can change several 802.1X timeout attributes using the **dot1x timeout {attribute}** *seconds* command form in the interface configuration mode. This section shows in detail how to change the quiet period timeout, followed by descriptions of how to change other 802.1X timeouts using the same command form.

Setting the Quiet Period

When the switch cannot authenticate the client, the switch remains idle for a set period of time and then tries again. The **dot1x timeout quiet-period** interface configuration command controls the idle period. A failed authentication of the client might occur because the client provided an invalid password. You can provide a faster response time to the user by entering a number smaller than the default.

To change the quiet period, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	Router(config-if)# **dot1x timeout quiet-period** *seconds*	Sets the number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client. The range is 0 to 65535 seconds; the default is 60.
Step 3	Router(config-if)# **end**	Returns to privileged EXEC mode.
Step 4	Router# **show dot1x all**	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to set the quiet period on the switch to 30 seconds:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout quiet-period 30
```

This example shows how to restore the default quiet period on the switch:

```
Router(config-if)# no dot1x timeout quiet-period
```

Setting the Switch-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the switch with an EAP-response/identity frame. If the switch does not receive this response, it waits a set period of time (known as the retransmission time), and then retransmits the frame.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific operational problems with certain clients and authentication servers.

To change the amount of time that the switch switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request, use the **dot1x timeout tx-period** *seconds* command in the interface configuration mode. The range is 1 to 65535 seconds; the default is 30. To return to the default retransmission time, use the **no dot1x timeout tx-period** command.

This example shows how to set 60 as the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout tx-period 60
```

Setting the Switch-to-Client Retransmission Time for EAP-Request Frames

The client notifies the switch that it received the EAP-request frame. If the switch does not receive this notification, the switch waits a set period of time, and then retransmits the frame.

To set the amount of time that the switch waits for notification, use the **dot1x timeout supp-timeout** *seconds* command in the interface configuration mode. The range is 1 to 65535 seconds; the default is 30. To return to the default retransmission time, use the **no dot1x supp-timeout** command.

This example shows how to set the switch-to-client retransmission time for the EAP-request frame to 25 seconds:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout supp-timeout 25
```

Setting the Switch-to-Authentication-Server Retransmission Time for Layer 4 Packets

The authentication server notifies the switch each time it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the switch waits a set period of time and then retransmits the packet.

To set the value for the retransmission of Layer 4 packets from the switch to the authentication server, use the **dot1x timeout server-timeout** *seconds* command in the interface configuration mode. The range is 1 to 65535 seconds; the default is 30. To return to the default retransmission time, use the **no dot1x server-timeout** command.

This example shows how to set the switch-to-authentication-server retransmission time for Layer 4 packets to 25 seconds:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout server-timeout 25
```

Setting the Switch-to-Client Frame Retransmission Number

In addition to changing the switch-to-client retransmission time, you can change the number of times that the switch sends an EAP-request/identity frame (assuming no response is received) to the client before restarting the authentication process.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific operational problems with certain clients and authentication servers.

To set the switch-to-client frame retransmission number, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)#` **`dot1x max-req`** *`count`*	Sets the number of times that the switch sends an EAP-request/identity frame to the client before restarting the authentication process. The range is 1 to 10; the default is 2.
Step 3	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Router#` **`show dot1x all`**	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to set 5 as the number of times that the switch sends an EAP-request/identity request before restarting the authentication process:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x max-req 5
```

Setting the Reauthentication Number

You can also change the number of times that the switch restarts the authentication process before the port changes to the unauthorized state.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific operational problems with certain clients and authentication servers.

To set the reauthentication number, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)#` **`dot1x max-reauth-req`** *`count`*	Sets the number of times that the switch restarts the authentication process before the port changes to the unauthorized state. The range is 0 to 10; the default is 2.

	Command	Purpose
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to set 4 as the number of times that the switch restarts the authentication process before the port changes to the unauthorized state:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x max-reauth-req 4
```

Configuring IEEE 802.1X Accounting

Enabling AAA system accounting with 802.1X accounting allows system reload events to be sent to the accounting RADIUS server for logging. The server then can determine that all active 802.1X sessions are closed.

Because RADIUS uses the unreliable UDP transport protocol, accounting messages might be lost due to poor network conditions. If the switch does not receive the accounting response message from the RADIUS server after a configurable number of retransmissions of an accounting request, this system message appears:

```
Accounting message %s for session %s failed to receive Accounting Response.
```

When the stop message is not sent successfully, this message appears:

```
00:09:55: %RADIUS-3-NOACCOUNTINGRESPONSE: Accounting message Start for session
172.20.50.145 sam 11/06/03 07:01:16 11000002 failed to receive Accounting Response.
```

Note You must configure the RADIUS server to perform accounting tasks, such as logging start, stop, and interim-update messages and time stamps. To turn on these functions, enable logging of "Update/Watchdog packets from this AAA client" in your RADIUS server Network Configuration tab. Next, enable "CVS RADIUS Accounting" in your RADIUS server System Configuration tab.

To configure 802.1X accounting after AAA is enabled on your switch, perform this task:

	Command	Purpose
Step 1	`Router(config)# aaa accounting dot1x default start-stop group radius`	Enables 802.1X accounting using the list of all RADIUS servers.
Step 2	`Router(config)# aaa accounting system default start-stop group radius`	(Optional) Enables system accounting (using the list of all RADIUS servers) and generates system accounting reload event messages when the switch reloads.
Step 3	`Router(config)# end`	Returns to privileged EXEc mode.
Step 4	`Router# show running-config`	Verifies your entries.

Use the **show radius statistics** privileged EXEC command to display the number of RADIUS messages that do not receive the accounting response message.

This example shows how to configure 802.1X accounting. The first command configures the RADIUS server, specifying 1813 as the UDP port for accounting:

```
Router(config)# radius-server host 172.120.39.46 auth-port 1812 acct-port 1813 key rad123
Router(config)# aaa accounting dot1x default start-stop group radius
Router(config)# aaa accounting system default start-stop group radius
```

Configuring VLAN User Distribution

With Cisco IOS Release 12.2(33)SXI1 and later releases, you can define a VLAN group that contains multiple VLANs. For VLAN load balancing, you can then configure the RADIUS server to supply a VLAN group name as part of the response to a user during 802.1X authentication. If the supplied VLAN group name is found among the VLAN group names that you have defined, the newly authenticated user is placed in the least populated VLAN within the VLAN group.

To configure a VLAN group, perform this task:

	Command	Purpose
Step 1	Router(config)# **vlan group** *group-name* **vlan-list** *vlan-list*	Creates a VLAN group or adds VLANs to an existing VLAN group. • *group-name*—The name of the VLAN group. The name may contain up to 32 characters and must begin with a letter. • **vlan-list** *vlan-list*—The VLANs that belong to the VLAN group. Group members can be specified as a single VLAN ID, a list of VLAN IDs, or a VLAN ID range. Multiple entries are separated by a hyphen (-) or a comma (,).
Step 2	Router(config)# **no vlan group** *group-name* **vlan-list** *vlan-list*	(Optional) Removes the members specified by *vlan-list* from a VLAN group. **Note** When no VLANs remain in the VLAN group, the VLAN group is deleted.
Step 3	Router# **show vlan group** [**group-name** *group-name*]	Displays the VLANs and VLAN ranges that are members of the specified VLAN group or of all VLAN groups.

This example shows how to map VLANs 7 through 9 and 11 to a VLAN group:

```
Router(config)# vlan group ganymede vlan-list 7-9,11
```

This example shows how to remove VLAN 7 from a VLAN group:

```
Router(config)# no vlan group ganymede vlan-list 7
```

Configuring a Guest VLAN

With Cisco IOS Release 12.2(33)SXH and later releases, when you configure a guest VLAN, clients that are not 802.1X-capable are put into the guest VLAN when the server does not receive a response to its EAP request/identity frame. Clients that are 802.1X-capable but that fail authentication are not granted network access. When operating as a guest VLAN, a port functions in multiple-hosts mode regardless of the configured host mode of the port.

To configure a guest VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.

Step 2	`Router(config-if)# switchport mode access`
	or
	`Router(config-if)# switchport mode private-vlan host`
		Sets the port to access mode.
		or
		Configures the port as a private VLAN host port.
		The guest VLAN is not supported on routed or trunk ports.

Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control auto`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control auto`
		Enables authentication on the port.

Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication event no-response action authorize vlan vlan-id`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x guest-vlan vlan-id`
		Specifies an active VLAN as a guest VLAN. The range is 1 to 4094.
		You can configure any active VLAN except an internal VLAN (routed port), an RSPAN VLAN, a private primary PVLAN, or a voice VLAN as a guest VLAN.

Step 5	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# dot1x pae authenticator`
	or
	`Router(config-if)# mab`
		Specifies whether the port authentication method is 802.1X or MAC address bypass.
		In Cisco IOS Release 12.2(33)SXH and earlier releases, this command is not needed, and the method will be 802.1X.

Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show dot1x interface type slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable VLAN 2 as an 802.1X guest VLAN:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# authentication event no-response action authorize vlan 2
Router(config-if)# dot1x pae authenticator
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x guest-vlan 2
```

This example shows how to set 3 seconds as the client notification timeout on the switch, to set 15 as the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before resending the request, and to enable VLAN 2 as an 802.1X guest VLAN when an 802.1X port is connected to a DHCP client:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config-if)# dot1x timeout supp-timeout 3
Router(config-if)# dot1x timeout tx-period 15
Router(config-if)# authentication event no-response action authorize vlan 2
Router(config-if)# dot1x pae authenticator
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config-if)# dot1x timeout supp-timeout 3
Router(config-if)# dot1x timeout tx-period 15
Router(config-if)# dot1x guest-vlan 2
```

Configuring a Restricted VLAN

When you configure a restricted VLAN on a switch, clients that are 802.1X-compliant are moved into the restricted VLAN when the authentication server does not receive a valid username and password. When operating as a restricted VLAN, a port functions in single-host mode regardless of the configured host mode of the port.

To configure a restricted VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)#` **`switchport mode access`** `or` `Router(config-if)#` **`switchport mode private-vlan host`**	Sets the port to access mode. or Configures the port as a private-VLAN host port.
Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:** `Router(config-if)#` **`authentication port-control auto`** **Releases earlier than Release 12.2(33)SXI:** `Router(config-if)#` **`dot1x port-control auto`**	Enables port-based authentication on the port.
Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:** `Router(config-if)#` **`authentication event fail`** `[`**`retry`** *`retries`*`]` **`action authorize vlan`** *`vlan-id`* **Releases earlier than Release 12.2(33)SXI:** `Router(config-if)#` **`dot1x auth-fail vlan`** *`vlan-id`*	Specifies an active VLAN as a restricted VLAN. The range for *vlan-id* is 1 to 4094. (Optional) The **retry** keyword specifies a number of authentication attempts to allow before a port moves to the restricted VLAN. You can configure any active VLAN except an internal VLAN (routed port), an RSPAN VLAN, a private primary PVLAN, or a voice VLAN as a restricted VLAN.

	Command	Purpose
Step 5	**Releases earlier than Release 12.2(33)SXI:** `Router(config-if)# dot1x auth-fail max-attempts max-attempts`	(Optional) The **max-attempts** keyword specifies a number of authentication attempts to allow before a port moves to the restricted VLAN.
Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show dot1x interface type slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To disable and remove the restricted VLAN, use the **no** form of the **authentication event fail** command or the **dot1x auth-fail** command. The port returns to the unauthorized state.

You can configure the maximum number of authentication attempts allowed before a user is assigned to the restricted VLAN.

- In Cisco IOS Release 12.2(33)SXI and later releases, you can set the number of attempts by using the **retry** keyword in the **authentication event fail** [**retry** *retries*] **action authorize vlan** command. The range of *retries* (allowable authentication attempts) is 1 to 5. The default is 2 attempts.
- In Cisco IOS Release 12.2(33)SXH, you can set the number of attempts by using the **dot1x auth-fail max-attempts** *max-attempts* interface configuration command. The range of *max-attempts* (allowable authentication attempts) is 1 to 3. The default is 3 attempts.

This example shows how to enable VLAN 2 as a restricted VLAN, with assignment of a host after 3 failed attempts:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# authentication event fail retry 3 action authorize vlan 2
Router(config-if)# dot1x pae authenticator
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x auth-fail vlan 2
Router(config-if)# dot1x auth-fail max-attempts 3
```

Configuring the Inaccessible Authentication Bypass Feature

You can configure the inaccessible bypass feature, also referred to as critical authentication or the AAA fail policy.

To configure the port as a critical port and enable the inaccessible authentication bypass feature, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`radius-server dead-criteria time`** *`time`* **`tries`** *`tries`*	(Optional) Sets the conditions that are used to decide when a RADIUS server is considered unavailable or *dead*. The range for *time* is from 1 to 120 seconds. The switch dynamically determines the default *seconds* value that is 10 to 60 seconds. The range for *tries* is from 1 to 100. The switch dynamically determines the default *tries* parameter that is 10 to 100.
Step 2	`Router(config)#` **`radius-server deadtime`** *`minutes`*	(Optional) Sets the number of minutes that a RADIUS server is not sent requests. The range is from 0 to 1440 minutes (24 hours). The default is 0 minutes.

	Command	Purpose
Step 3	`Router(config)#` **radius-server host** *ip-address* [**acct-port** *udp-port*] [**auth-port** *udp-port*] [**key** *string*] [**test username** *name* [**idle-time** *time*] [**ignore-acct-port**] [**ignore-auth-port**]]	(Optional) Configures the RADIUS server parameters by using these keywords: • **acct-port** *udp-port*—Specifies the UDP port for the RADIUS accounting server. The range for the UDP port number is from 0 to 65536. The default is 1646. • **auth-port** *udp-port*—Specifies the UDP port for the RADIUS authentication server. The range for the UDP port number is from 0 to 65536. The default is 1645. **Note** You should configure the UDP port for the RADIUS accounting server and the UDP port for the RADIUS authentication server to nondefault values. • **key** *string*—Specifies the authentication and encryption key for all RADIUS communication between the switch and the RADIUS daemon. **Note** You can also configure the authentication and encryption key by using the **radius-server key** {**0** *string* | **7** *string* | *string*} global configuration command. • **test username** *name*—Enables automated testing of the RADIUS server status, and specify the username to be used. • **idle-time** *time*—Sets the interval of time in minutes after which the switch sends test packets to the server. The range is from 1 to 35791 minutes. The default is 60 minutes (1 hour). • **ignore-acct-port**—Disables testing on the RADIUS server accounting port. • **ignore-auth-port**—Disables testing on the RADIUS server authentication port.
Step 4	`Router(config)#` **dot1x critical eapol**	(Optional) Specifies that the switch sends an EAPOL-Success message when the switch successfully authenticates the critical port.
Step 5	`Router(config)#` **interface** *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 6	**Cisco IOS Release 12.2(33)SXI or later releases:** `Router(config-if)#` **authentication critical recovery delay** *milliseconds* **Releases earlier than Release 12.2(33)SXI:** `Router(config)#` **dot1x critical recovery delay** *milliseconds*	(Optional) Sets the recovery delay period during which the switch waits to reinitialize a critical port when a RADIUS server that was unavailable becomes available. The range is from 1 to 10000 milliseconds. The default is 1000 milliseconds (a port can be reinitialized every second).

	Command	Purpose

Step 7	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication event server dead action authorize [vlan vlan-id]`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x critical [vlan vlan-id]`
		Enables the inaccessible authentication bypass feature, authorizing ports on the specified VLAN when the AAA server is unreachable. If no VLAN is specified, the access VLAN will be used.
		Note The **vlan** keyword is only available on a switch port.

Step 8	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication event server alive action reinitialize`
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x critical recovery action reinitialize`
		Configures the inaccessible authentication bypass recovery feature, specifying that the recovery action is to authenticate the port when an authentication server becomes available.

Step 9	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 10	`Router# show dot1x [interface type slot/port]`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To return to the RADIUS server default settings, use the **no radius-server dead-criteria**, the **no radius-server deadtime**, and the **no radius-server host** global configuration commands. To return to the default settings of inaccessible authentication bypass, use the **no dot1x critical eapol** global configuration command. To disable inaccessible authentication bypass, use the **no authentication event server dead action authorize** (or **no dot1x critical**) interface configuration command.

This example shows how to configure the inaccessible authentication bypass feature:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# radius-server dead-criteria time 30 tries 20
Router(config)# radius-server deadtime 60
Router(config)# radius-server host 1.1.1.2 acct-port 1550 auth-port 1560 key abc1234 test
username user1 idle-time 30
Router(config)# dot1x critical eapol
Router(config)# authentication critical recovery delay 2000
Router(config)# interface gigabitethernet 1/1
Router(config-if)# authentication event server dead action authorize vlan 123
Router(config-if)# authentication event server alive action reinitialize
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# radius-server dead-criteria time 30 tries 20
Router(config)# radius-server deadtime 60
Router(config)# radius-server host 1.1.1.2 acct-port 1550 auth-port 1560 key abc1234 test
username user1 idle-time 30
Router(config)# dot1x critical eapol
Router(config)# dot1x critical recovery delay 2000
Router(config)# interface gigabitethernet 1/1
Router(config-if)# dot1x critical vlan 123
Router(config-if)# dot1x critical recovery action reinitialize
```

Enabling Critical Voice VLAN Support

Release 12.2(33)SXJ1 and later releases support the critical voice VLAN support feature (see the "Understanding 802.1X Authentication Critical Voice VLAN Support" section on page 60-19).

Note When enabling critical voice VLAN support, follow these guidelines and restrictions:

- Use different VLANs for voice and data.
- The voice VLAN must be configured on the switch (see "Configuring Voice Traffic Support" section on page 15-5).

Enabling Critical Voice VLAN Support in Multidomain Authentication Mode

To enable critical voice VLAN support in multidomain authentication (MDA) mode, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)#` **`authentication event server dead action reinitialize vlan`** *`critical_data_vlan_id`*	Configures a critical data VLAN. **Note** Only required if the **authentication event server dead action authorize vlan** *critical_data_vlan_id* command is not configured on the port (see the "Configuring the Inaccessible Authentication Bypass Feature" section on page 60-53).
Step 3	`Router(config-if)#` **`authentication event server dead action authorize voice`**	Enables the critical voice VLAN support feature, which puts phone traffic into the configured voice VLAN of a port if the authentication server becomes unreachable.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable the critical voice VLAN support feature in MDA mode when the **authentication event server dead action authorize vlan** *critical_data_vlan_id* command is also configured on the port as part of the inaccessible authentication bypass feature:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication event server dead action authorize voice
```

This example shows how to enable the critical voice VLAN support feature in MDA mode when inaccessible authentication bypass is not configured on the port:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication event server dead action reinitialize vlan 10
Router(config-if)# authentication event server dead action authorize voice
```

Enabling Critical Voice VLAN Support in Multiauthentication Mode

To enable critical voice VLAN support in multiauthentication mode, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	Router(config-if)# **authentication event server dead action reinitialize vlan** *critical_data_vlan_id*	Configures a critical data VLAN.
Step 3	Router(config-if)# **authentication event server dead action authorize voice**	Enables the critical voice VLAN support feature, which puts phone traffic into the configured voice VLAN of a port if the authentication server becomes unreachable.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable the critical voice VLAN support feature in multiauthentication mode:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication event server dead action reinitialize vlan 10
Router(config-if)# authentication event server dead action authorize voice
```

Configuring MAC Authentication Bypass

To configure MAC authentication bypass on an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:** Router(config-if)# **authentication port-control** {**auto** | **force-authorized** | **force-unauthorized**} **Releases earlier than Release 12.2(33)SXI:** Router(config-if)# **dot1x port-control** {**auto** | **force-authorized** | **force-unauthorized**}	Enables 802.1X authentication on the port. The keywords have these meanings: • **auto**—Allows only EAPOL traffic until successful authentication. • **force-authorized**—Allows all traffic, requires no authentication. • **force-unauthorized**—Allows no traffic.
Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:** Router(config-if)# **mab** [**eap**] **Releases earlier than Release 12.2(33)SXI:** Router(config-if)# **dot1x mac-auth-bypass** [**eap**]	Enables MAC authentication bypass on the interface. (Optional) Use the **eap** keyword to configure the switch to use EAP for authorization.
Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:** Router(config-if)# **no mab eap** **Releases earlier than Release 12.2(33)SXI:** Router(config-if)# **no dot1x mac-auth-bypass eap**	(Optional) Disables the use of EAP for authorization if EAP was previously configured using the **mab eap** or the **dot1x mac-auth-bypass eap** command.

	Command	Purpose

Step 5	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# no mab`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# no dot1x mac-auth-bypass`
		(Optional) Disables MAC authentication bypass on the interface.
		Note When MAC authentication bypass with EAP has been enabled on an interface, it is not disabled by a subsequent **default interface** command on the interface.

Step 6	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Router# show dot1x interface type slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note To use MAC authentication bypass on a routed port, ensure that MAC address learning is enabled on the port.

This example shows how to enable MAC authentication bypass on a port:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication port-control auto
Router(config-if)# mab
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x mac-auth-bypass
```

Configuring NAC Layer 2 IEEE 802.1X Validation

With Cisco IOS Release 12.2(33)SXH and later releases, you can configure NAC Layer 2 IEEE 802.1X validation, which is also referred to as 802.1X authentication with a RADIUS server. NAC Layer 2 IEEE 802.1X configuration is the same as 802.1X configuration with the additional step of configuring the RADIUS server with a posture token and VLAN assignment.

To configure NAC Layer 2 IEEE 802.1X validation, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication port-control auto`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x port-control auto`
		Enables port-based authentication on the interface.
		The **no** form of the command disables port-based authentication on the interface.
		For feature interaction information, see the "802.1X Authentication Feature Configuration Guidelines" section on page 60-29.

	Command	Purpose

Step 3	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication periodic`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x reauthentication`
		Enables periodic reauthentication of the client, which is disabled by default.

Step 4	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication timer reauthenticate [seconds | server]`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x timeout reauth-period [seconds | server]`
		Specifies the number of seconds between reauthentication attempts using these keywords:
		• *seconds*—Sets the number of seconds from 1 to 65535; the default is 3600 seconds.
		• **server**—Sets the number of seconds based on the value of the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]).
		This command affects the operation of the switch only if periodic reauthentication is enabled.

Step 5	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 6	`Router# show dot1x interface type slot/port`	Verifies your 802.1X authentication configuration. Verify that a NAC posture token is displayed with the 802.1X authentication configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure NAC Layer 2 IEEE 802.1X validation:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config)# authentication port-control auto
Router(config-if)# authentication periodic
Router(config-if)# authentication timer reauthenticate server
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config)# dot1x port-control auto
Router(config-if)# dot1x reauthentication
Router(config-if)# dot1x timeout reauth-period server
```

Configuring NAC Agentless Audit Support

To support the NAC audit architecture for agentless clients, the switch must snoop an authenticating 802.1X client's IP address, create and assign a unique session identifier for the agentless client, and pass this information to the RADIUS server for sharing with the NAC audit server. To allow the switch to obtain and share this information, you must enable ARP and DHCP snooping on the switch and you must enable the sending of certain RADIUS attributes.

To configure the RADIUS and tracking settings to support NAC agentless audit, perform this task:

	Command	Purpose
Step 1	`Router(config)# radius-server attribute 8 include-in-access-req`	Configures the switch to send the Framed-IP-Address RADIUS attribute (Attribute[8]) in access-request or accounting-request packets.
Step 2	`Router(config)# radius-server vsa send authentication`	Configures the network access server to recognize and use vendor-specific attributes (VSAs) (specifically audit-session-id) in RADIUS Access-Requests generated by the switch during the authentication phase.
Step 3	`Router(config)# radius-server vsa send accounting`	Allows VSAs to be included in subsequent RADIUS Accounting-Requests.
Step 4	`Router(config)# ip device tracking`	Enables the IP device tracking table.
Step 5	`Router(config)# ip device tracking [probe {count count | delay delay_interval | interval interval}]`	(Optional) Configures these parameters for the IP device tracking table: • *count*—Number of times that the switch sends the ARP probe. The range is 1 to 5. The default is 3. • *delay_interval* (implemented in releases where CSCtn27420 is resolved)—Number of seconds that the switch delays sending an ARP probe, triggered by link-up and ARP probe generation by the tracked device. The range is 1 to 120 seconds. The default is 0 seconds. • *interval*—Number of seconds that the switch waits for a response before resending the ARP probe. The range is 30 to 300 seconds. The default is 30 seconds.

Configuring the Switch for DACLs or Redirect URLs

To configure switch ports to accept DACLs or redirect URLs from the RADIUS server during authentication of an attached host, perform this task:

	Command	Purpose
Step 1	`Router# config terminal`	Enters global configuration mode.
Step 2	`Router(config)# radius-server vsa send authentication`	Configures the network access server to recognize and use vendor-specific attributes (VSAs) in RADIUS Access-Requests generated by the switch during the authentication phase. **Note** This step is necessary only with redirect URLs or when DACLs are downloaded using VSAs rather than the Filter-ID attribute.
Step 3	`Router(config)# ip device tracking`	Enables the IP device tracking table.

	Command	Purpose
Step 4	Router(config)# **ip device tracking** [probe {**count** *count* | **delay** *delay_interval* | **interval** *interval*}]	(Optional) Configures these parameters for the IP device tracking table: • *count*—Number of times that the switch sends the ARP probe. The range is 1 to 5. The default is 3. • *delay_interval* (implemented in releases where CSCtn27420 is resolved)—Number of seconds that the switch delays sending an ARP probe, triggered by link-up and ARP probe generation by the tracked device. The range is 1 to 120 seconds. The default is 0 seconds. • *interval*—Number of seconds that the switch waits for a response before resending the ARP probe. The range is 30 to 300 seconds. The default is 30 seconds.
Step 5	Router(config)# **ip access-list extended** *dacl-name*	Configures an ACL that will be referenced by the VSA or Filter-ID attribute. **Note** This step is not necessary for DACLs defined on the RADIUS server and downloaded using VSAs.
Step 6	Router(config-std-nacl)# {**permit** | **deny**} ...	Defines the ACL. **Note** The source address must be ANY.
Step 7	Router(config-std-nacl)# **exit**	Returns to global configuration mode.
Step 8	Router(config)# **ip access-list extended** *acl-name*	Configures a default ACL for the ports.
Step 9	Router(config-std-nacl)# {**permit** | **deny**} ...	Defines the ACL.
Step 10	Router(config-std-nacl)# **exit**	Returns to global configuration mode.
Step 11	Router(config)# **interface** *type*[1] *slot/port*	Specifies the port to be configured, and enters interface configuration mode.
Step 12	Router(config-if)# **ip access-group** *acl-name* **in**	Applies the default static ACL on the interface.
Step 13	Router(config-if)# **exit**	Returns to global configuration mode.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure a switch for a downloadable policy:

```
Router# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# radius-server vsa send authentication
Router(config)# ip device tracking
Router(config)# ip access-list extended my_dacl
Router(config-ext-nacl)# permit tcp any host 10.2.3.4
Router(config-ext-nacl)# exit
Router(config)# ip access-list extended default_acl
Router(config-ext-nacl)# permit ip any any
Router(config-ext-nacl)# exit
Router(config)# interface fastEthernet 2/13
Router(config-if)# ip access-group default_acl in
Router(config-if)# exit
```

Configuring a Port to Ignore CoA Commands

To configure the switch to disregard a CoA command to shut down or bounce a specific port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)# [no] authentication command disable-port ignore`	Configures the switch to ignore any CoA command requesting that this port be administratively shut-down.
	`Router(config-if)# [no] authentication command bounce-port ignore`	Configures the switch to ignore any CoA command requesting that this port be held down for a period of time.
Step 3	`Router(config-if)# exit`	Returns to global configuration mode.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Configuring 802.1X Authentication with WoL

Note Wake-on-LAN (WoL) is supported in multiauthentication (multiauth) mode only in releases where CSCti92970 is resolved.

To enable 802.1X authentication with WoL, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Specifies the port to be configured, and enters interface configuration mode.

Step 2	**Cisco IOS Release 12.2(33)SXI or later releases:**
	`Router(config-if)# authentication control-direction {both | in}`
	
	Releases earlier than Release 12.2(33)SXI:
	`Router(config-if)# dot1x control-direction {both | in}`
		Enables 802.1X authentication with WoL on the port, and uses these keywords to configure the port as bidirectional or unidirectional.
		• **both**—Sets the port as bidirectional. The port cannot receive packets from or send packets to the host. By default, the port is bidirectional.
		• **in**—Sets the port as unidirectional. The port can send packets to the host but cannot receive packets from the host.

Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x interface type slot/port`	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To disable 802.1X authentication with WoL, use the **no authentication control-direction** (or the **no dot1x control-direction**) interface configuration command.

This example shows how to enable 802.1X authentication with WoL and set the port as bidirectional:

Cisco IOS Release 12.2(33)SXI or later releases:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# authentication control-direction both
```

Releases earlier than Release 12.2(33)SXI:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x control-direction both
```

Disabling 802.1X Authentication on the Port

You can disable 802.1X authentication on the port by using the **no dot1x pae** interface configuration command.

To disable 802.1X authentication on the port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)#` **`no dot1x pae authenticator`**	Disables 802.1X authentication on the port.
Step 3	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Router#` **`show dot1x interface`** *`type slot/port`*	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

To configure the port as an 802.1X port access entity (PAE) authenticator, which enables 802.1X on the port but does not allow clients connected to the port to be authorized, use the **dot1x pae authenticator** interface configuration command.

This example shows how to disable 802.1X authentication on the port:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# no dot1x pae authenticator
```

Resetting the 802.1X Configuration to the Default Values

To reset the 802.1X configuration to the default values, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Specifies the port to be configured, and enters interface configuration mode.
Step 2	`Router(config-if)#` **`dot1x default`**	Resets the configurable 802.1X parameters to the default values.
Step 3	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Router#` **`show dot1x all`**	Verifies your entries.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to reset a port's 802.1X authentication settings to the default values:

```
Router(config)# interface gigabitethernet 3/27
Router(config-if)# dot1x default
```

Configuring an Authenticator and a Supplicant Switch with NEAT

Release 12.2(33)SXJ and later releases support Network Edge Access Topology (NEAT), which requires one switch to be configured as a supplicant and to be connected to an authenticator switch.

For overview information, see the "Understanding 802.1x Supplicant and Authenticator Switches with Network Edge Access Topology (NEAT)" section on page 60-27.

Note The cisco-av-pairs value must be configured as "device-traffic-class=switch" on the ACS, which sets the interface as a trunk after the supplicant is successfully authenticated.

To configure a switch as an authenticator, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# cisp enable`	Enables CISP.
Step 3	`Router(config)# interface interface-id`	Specifies the port to be configured, and enters interface configuration mode.
Step 4	`Router(config-if)# switchport mode access`	Sets the port mode to **access**.
Step 5	`Router(config-if)# authentication port-control auto`	Sets the port-authentication mode to auto.
Step 6	`Router(config-if)# dot1x pae authenticator`	Configures the interface as a port access entity (PAE) authenticator.
Step 7	`Router(config-if)# spanning-tree portfast`	Enables PortFast on an access port connected to a single workstation or server.
Step 8	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 9	`Router# show running-config interface interface-id`	Verifies your configuration.
Step 10	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to configure a switch as an 802.1x authenticator:

```
Router# configure terminal
Router(config)# cisp enable
Router(config)# interface gigabitethernet1/1
Router(config-if)# switchport mode access
Router(config-if)# authentication port-control auto
Router(config-if)# dot1x pae authenticator
Router(config-if)# spanning-tree portfast trunk
```

To configure a switch as a supplicant, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`cisp enable`**	Enables CISP.
Step 3	`Router(config)#` **`dot1x credentials`** *`profile`*	Creates 802.1x credentials profile. This must be attached to the port that is configured as supplicant.
Step 4	`Router(config)#` **`username`** *`suppswitch`*	Creates a username.
Step 5	`Router(config)#` **`password`** *`password`*	Creates a password for the new username.
Step 6	`Router(config)#` **`dot1x supplicant force-multicast`**	Forces the switch to send *only* multicast EAPOL packets when it receives either unicast or multicast packets, which allows NEAT to work on the supplicant switch in all host modes.
Step 7	`Router(config)#` **`interface`** *`interface-id`*	Specifies the port to be configured, and enters interface configuration mode.
Step 8	`Router(config-if)#` **`switchport trunk encapsulation dot1q`**	Sets the port to trunk mode.
Step 9	`Router(config-if)#` **`switchport mode trunk`**	Configures the interface as a VLAN trunk port.
Step 10	`Router(config-if)#` **`dot1x pae supplicant`**	Configures the interface as a port access entity (PAE) supplicant.
Step 11	`Router(config-if)#` **`dot1x credentials`** *`profile-name`*	Attaches the 802.1x credentials profile to the interface.
Step 12	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 13	`Router#` **`show running-config interface`** *`interface-id`*	Verifies your configuration.
Step 14	`Router#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

This example shows how to configure a switch as a supplicant:

```
Router# configure terminal
Router(config)# cisp enable
Router(config)# dot1x credentials test
Router(config)# username suppswitch
Router(config)# password myswitch
Router(config)# dot1x supplicant force-multicast
Router(config)# interface gigabitethernet1/1
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# dot1x pae supplicant
Router(config-if)# dot1x credentials test
Router(config-if)# end
```

Displaying Authentication Status and Information

This section describes the **show** commands used to display authentication status and information.

- Displaying 802.1X Status, page 60-66
- Displaying Authentication Methods and Status, page 60-67
- Displaying MAC Authentication Bypass Status, page 60-70

For detailed information about the fields in these displays, see the Cisco IOS Master Command List.

Displaying 802.1X Status

To display the global 802.1X administrative and operational status for the switch or the 802.1X settings for individual ports, perform this task:

Command	Purpose
Router# **show dot1x** [**all** | **interface** *type*[1] *slot/port*]	Displays the global 802.1X administrative and operational status for the switch. (Optional) Use the **all** keyword to display the global 802.1X status and the 802.1X settings for all interfaces using 802.1X authentication. (Optional) Use the **interface** keyword to display the 802.1X settings for a specific interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to view only the global 802.1X status:

```
Router# show dot1x
Sysauthcontrol             Disabled
Dot1x Protocol Version            2
Critical Recovery Delay         100
Critical EAPOL             Disabled

Router#
```

This example shows how to view the global 802.1X status and the 802.1X settings for all interfaces using 802.1X authentication:

```
Router# show dot1x all
Sysauthcontrol             Disabled
Dot1x Protocol Version            2
Critical Recovery Delay         100
Critical EAPOL             Disabled

Dot1x Info for GigabitEthernet3/27
-----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = FORCE_AUTHORIZED
ControlDirection          = Both
HostMode                  = SINGLE_HOST
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod           = 0

Router#
```

Displaying Authentication Methods and Status

To display the authentication methods and status, perform any of these tasks:

Command	Purpose
Router# **show authentication registrations**	Displays details of all registered methods.
Router# **show authentication interface** *interface*	Displays authentication information for a specific interface
Router# **show authentication method** *method*	Lists current authentication sessions that were authorized using the specified method.
Router# **show authentication sessions** [**handle** *handle*] [**interface** *interface*] [**mac** *mac*] [**method** *method*] [**session-id** *session-id*]	Displays information about current authentication sessions. With no options specified, all current active sessions will be listed. The keywords can be added and combined to display detailed information about specific sessions or subset of sessions.

Table 60-3 shows the possible states of the authentication session.

Table 60-3 Authentication Session States

State	Description
Idle	The session has been initialized and no methods have run yet.
Running	A method is running for this session.
No methods	No method has provided a result for this session.
Authc Success	A method has provided a successful authentication result for the session.
Authc Failed	A method has provided a failed authentication result for the session.
Authz Success	All features have been successfully applied for this session.
Authz Failed	A feature has failed to be applied for this session.

Table 60-4 shows the possible states of the authentication methods.

Table 60-4 Authentication Method States

State	Description
Not run	The method has not run for this session
Running	The method is running for this session.
Failed over	The method has failed and the next method is expected to provide a result.

Table 60-4 Authentication Method States (continued)

State	Description
Success	The method has provided a successful authentication result for the session.
Authc Failed	The method has provided a failed authentication result for the session.

This example shows how to display the registered authentication methods:

```
Router# show authentication registrations
Auth Methods registered with the Auth Manager:
  Handle  Priority  Name
    3        0       dot1x
    2        1       mab
    1        2       webauth
```

This example shows how to display the authentication details for a given interface:

```
Router# show authentication interface g1/23
Client list:
  MAC Address     Domain   Status              Handle          Interface
  0123.4567.abcd  DATA     Authz Success  0xE0000000  GigabitEthernet1/23

Available methods list:
  Handle  Priority  Name
    3        0      dot1x
    2        1      mab
Runnable methods list:
  Handle  Priority  Name
    2        0      mab
    3        1      dot1x
```

This example shows how to display all authentication sessions on the switch:

```
Router# show authentication sessions

Interface  MAC Address     Method   Domain   Status         Session ID
Gi1/48     0015.63b0.f676  dot1x    DATA     Authz Success  0A3462B1000000102983C05C
Gi1/5      000f.23c4.a401  mab      DATA     Authz Success  0A3462B10000000D24F80B58
Gi1/5      0014.bf5d.d26d  dot1x    DATA     Authz Success  0A3462B10000000E29811B94
```

This example shows how to display sessions authorized using a specified authentication method:

```
Router# show authentication method dot1x
Interface  MAC Address     Method   Domain   Status         Session ID
Gi1/48     0015.63b0.f676  dot1x    DATA     Authz Success  0A3462B1000000102983C05C
Gi1/5      0014.bf5d.d26d  dot1x    DATA     Authz Success  0A3462B10000000E29811B94
```

This example shows how to display all authentication sessions on an interface:

```
Router# show authentication sessions interface f1/47

Interface:  FastEthernet1/47
          MAC Address:  Unknown
           IP Address:  Unknown
               Status:  Authz Success
               Domain:  DATA
       Oper host mode:  multi-host
     Oper control dir:  both
        Authorized By:  Guest Vlan
          Vlan Policy:  20
```

```
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0A3462C8000000000002763C
      Acct Session ID:  0x00000002
               Handle:  0x25000000

Runnable methods list:
       Method   State
       mab      Failed over
       dot1x    Failed over

----------------------------------------

            Interface:  FastEthernet1/47
          MAC Address:  0005.5e7c.da05
           IP Address:  Unknown
            User-Name:  00055e7cda05
               Status:  Authz Success
               Domain:  VOICE
       Oper host mode:  multi-domain
     Oper control dir:  both
        Authorized By:  Authentication Server
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0A3462C8000000010002A238
      Acct Session ID:  0x00000003
               Handle:  0x91000001

Runnable methods list:
       Method   State
       mab      Authc Success
       dot1x    Not run
```

This example shows how to display the authentication session for a specified session ID:

```
Router# show authentication sessions session-id 0B0101C70000004F2ED55218

            Interface:  GigabitEthernet9/2
          MAC Address:  0000.0000.0011
           IP Address:  20.0.0.7
             Username:  johndoe
               Status:  Authz Success
               Domain:  DATA
       Oper host mode:  multi-host
     Oper control dir:  both
        Authorized By:  Critical Auth
          Vlan policy:  N/A
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0B0101C70000004F2ED55218
      Acct Session ID:  0x00000003
               Handle:  0x91000001

Runnable methods list:
       Method   State
       mab      Authc Success
       dot1x    Not run
```

This example shows how to display all clients authorized by the specified authentication method:

```
Router# show authentication sessions method mab

No Auth Manager contexts match supplied criteria
```

```
Router# show authentication sessions method dot1x

Interface  MAC Address     Domain   Status         Session ID
Gi9/2      0000.0000.0011  DATA     Authz Success  0B0101C70000004F2ED55218
```

Displaying MAC Authentication Bypass Status

To display the MAB status, perform this task:

Command	Purpose
`Router# show mab {all | interface type[1] slot/port} [detail]`	Displays MAB authentication details for all interfaces or for a specific interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Table 60-5 shows the possible states of the MAB authentication state machine.

Table 60-5 MAB States

State	Description
INITIALIZE	The authorization session is initialized.
ACQUIRING	The session is obtaining the client MAC address.
AUTHORIZING	The session is waiting for MAC-based authorization.
TERMINATE	The authorization session result has been obtained.

This example shows how to display the brief MAB status for a single interface:

```
Router# show mab interface fa1/1

MAB details for FastEthernet1/1
-------------------------------------
Mac-Auth-Bypass          = Enabled
Inactivity Timeout       = None
```

This example shows how to display the detailed MAB status for a single interface:

```
Router# show mab interface fa1/1 detail

MAB details for FastEthernet1/1
-------------------------------------
Mac-Auth-Bypass          = Enabled
Inactivity Timeout       = None

MAB Client List
---------------
Client MAC               = 000f.23c4.a401
MAB SM state             = TERMINATE
Auth Status              = AUTHORIZED
```

Enabling MAC Move

Release 12.2(33)SXI4 and later releases support the Mac move feature. To globally enable MAC move on the switch, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **authentication mac-move permit**	Enables MAC move on the switch.
Step 3	Router(config)# **end**	Returns to privileged EXEC mode.
Step 4	Router# **show running-config**	(Optional) Verifies your entries.
Step 5	Router# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to globally enable MAC move on a switch:

```
Router(config)# authentication mac-move permit
```

Enabling MAC Replace

Release 12.2(33)SXI4 and later releases support the Mac replace feature. To enable MAC replace on an interface, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **interface** *interface-id*	Specifies the port to be configured, and enter interface configuration mode.
Step 3	Router(config-if)# **authentication violation** {**protect** | **replace** | **restrict** | **shutdown**}	Uses the **replace** keyword to enable MAC replace on the interface. The port removes the current session and initiates authentication with the new host. The other keywords have these effects: • **protect**: the port drops packets with unexpected MAC addresses without generating a system message. • **restrict**: violating packets are dropped by the CPU and a system message is generated. • **shutdown**: the port is error disabled when it receives an unexpected MAC address.
Step 4	Router(config)# **end**	Returns to privileged EXEc mode.
Step 5	Router# **show running-config**	Verifies your entries.
Step 6	Router# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to enable MAC replace on an interface:

```
Router(config)# interface gigabitethernet2/2
Router(config-if)# authentication violation replace
```

Configuring Web-Based Authentication

This chapter describes how to configure web-based authentication. Cisco IOS Release 12.2(33)SXH and later releases support web-based authentication.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Web-Based Authentication

The web-based authentication feature implements web-based authentication, which is also known as Web Authentication Proxy.

You can use the web-based authentication feature to authenticate end users on host systems that do not run the IEEE 802.1X supplicant. You can configure the web-based authentication feature on Layer 2 and Layer 3 interfaces.

When a user initiates an HTTP session, the web-based authentication feature intercepts ingress HTTP packets from the host and sends an HTML login page to the user. The user keys in their credentials, which the web-based authentication feature sends to the AAA server for authentication. If the authentication succeeds, web-based authentication sends a Login-Successful HTML page to the host and applies the access policies returned by the AAA server.

If the authentication fails, web-based authentication feature sends a Login-Fail HTML page to the user, which prompts the user to retry the login attempt. If the user exceeds the maximum number of failed login attempts, web-based authentication sends a Login-Expired HTML page to the host and the user is placed on a watch list for a waiting period.

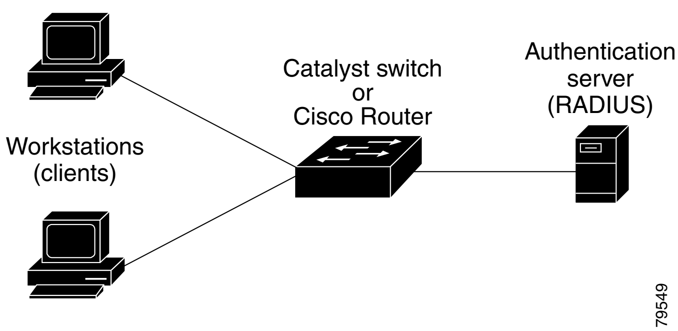
These sections describe the role of web-based authentication as a part of the authentication, authorization, and accounting (AAA) system:

- Device Roles, page 61-2
- Host Detection, page 61-3
- Session Creation, page 61-3
- Authentication Process, page 61-3
- AAA Fail Policy, page 61-4
- Customization of the Authentication Proxy Web Pages, page 61-4
- Web-based Authentication Interactions with Other Features, page 61-5

Device Roles

With web-based authentication, the devices in the network have specific roles as shown in Figure 61-1.

Figure 61-1 Web-based Authentication Device Roles



The specific roles shown in Figure 61-1 are as follows:

- *Client*—The device (workstation) that requests access to the LAN and switch services and responds to requests from the switch. The workstation must be running an HTML browser with Java Script enabled.
- *Authentication server*—Performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch whether or not the client is authorized to access the LAN and switch services.
- *Switch*—Controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client.

Host Detection

The switch maintains an IP device tracking table to store information about detected hosts.

Note By default, the IP device tracking feature is disabled on a switch. You must enable the IP device tracking feature to use web-based authentication.

For Layer 3 interfaces, web-based authentication sets an HTTP intercept ACL when the feature is configured on the interface (or when the interface is put in service).

For Layer 2 interfaces, web-based authentication detects IP hosts using the following mechanisms:

- ARP based trigger—ARP redirect ACL allows web-based authentication to detect hosts with static IP address or dynamically acquired IP address.
- Dynamic ARP Inspection
- DHCP snooping—Web-based authentication is notified when the switch creates a DHCP binding entry for the host.

Session Creation

When web-based authentication detects a new host, it creates a session as follows:

- Checks the exception list

  If the host IP is included in the exception list, the policy from the exception list entry is applied, and the session is considered to be established.
- Checks for Auth bypass

  If the host IP is not on the exception list, web-based authentication sends a nonresponsive host (NRH) request to the server.

  If the server response is Access Accepted, authorization is bypassed for this host. The session is considered to be established.
- Sets up the HTTP Intercept ACL

  If the server response to the NRH request is Access Rejected, the HTTP intercept ACL is activated and the session waits for HTTP traffic from the host.

Authentication Process

When web-based authentication is enabled, the following events occur:

- The user initiates an HTTP session.
- The HTTP traffic is intercepted, and authorization is initiated. The switch sends the login page to the user. The user enters a username and password on the login page, and the switch sends the entries to the authentication server.
- If the client identity is valid and the authentication succeeds, the switch downloads and activates the user's access policy from the authentication server. The login success page is sent to the user.
- If the authentication fails, the switch sends the login fail page. The user retries the login, but if the maximum number of attempts fail, the switch sends the login expired page and the host is placed in a watch list. After a watch list timeout, the user can retry the authentication process.

- If the authentication server does not respond to the switch, and if an AAA fail policy is configured, the switch will apply the failure access policy to the host. The login success page is sent to the user. The AAA fail policy feature is available in Cisco IOS Release 12.2(33)SXI and later releases.

The switch reauthenticates a client when the host does not respond to an ARP probe on a Layer 2 interface, or the host does not send any traffic within the idle timeout on a Layer 3 interface.

- The feature applies the downloaded timeout or the locally configured session timeout.
- If the terminate action is RADIUS, the feature sends a nonresponsive host (NRH) request to the server. The terminate action is included in the response from the server.
- If the terminate action is default, the session is dismantled and the applied policy is removed.

AAA Fail Policy

The AAA fail policy, supported in Cisco IOS Release 12.2(33)SXI and later releases, is a method for allowing a user to connect or to remain connected to the network if the AAA server is not available. If the AAA server cannot be reached when web-based authentication of a client is needed, instead of rejecting the user (that is, not providing the access to the network), an administrator can configure a default AAA fail policy that can be applied to the user.

This policy is advantageous for the following reasons:

- While AAA is unavailable, the user will still have connectivity to the network, although access may be restricted.
- When the AAA server is again available, a user can be revalidated, and the user's normal access policies can be downloaded from the AAA server.

Note When the AAA server is down, the AAA fail policy is applied only if there is no existing policy associated with the user. Typically, if the AAA server is unavailable when a user session requires reauthentication, the policies currently in effect for the user are retained.

While the AAA fail policy is in effect, the session state is maintained as AAA Down.

Customization of the Authentication Proxy Web Pages

The switch's internal HTTP server hosts four HTML pages for delivery to an authenticating client during the web-based authentication process. The four pages allow the server to notify the user of the following four states of the authentication process:

- Login—The user's credentials are requested
- Success—The login was successful
- Fail—The login has failed
- Expire—The login session has expired due to excessive login failures

In Cisco IOS Release 12.2(33)SXI and later releases, you can substitute your custom HTML pages for the four default internal HTML pages, or you can specify a URL to which the user will be redirected upon successful authentication, effectively replacing the internal Success page.

Web-based Authentication Interactions with Other Features

These sections describe web-based authentication interactions with these features:

Port Security

You can configure web-based authentication and port security on the same port. (You configure port security on the port by using the **switchport port-security** interface configuration command.) When you enable port security and web-based authentication on a port, web-based authentication authenticates the port, and port security manages network access for all MAC addresses, including that of the client. You can then limit the number or group of clients that can access the network through the port.

For more information about enabling port security, see the "Configuring Port Security" section on page 62-5.

LAN Port IP

You can configure LAN port IP (LPIP) and Layer 2 web-based authentication on the same port. The host is authenticated using web-based authentication first, and then LPIP posture validation takes place. The LPIP host policy overrides the web-based authentication host policy.

If the web-based authentication idle timer expires, the NAC policy is removed. The host is authenticated and posture validated again.

Gateway IP

You cannot configure Gateway IP on a Layer 3 VLAN interface if web-based authentication is configured on any of the switch ports in the VLAN.

You can configure web-based authentication on the same Layer 3 interface as Gateway IP. The host policies for both features are applied in software. The GWIP policy overrides the web-based authentication host policy.

ACLs

If you configure a VLAN ACL or Cisco IOS ACL on an interface, the ACL is applied to the host traffic only after the web-based authentication host policy is applied.

For Layer 2 web-based authentication, you must configure a port ACL (PACL) as the default access policy for ingress traffic from hosts connected to the port. After authentication, the web-based authentication host policy overrides the PACL.

You cannot configure a MAC ACL and web-based authentication on the same interface.

You cannot configure web-based authentication on a port whose access VLAN has VACL capture configured.

IP Source Guard

In releases earlier than Cisco IOS Release 12.2(33)SXI2, configuring IP Source Guard and web-based authentication on the same interface is not supported.

In Cisco IOS Release 12.2(33)SXI2 and later releases, you can configure IP Source Guard and web-based authentication on the same interface. If DHCP snooping is also enabled on the access VLAN, you must enter the **mls acl tcam override dynamic dhcp-snooping** command in global configuration mode to avoid conflict between the two features. Other VLAN-based features are not supported when IP Source Guard and web-based authentication are combined.

EtherChannel

You can configure web-based authentication on a Layer 2 EtherChannel interface. The web-based authentication configuration applies to all member channels.

Switchover

On Catalyst 6500 series switches with redundant supervisor engines in RPR mode redundancy, information about currently authenticated hosts is maintained during a switchover. Users will not need to reauthenticate.

Configuring Web-Based Authentication

These sections describe how to configure web-based authentication:

Default Web-Based Authentication Configuration

Table 61-1 shows the default web-based authentication configuration.

Table 61-1 Default Web-based Authentication Configuration

Feature	Default Setting
AAA	Disabled
RADIUS server • IP address • UDP authentication port • Key	 • None specified • 1812 • None specified
Default value of inactivity timeout	3600 seconds
Inactivity timeout	Enabled

Web-based Authentication Configuration Guidelines and Restrictions

These are the web-based authentication configuration guidelines:

- Web-based authentication is an ingress-only feature.
- You can configure web-based authentication only on access ports. Web-based authentication is not supported on trunk ports, EtherChannel member ports, or dynamic trunk ports.
- You must configure the default ACL on the interface before configuring web-based authentication. Configure a port ACL for a Layer 2 interface, or a Cisco IOS ACL for a Layer 3 interface.
- On Layer 2 interfaces, you cannot authenticate hosts with static ARP cache assignment. These hosts are not detected by the web-based authentication feature, because they do not send ARP messages.
- By default, the IP device tracking feature is disabled on a switch. You must enable the IP device tracking feature to use web-based authentication.
- You must configure at least one IP address to run the HTTP server on the switch. You must also configure routes to reach each host IP address. The HTTP server sends the HTTP login page to the host.
- Hosts that are more than one hop away may experience traffic disruption if an STP topology change results in the host traffic arriving on a different port. This is because ARP and DHCP updates may not be sent after a Layer 2 (STP) topology change.
- Web-based authentication does not support VLAN assignment as a downloadable host policy.
- Cisco IOS Release 12.2(33)SXI and later releases support downloadable ACLs (DACLs) from the RADIUS server.
- Web-based authentication is not supported for IPv6 traffic.

Web-based Authentication Configuration Task List

To configure the web-based authentication feature, perform the following tasks:

- Configuring the Authentication Rule and Interfaces, page 61-8
- Configuring AAA Authentication, page 61-9
- Configuring Switch-to-RADIUS-Server Communication, page 61-9
- Configuring the HTTP Server, page 61-11
- Configuring an AAA Fail Policy, page 61-13
- Configuring the Web-based Authentication Parameters, page 61-14
- Removing Web-based Authentication Cache Entries, page 61-15

Configuring the Authentication Rule and Interfaces

To configure web-based authentication, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip admission name name proxy http`	Configures an authentication rule for web-based authorization.
Step 2	`Router(config)# interface type[1] slot/port`	Enters interface configuration mode and specifies the ingress Layer 2 or Layer 3 interface to be enabled for web-based authentication.
Step 3	`Router(config-if)# ip access-group name`	Applies the default ACL.
Step 4	`Router(config-if)# ip admission name`	Configures web-based authentication on the specified interface.
Step 5	`Router(config-if)# authentication order method1 [method2] [method3]`	(Optional) Specifies the fallback order of authentication methods to be used. The three values of *method*, in the default order, are **dot1x**, **mab**, and **webauth**. Omitting a method disables that method on the interface.
Step 6	`Router(config-if)# exit`	Returns to configuration mode.
Step 7	`Router(config)# ip device tracking`	Enables the IP device tracking table.
Step 8	`Router(config)# end`	Returns to privileged EXEC mode.
Step 9	`Router# show ip admission configuration`	Displays the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable web-based authentication, while disabling 802.1X or MAB authentication, on Fast Ethernet port 5/1:

```
Router(config)# ip admission name webauth1 proxy http
Router(config)# interface fastethernet 5/1
Router(config-if)# ip admission webauth1
Router(config-if)# authentication order webauth
Router(config-if)# exit
Router(config)# ip device tracking
```

This example shows how to verify the configuration:

```
Router# show ip admission configuration
Authentication Proxy Banner not configured
Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication global init state time is 2 minutes
Authentication Proxy Watch-list is disabled

Authentication Proxy Rule Configuration
 Auth-proxy name webauth1
    http list not specified inactivity-time 60 minutes

Authentication Proxy Auditing is disabled
Max Login attempts per user is 5
```

Configuring AAA Authentication

To enable web-based authentication, you must enable AAA and specify the authentication method. perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`aaa new-model`**	Enables AAA functionality.
Step 2	`Router(config)#` **`aaa authentication login default group`** `{`*`tacacs+`* `|` *`radius`*`}`	Defines the list of authentication methods at login.
Step 3	`Router(config)#` **`aaa authorization auth-proxy default group`** `{`*`tacacs+`* `|` *`radius`*`}`	Creates an authorization method list for web-based authorization.
Step 4	`Router(config)#` **`tacacs-server host`** `{`*`hostname`* `|` *`ip_address`*`}`	Specifies an AAA server. For Radius servers, see the section "Configuring Switch-to-RADIUS-Server Communication" section on page 61-9.
Step 5	`Router(config)#` **`tacacs-server key`** `{`*`key-data`*`}`	Configures the authorization and encryption key used between the switch and the TACACS server.

This example shows how to enable AAA:

```
Router(config)# aaa new-model
Router(config)# aaa authentication login default group tacacs+
Router(config)# aaa authorization auth-proxy default group tacacs+
```

Configuring Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by any of the following:

- Host name
- Host IP address
- Host name and specific UDP port numbers
- IP address and specific UDP port numbers

The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service (for example, authentication) the second host entry that is configured functions as the failover backup to the first one. The RADIUS host entries are chosen in the order that they were configured.

To configure the RADIUS server parameters, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip radius source-interface`** *`interface_name`*	Specifies that the RADIUS packets have the IP address of the indicated interface.
Step 2	`Router(config)#` **`radius-server host`** `{`*`hostname`* `|` *`ip-address`*`}` **`test username`** *`username`*	Specifies the host name or IP address of the remote RADIUS server. The **test username** *username* option enables automated testing of the RADIUS server connection. The specified *username* does not need to be a valid user name. The **key** option specifies an authentication and encryption key to be used between the switch and the RADIUS server. To use multiple RADIUS servers, reenter this command.
Step 3	`Router(config)#` **`radius-server key`** *`string`*	Configures the authorization and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 4	`Router(config)#` **`radius-server vsa send authentication`**	Enables downloading of an ACL from the RADIUS server. This feature is supported in Cisco IOS Release 12.2(33)SXI and later releases.
Step 5	`Router(config)#` **`radius-server dead-criteria tries`** *`num-tries`*	Specifies the number of unanswered transmits to a RADIUS server before considering the server to be dead. The range of *num-tries* is 1 to 100.

When you configure the RADIUS server parameters, note the following information:

- Specify the **key** *string* on a separate command line.
- For **key** *string*, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server.
- When you specify the **key** *string*, spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon.
- You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius-server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and the **radius-server key** global configuration commands. For more information, see the *Cisco IOS Security Configuration Guide*, Release 12.2, publication and the *Cisco IOS Security Command Reference*, Release 12.2, publication at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/command/reference/fsecur_r.html
- Cisco IOS Release 12.2(33)SXI and later releases support downloadable ACLs (DACLs) from the RADIUS server.

Note You also need to configure some settings on the RADIUS server. These settings include the IP address of the switch, the key string to be shared by both the server and the switch, and the downloadable ACL. For more information, see the RADIUS server documentation.

This example shows how to configure the RADIUS server parameters on the switch:

```
Router(config)# ip radius source-interface Vlan80
Router(config)# radius-server host 172.120.39.46 test username user1
Router(config)# radius-server key rad123
Router(config)# radius-server dead-criteria tries 2
```

Configuring the HTTP Server

To use web-based authentication, you must enable the HTTP server within the switch. You can enable the server for either HTTP or HTTPS. To enable the server, perform one of these tasks in global configuration mode:

Command	Purpose
`Router(config)# ip http server`	Enables the HTTP server. The web-based authentication feature uses the HTTP server to communicate with the hosts for user authentication.
`Router(config)# ip http secure-server`	Enables HTTPS.

With Cisco IOS Release 12.2(33)SXI and later releases, you can optionally configure custom authentication proxy web pages or specify a redirection URL for successful login, as described in the following sections:

- Customizing the Authentication Proxy Web Pages
- Specifying a Redirection URL for Successful Login

Customizing the Authentication Proxy Web Pages

With Cisco IOS Release 12.2(33)SXI and later releases, you have the option to provide four substitute HTML pages to be displayed to the user in place of the switch's internal default HTML pages during web-based authentication.

To specify the use of your custom authentication proxy web pages, first store your custom HTML files on the switch's internal disk or flash memory, then perform this task in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip admission proxy http login page file device:login-filename`	Specifies the location in the switch memory file system of the custom HTML file to be used in place of the default login page. The *device:* is either disk or flash memory, such as disk0:.
Step 2	`Router(config)# ip admission proxy http success page file device:success-filename`	Specifies the location of the custom HTML file to be used in place of the default login success page.

	Command	Purpose
Step 3	Router(config)# **ip admission proxy http failure page file** *device:fail-filename*	Specifies the location of the custom HTML file to be used in place of the default login failure page.
Step 4	Router(config)# **ip admission proxy http login expired page file** *device:expired-filename*	Specifies the location of the custom HTML file to be used in place of the default login expired page.

When configuring the use of customized authentication proxy web pages, consider the following guidelines:

- To enable the custom web pages feature, you must specify all four custom HTML files. If fewer than four files are specified, the internal default HTML pages will be used.
- The four custom HTML files must be present on the disk or flash of the switch.
- An image file has a size limit of 256 KB.
- All image files must have a filename that begins with "web_auth_" (like "web_auth_logo.jpg" instead of "logo.jpg").
- All image file names must be less than 33 characters.
- Any images on the custom pages must be located on an accessible HTTP server. An intercept ACL must be configured within the admission rule to allow access to the HTTP server.
- Any external link from a custom page will require configuration of an intercept ACL within the admission rule.
- Any name resolution required for external links or images will require configuration of an intercept ACL within the admission rule to access a valid DNS server.
- If the custom web pages feature is enabled, a configured auth-proxy-banner will not be used.
- If the custom web pages feature is enabled, the redirection URL for successful login feature will not be available.
- To remove the specification of a custom file, use the **no** form of the command.

Because the custom login page is a public web form, consider the following guidelines for this page:

- The login form must accept user input for the username and password and must POST the data as **uname** and **pwd**.
- The custom login page should follow best practices for a web form, such as page timeout, hidden password, and prevention of redundant submissions.

The following example shows how to configure custom authentication proxy web pages:

```
Router(config)# ip admission proxy http login page file disk1:login.htm
Router(config)# ip admission proxy http success page file disk1:success.htm
Router(config)# ip admission proxy http fail page file disk1:fail.htm
Router(config)# ip admission proxy http login expired page file disk1:expired.htm
```

The following example shows how to verify the configuration of custom authentication proxy web pages:

```
Router# show ip admission configuration

Authentication proxy webpage
 Login page           : disk1:login.htm
 Success page         : disk1:success.htm
 Fail Page            : disk1:fail.htm
 Login expired Page   : disk1:expired.htm

Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
```

```
Authentication global init state time is 2 minutes
Authentication Proxy Session ratelimit is 100
Authentication Proxy Watch-list is disabled
Authentication Proxy Auditing is disabled
Max Login attempts per user is 5
```

Specifying a Redirection URL for Successful Login

With Cisco IOS Release 12.2(33)SXI and later releases, you have the option to specify a URL to which the user will be redirected upon successful authentication, effectively replacing the internal Success HTML page.

To specify a redirection URL for successful login, perform this task in global configuration mode:

Command	Purpose
Router(config)# **ip admission proxy http success redirect** *url-string*	Specifies a URL for redirection of the user in place of the default login success page.

When configuring a redirection URL for successful login, consider the following guidelines:

- If the custom authentication proxy web pages feature is enabled, the redirection URL feature is disabled and will not be available in the CLI. You can perform redirection in the custom login success page.
- If the redirection URL feature is enabled, a configured auth-proxy-banner will not be used.
- To remove the specification of a redirection URL, use the **no** form of the command.

The following example shows how to configure a redirection URL for successful login:

```
Router(config)# ip admission proxy http success redirect www.cisco.com
```

The following example shows how to verify the redirection URL for successful login:

```
Router# show ip admission configuration

Authentication Proxy Banner not configured
Customizable Authentication Proxy webpage not configured
HTTP Authentication success redirect to URL: http://www.cisco.com
Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication global init state time is 2 minutes
Authentication Proxy Watch-list is disabled
Authentication Proxy Max HTTP process is 7
Authentication Proxy Auditing is disabled
Max Login attempts per user is 5
```

Configuring an AAA Fail Policy

The AAA fail policy for web-based authentication is supported in Cisco IOS Release 12.2(33)SXI and later releases.

To configure an AAA fail policy, perform this task in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`ip admission name`** *`rule-name`* **`proxy http event timeout aaa policy identity`** *`identity_policy_name`*	Creates an AAA fail rule and associates an identity policy to be applied to sessions when the AAA server is unreachable. To remove the rule on the switch, use the **no ip admission name** *rule-name* **proxy http event timeout aaa policy identity** global configuration command.
Step 2	`Router(config)#` **`ip admission ratelimit aaa-down`** *`number_of_sessions`*	(Optional) To avoid flooding the AAA server when it returns to service, you can rate limit the authentication attempts from hosts in the AAA Down state.

The following example shows how to apply an AAA fail policy:

```
Router(config)# ip admission name AAA_FAIL_POLICY proxy http event timeout aaa policy
identity GLOBAL_POLICY1
```

The following example shows how to determine whether any hosts are connected in the AAA Down state:

```
Router# show ip admission cache
Authentication Proxy Cache
 Client IP 209.165.201.11 Port 0, timeout 60, state ESTAB (AAA Down)
```

The following example shows how to view detailed information about a particular session based on the host IP address:

```
Router# show ip admission cache 209.165.201.11
Address              : 209.165.201.11
MAC Address          : 0000.0000.0000
Interface            : Vlan333
Port                 : 3999
Timeout              : 60
Age                  : 1
State                : AAA Down
AAA Down policy      : AAA_FAIL_POLICY
```

Configuring the Web-based Authentication Parameters

You can configure the maximum number of failed login attempts before the client is placed in a watch list for a waiting period.

To configure the web-based authentication parameters, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip admission max-login-attempts`** *`number`*	Sets the maximum number of failed login attempts. The range is 1 to 2147483647 attempts; the default is 5.
Step 2	`Router(config)#` **`end`**	Returns to privileged EXEC mode.
Step 3	`Router#` **`show ip admission configuration`**	Displays the authentication proxy configuration.
Step 4	`Router#` **`show ip admission cache`**	Displays the list of authentication entries.

This example shows how to set the maximum number of failed login attempts to 10:

```
Router(config)# ip admission max-login-attempts 10
```

Removing Web-based Authentication Cache Entries

To delete existing session entries, perform either of these tasks:

Command	Purpose
Router# **clear ip auth-proxy cache** {* | *host ip address*}	Deletes authentication proxy entries. Use an asterisk to delete all cache entries. Enter a specific IP address to delete the entry for a single host.
Router# **clear ip admission cache** {* | *host ip address*}	Deletes authentication proxy entries. Use an asterisk to delete all cache entries. Enter a specific IP address to delete the entry for a single host.

This example shows how to remove the web-based authentication session for the client at a specific IP address:

```
Router# clear ip auth-proxy cache 209.165.201.1
```

Displaying Web-Based Authentication Status

To display the web-based authentication settings for all interfaces or for specific ports, perform this task:

	Command	Purpose
Step 1	Router# **show fm ip-admission l2http** [**all** | **interface** *type*[1] *slot/port*]	Displays the web-based authentication settings. (Optional) Use the **all** keyword to display the settings for all interfaces using web-based authentication. (Optional) Use the **interface** keyword to display the web-based authentication settings for a specific interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to view only the global web-based authentication status:

```
Router# show fm ip-admission l2http all
```

This example shows how to view the web-based authentication settings for interface GigabitEthernet 3/27:

```
Router# show fm ip-admission l2http interface gigabitethernet 3/27
```

For detailed information about the fields in these displays, see the Cisco IOS Master Command List.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Port Security

This chapter describes how to configure the port security feature.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

---

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Port Security

These sections describe port security:

Port Security with Dynamically Learned and Static MAC Addresses

You can use port security with dynamically learned and static MAC addresses to restrict a port's ingress traffic by limiting the MAC addresses that are allowed to send traffic into the port. When you assign secure MAC addresses to a secure port, the port does not forward ingress traffic that has source addresses outside the group of defined addresses. If you limit the number of secure MAC addresses to one and assign a single secure MAC address, the device attached to that port has the full bandwidth of the port.

A security violation occurs in either of these situations:

- When the maximum number of secure MAC addresses is reached on a secure port and the source MAC address of the ingress traffic is different from any of the identified secure MAC addresses, port security applies the configured violation mode.
- If traffic with a secure MAC address that is configured or learned on one secure port attempts to access another secure port in the same VLAN, applies the configured violation mode.

Note After a secure MAC address is configured or learned on one secure port, the sequence of events that occurs when port security detects that secure MAC address on a different port in the same VLAN is known as a MAC move violation.

See the "Configuring the Port Security Violation Mode on a Port" section on page 62-6 for more information about the violation modes.

After you have set the maximum number of secure MAC addresses on a port, port security includes the secure addresses in the address table in one of these ways:

- You can statically configure all secure MAC addresses by using the **switchport port-security mac-address** *mac_address* interface configuration command.
- You can allow the port to dynamically configure secure MAC addresses with the MAC addresses of connected devices.
- You can statically configure a number of addresses and allow the rest to be dynamically configured.

If the port has a link-down condition, all dynamically learned addresses are removed.

Following bootup, a reload, or a link-down condition, port security does not populate the address table with dynamically learned MAC addresses until the port receives ingress traffic.

A security violation occurs if the maximum number of secure MAC addresses have been added to the address table and the port receives traffic from a MAC address that is not in the address table.

You can configure the port for one of three violation modes: protect, restrict, or shutdown. See the "Configuring Port Security" section on page 62-5.

To ensure that an attached device has the full bandwidth of the port, set the maximum number of addresses to one and configure the MAC address of the attached device.

Port Security with Sticky MAC Addresses

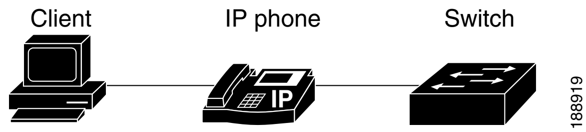
Port security with sticky MAC addresses provides many of the same benefits as port security with static MAC addresses, but sticky MAC addresses can be learned dynamically. Port security with sticky MAC addresses retains dynamically learned MAC addresses during a link-down condition.

If you enter a **write memory** or **copy running-config startup-config** command, then port security with sticky MAC addresses saves dynamically learned MAC addresses in the startup-config file and the port does not have to learn addresses from ingress traffic after bootup or a restart.

Port Security with IP Phones

Figure 62-1 shows an application in which a device connects to the switch through the data port of an IP phone.

Figure 62-1 Device Connected Through IP Phone



Because the device is not directly connected to the switch, the switch cannot physically detect a loss of port link if the device is disconnected. Later Cisco IP phones send a Cisco Discovery Protocol (CDP) host presence type length value (TLV) to notify the switch of changes in the attached device's port link state. With Cisco IOS Release 12.2(33)SXI and later releases, the switch recognizes the host presence TLV. Upon receiving a host presence TLV notification of a link down on the IP phone's data port, port security removes from the address table all static, sticky, and dynamically learned MAC addresses. The removed addresses are added again only when the addresses are learned dynamically or configured.

Default Port Security Configuration

Table 62-1 shows the default port security configuration for an interface.

Table 62-1 Default Port Security Configuration

Feature	Default Setting
Port security	Disabled.
Maximum number of secure MAC addresses	1.
Violation mode	Shutdown. The port shuts down when the maximum number of secure MAC addresses is exceeded, and an SNMP trap notification is sent.

Port Security Guidelines and Restrictions

When configuring port security, follow these guidelines:

- With the default port security configuration, to bring all secure ports out of the error-disabled state, enter the **errdisable recovery cause psecure-violation** global configuration command, or manually reenable the port by entering the **shutdown** and **no shut down** interface configuration commands.
- Enter the **clear port-security dynamic** global configuration command to clear all dynamically learned secure addresses. See the Cisco IOS Master Command List for complete syntax information.

- Port security learns unauthorized MAC addresses with a bit set that causes traffic to them or from them to be dropped. The **show mac-address-table** command displays the unauthorized MAC addresses, but does not display the state of the bit. (CSCeb76844)
- To preserve dynamically learned sticky MAC addresses and configure them on a port following a bootup or a reload and after the dynamically learned sticky MAC addresses have been learned, you must enter a **write memory** or **copy running-config startup-config** command to save them in the startup-config file.
- Port security supports private VLAN (PVLAN) ports.
- Port security supports IEEE 802.1Q tunnel ports.
- Port security does not support Switch Port Analyzer (SPAN) destination ports.
- Port security does not support EtherChannel port-channel interfaces.
- With Cisco IOS Release 12.2(33)SXH and later releases, you can configure port security and 802.1X port-based authentication on the same port. With releases earlier than Cisco IOS Release 12.2(33)SXH:
  - If you try to enable 802.1X port-based authentication on a secure port, an error message appears and 802.1X port-based authentication is not enabled on the port.
  - If you try to enable port security on a port configured for 802.1X port-based authentication, an error message appears and port security is not enabled on the port.
- Port security supports nonnegotiating trunks.
  - Port security only supports trunks configured with these commands:

    switchport
    switchport trunk encapsulation
    switchport mode trunk
    switchport nonegotiate
  - If you reconfigure a secure access port as a trunk, port security converts all the sticky and static secure addresses on that port that were dynamically learned in the access VLAN to sticky or static secure addresses on the native VLAN of the trunk. Port security removes all secure addresses on the voice VLAN of the access port.
  - If you reconfigure a secure trunk as an access port, port security converts all sticky and static addresses learned on the native VLAN to addresses learned on the access VLAN of the access port. Port security removes all addresses learned on VLANs other than the native VLAN.

Note Port security uses the VLAN ID configured with the **switchport trunk native vlan** command for both IEEE 802.1Q trunks and ISL trunks.

- Take care when you enable port security on the ports connected to the adjacent switches when there are redundant links running between the switches because port security might error-disable the ports due to port security violations.
- Flex Links and port security are not compatible with each other.

Configuring Port Security

These sections describe how to configure port security:

Enabling Port Security

These sections describe how to enable port security:

Enabling Port Security on a Trunk

Port security supports nonnegotiating trunks.

Caution Because the default number of secure addresses is one and the default violation action is to shut down the port, configure the maximum number of secure MAC addresses on the port before you enable port security on a trunk (see "Configuring the Maximum Number of Secure MAC Addresses on a Port" section on page 62-9).

To enable port security on a trunk, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`switchport`**	Configures the port as a Layer 2 port.
Step 3	`Router(config-if)#` **`switchport trunk encapsulation`** `{`**`isl`** `|` **`dot1q`**`}`	Configures the encapsulation, which configures the Layer 2 switching port as either an ISL or 802.1Q trunk.
Step 4	`Router(config-if)#` **`switchport mode trunk`**	Configures the port to trunk unconditionally.
Step 5	`Router(config-if)#` **`switchport nonegotiate`**	Configures the trunk not to use DTP.
Step 6	`Router(config-if)#` **`switchport port-security`**	Enables port security on the trunk.
Step 7	`Router(config-if)#` **`do show port-security interface`** *`type`*[1] *`slot/port`* `|` **`include Port Security`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure Fast Ethernet port 5/36 as a nonnegotiating trunk and enable port security:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# switchport
Router(config-if)# switchport mode trunk
Router(config-if)# switchport nonegotiate
Router(config-if)# switchport port-security
Router(config-if)# do show port-security interface fastethernet 5/36 | include Port
Security
Port Security              : Enabled
```

Enabling Port Security on an Access Port

To enable port security on an access port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure. **Note** The port can be a tunnel port or a PVLAN port.
Step 2	`Router(config-if)#` **`switchport`**	Configures the port as a Layer 2 port.
Step 3	`Router(config-if)#` **`switchport mode access`**	Configures the port as a Layer 2 access port. **Note** A port in the default mode (dynamic desirable) cannot be configured as a secure port.
Step 4	`Router(config-if)#` **`switchport port-security`**	Enables port security on the port.
Step 5	`Router(config-if)#` **`do show port-security interface`** *`type`*[1] *`slot/port`* **`| include Port Security`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to enable port security on Fast Ethernet port 5/12:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# switchport port-security
Router(config-if)# do show port-security interface fastethernet 5/12 | include Port
Security
Port Security              : Enabled
```

Configuring the Port Security Violation Mode on a Port

To configure the port security violation mode on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.

	Command	Purpose
Step 2	`Router(config-if)#` **switchport port-security violation** {**protect** | **restrict** | **shutdown**}	(Optional) Sets the violation mode and the action to be taken when a security violation is detected.
Step 3	`Router(config-if)#` **do show port-security interface** *type*[1] *slot/port* | **include** *violation_mode*[2]	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**
2. *violation_mode* = **protect**, **restrict**, or **shutdown**

When configuring port security violation modes, note the following information:

- **protect**—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value.
- **restrict**—Drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value and causes the SecurityViolation counter to increment.
- **shutdown**—Puts the interface into the error-disabled state immediately and sends an SNMP trap notification.

Note
- To bring a secure port out of the error-disabled state, enter the **errdisable recovery cause** *violation_mode* global configuration command, or you can manually reenable it by entering the **shutdown** and **no shut down** interface configuration commands.
- To protect the CPU against overutilization, when you configure the protect or restrict violation modes, configure the packet drop rate limiter (see the "Configuring the Port Security Rate Limiter" section on page 62-7).

This example shows how to configure the protect security violation mode on Fast Ethernet port 5/12:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/12
Router(config-if)# switchport port-security violation protect
Router(config-if)# do show port-security interface fastethernet 5/12 | include Protect
Violation Mode             : Protect
```

This example shows how to configure the restrict security violation mode on Fast Ethernet port 5/12:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/12
Router(config-if)# switchport port-security violation restrict
Router(config-if)# do show port-security interface fastethernet 5/12 | include Restrict
Violation Mode             : Restrict
```

Configuring the Port Security Rate Limiter

Note The truncated switching mode does not support the port security rate limiter.

Port security examines all traffic received by secure ports to detect violations or to recognize and secure new MAC addresses. When the shutdown violation mode is configured, traffic cannot enter the secure port after a violation has been detected, which removes the possibility that violations might cause excessive CPU load.

When the protect or restrict violation modes are configured, port security continues to process traffic after a violation occurs, which might cause excessive CPU load. Configure the port security rate limiter to protect the CPU against excessive load when the protect or restrict violation modes are configured.

To configure the port security rate limiter, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`mls rate-limit layer2 port-security`** *`rate_in_pps`* `[`*`burst_size`*`]`	Configures the port security rate limiter.
Step 2	`Router(config)#` **`do show mls rate-limit	include PORTSEC`**	Verifies the configuration.

When configuring the port security rate limiter, note the following information:

- For the *rate_in_pps* value:
  - The range is 10 through 1,000,000 (entered as 1000000).
  - There is no default value.
  - The lower the value, the more the CPU is protected. The rate limiter is applied to traffic both before and after a security violation occurs. Configure a value high enough to permit nonviolating traffic to reach the port security feature.
  - Values lower than 1,000 (entered as 1000) should offer sufficient protection.
- For the *burst_size* value:
  - The range is 1 through 255.
  - The default is 10.
  - The default value should provide sufficient protection.

This example shows how to configure the port security rate limiter:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls rate-limit layer2 port-security 1000
Router(config)# end
```

This example shows how to verify the configuration:

```
Router# show mls rate-limit | include PORTSEC
LAYER_2 PORTSEC   On               1000       1  Not sharing
```

Configuring the Maximum Number of Secure MAC Addresses on a Port

To configure the maximum number of secure MAC addresses on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`switchport port-security maximum`** *`number_of_addresses`* **`vlan`** {*`vlan_ID`* | *`vlan_range`*}	Sets the maximum number of secure MAC addresses for the port (default is 1). **Note** Per-VLAN configuration is supported only on trunks.
Step 3	`Router(config-if)#` **`do show port-security interface`** *`type`*[1] *`slot/port`* **`| include Maximum`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring the maximum number of secure MAC addresses on a port, note the following information:

- The range for *number_of_addresses* is 1 to 4,097.
- Port security supports trunks.
  - On a trunk, you can configure the maximum number of secure MAC addresses both on the trunk and for all the VLANs on the trunk.
  - You can configure the maximum number of secure MAC addresses on a single VLAN or a range of VLANs.
  - For a range of VLANs, enter a dash-separated pair of VLAN numbers.
  - You can enter a comma-separated list of VLAN numbers and dash-separated pairs of VLAN numbers.

This example shows how to configure a maximum of 64 secure MAC addresses on Fast Ethernet port 5/12:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/12
Router(config-if)# switchport port-security maximum 64
Router(config-if)# do show port-security interface fastethernet 5/12 | include Maximum
Maximum MAC Addresses      : 64
```

Enabling Port Security with Sticky MAC Addresses on a Port

To enable port security with sticky MAC addresses on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`switchport port-security mac-address sticky`**	Enables port security with sticky MAC addresses on a port.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When enabling port security with sticky MAC addresses, note the following information:

- When you enter the **switchport port-security mac-address sticky** command:
  - All dynamically learned secure MAC addresses on the port are converted to sticky secure MAC addresses.
  - Static secure MAC addresses are not converted to sticky MAC addresses.
  - Secure MAC addresses dynamically learned in a voice VLAN are not converted to sticky MAC addresses.
  - New dynamically learned secure MAC addresses are sticky.
- When you enter the **no switchport port-security mac-address sticky** command, all sticky secure MAC addresses on the port are converted to dynamic secure MAC addresses.
- To preserve dynamically learned sticky MAC addresses and configure them on a port following a bootup or a reload, after the dynamically learned sticky MAC addresses have been learned, you must enter a **write memory** or **copy running-config startup-config** command to save them in the startup-config file.

This example shows how to enable port security with sticky MAC addresses on Fast Ethernet port 5/12:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport port-security mac-address sticky
```

Configuring a Static Secure MAC Address on a Port

To configure a static secure MAC address on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`switchport port-security mac-address`** `[`**`sticky`**`]` *`mac_address`* `[`**`vlan`** *`vlan_ID`*`]`	Configures a static MAC address as secure on the port. **Note** Per-VLAN configuration is supported only on trunks.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Router#` **`show port-security address`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring a static secure MAC address on a port, note the following information:

- You can configure sticky secure MAC addresses if port security with sticky MAC addresses is enabled (see the "Enabling Port Security with Sticky MAC Addresses on a Port" section on page 62-9).
- The maximum number of secure MAC addresses on the port, configured with the **switchport port-security maximum** command, defines how many secure MAC addresses you can configure.
- If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are learned dynamically.
- Port security is supported on trunks.
  - On a trunk, you can configure a static secure MAC address in a VLAN.

- On a trunk, if you do not configure a VLAN for a static secure MAC address, it is secure in the VLAN configured with the **switchport trunk native vlan** command.

This example shows how to configure a MAC address 1000.2000.3000 as secure on Fast Ethernet port 5/12 and verify the configuration:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport port-security mac-address 1000.2000.3000
Router(config-if)# end
Router# show port-security address
          Secure Mac Address Table
------------------------------------------------------------

Vlan    Mac Address       Type                Ports
----    -----------       ----                -----
   1    1000.2000.3000    SecureConfigured     Fa5/12
```

Configuring Secure MAC Address Aging on a Port

When the aging type is configured with the **absolute** keyword, all the dynamically learned secure addresses age out when the aging time expires. When the aging type is configured with the **inactivity** keyword, the aging time defines the period of inactivity after which all the dynamically learned secure addresses age out.

Note Static secure MAC addresses and sticky secure MAC addresses do not age out.

These sections describe how to configure secure MAC address aging on a port:

- Configuring the Secure MAC Address Aging Type on a Port, page 62-11
- Configuring Secure MAC Address Aging Time on a Port, page 62-12

Configuring the Secure MAC Address Aging Type on a Port

You can configure the secure MAC address aging type on a port. To configure the secure MAC address aging type on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`switchport port-security aging type`** `{`**`absolute`** `|` **`inactivity`**`}`	Configures the secure MAC address aging type on the port (default is absolute).
Step 3	`Router(config-if)#` **`do show port-security interface`** *`type`*[1] *`slot/port`* **`| include Time`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to set the aging type to inactivity on Fast Ethernet port 5/12:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport port-security aging type inactivity
Router(config-if)# do show port-security interface fastethernet 5/12 | include Type
Aging Type                 : Inactivity
```

Configuring Secure MAC Address Aging Time on a Port

To configure the secure MAC address aging time on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** *type*[1] *slot/port*	Selects the LAN port to configure.
Step 2	Router(config-if)# **switchport port-security aging time** *aging_time*	Configures the secure MAC address aging time on the port. The *aging_time* range is 1 to 1440 minutes (default is 0).
Step 3	Router(config-if)# **do show port-security interface** *type*[1] *slot/port* **| include Time**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure 2 hours (120 minutes) as the secure MAC address aging time on Fast Ethernet port 5/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/1
Router(config-if)# switchport port-security aging time 120
Router(config-if)# do show port-security interface fastethernet 5/12 | include Time
Aging Time                 : 120 mins
```

Displaying Port Security Settings

To display port security settings, enter this command:

Command	Purpose
Router# **show port-security** [**interface** {{**vlan** *vlan_ID*} | {*type*[1] *slot/port*}}] [**address**]	Displays port security settings for the switch or for the specified interface.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When displaying port security settings, note the following information:

- Port security supports the **vlan** keyword only on trunks.
- Enter the **address** keyword to display secure MAC addresses, with aging information for each address, globally for the switch or per interface.
- The display includes these values:
  - The maximum allowed number of secure MAC addresses for each interface
  - The number of secure MAC addresses on the interface
  - The number of security violations that have occurred
  - The violation mode.

This example displays output from the **show port-security** command when you do not enter an interface:

```
Router# show port-security
Secure Port      MaxSecureAddr  CurrentAddr  SecurityViolation  Security
Action
                    (Count)        (Count)      (Count)
```

```
--------------------------------------------------------------------------

     Fa5/1            11            11             0            Shutdown
     Fa5/5            15            5              0            Restrict
     Fa5/11           5             4              0            Protect
--------------------------------------------------------------------------

Total Addresses in System: 21
Max Addresses limit in System: 128
```

This example displays output from the **show port-security** command for a specified interface:

```
Router# show port-security interface fastethernet 5/1
Port Security: Enabled
Port status: SecureUp
Violation mode: Shutdown
Maximum MAC Addresses: 11
Total MAC Addresses: 11
Configured MAC Addresses: 3
Aging time: 20 mins
Aging type: Inactivity
SecureStatic address aging: Enabled
Security Violation count: 0
```

This example displays the output from the **show port-security address** privileged EXEC command:

```
Router# show port-security address
          Secure Mac Address Table
-------------------------------------------------------------------
Vlan    Mac Address       Type                Ports   Remaining Age
                                                         (mins)
----    -----------       ----                -----   -------------
   1    0001.0001.0001    SecureDynamic       Fa5/1      15 (I)
   1    0001.0001.0002    SecureDynamic       Fa5/1      15 (I)
   1    0001.0001.1111    SecureConfigured    Fa5/1      16 (I)
   1    0001.0001.1112    SecureConfigured    Fa5/1      -
   1    0001.0001.1113    SecureConfigured    Fa5/1      -
   1    0005.0005.0001    SecureConfigured    Fa5/5      23
   1    0005.0005.0002    SecureConfigured    Fa5/5      23
   1    0005.0005.0003    SecureConfigured    Fa5/5      23
   1    0011.0011.0001    SecureConfigured    Fa5/11     25 (I)
   1    0011.0011.0002    SecureConfigured    Fa5/11     25 (I)
-------------------------------------------------------------------
Total Addresses in System: 10
Max Addresses limit in System: 128
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 10

NetFlow

Configuring NetFlow

This chapter describes how to configure NetFlow statistics collection in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

http://www.cisco.com/en/US/docs/ios/netflow/command/reference/nf_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains the following sections:

Understanding NetFlow

NetFlow Overview

The NetFlow feature collects traffic statistics about the packets that flow through the switch and stores the statistics in the NetFlow table. The NetFlow table on the route processor (RP) captures statistics for flows routed in software and the NetFlow table on the PFC (and on each DFC) captures statistics for flows routed in hardware.

Several features use the NetFlow table. Features such as network address translation (NAT) use NetFlow to modify the forwarding result; other features (such as QoS microflow policing) use the statistics from the NetFlow table to apply QoS policies. The NetFlow Data Export (NDE) feature provides the ability to export the statistics to an external device (called a NetFlow collector).

In PFC3A mode, NetFlow collects statistics only for routed traffic. With other PFCs, you can configure NetFlow to collect statistics for both routed and bridged traffic.

Collecting and exporting a large volume of statistics can significantly impact switch processor (SP) and route processor (RP) CPU usage, so NetFlow provides configuration options to control the volume of statistics. These options include the following:

- NetFlow flow masks determine the granularity of the flows to be measured. Very specific flow masks generate a large number of NetFlow table entries and a large volume of statistics to export. Less specific flow masks aggregate the traffic statistics into fewer NetFlow table entries and generate a lower volume of statistics.
- Per-interface NetFlow allows you to enable or disable NetFlow data collection on Layer 3 interfaces.
- NetFlow Flow Sampling exports data for a subset of traffic in a flow, which can greatly reduce the volume of statistics exported. NetFlow Flow Sampling does not reduce the volume of statistics collected.
- NetFlow aggregation merges the collected statistics prior to export. Aggregation reduces the volume of records exported, but does not reduce the volume of statistics collected. NetFlow aggregation increases SP CPU utilization and reduces the data available at the collector. NetFlow aggregation uses NetFlow version 8.

NetFlow defines three configurable timers to identify stale flows that can be deleted from the table. NetFlow deletes the stale entries to clear table space for new entries.

NetFlow on the PFC

The NetFlow table on the PFC captures statistics for flows routed in hardware. A flow is a unidirectional stream of packets between a source and a destination. The flow mask specifies the fields in the incoming packet that NetFlow uses to match (or create) a NetFlow table entry.

All flow masks include the ingress interface in their definition. Therefore, NetFlow always collects statistics on a per-interface basis. You can also enable or disable NetFlow per-interface.

The PFC supports the following flow masks:

- interface-source—A less-specific flow mask. Statistics for all ingress flows on an interface from each source IP address aggregate into one entry.
- interface-destination—A less-specific flow mask. Statistics for all ingress flows on an interface to each destination IP address aggregate into one entry.

- interface-destination-source—A more-specific flow mask. Statistics for all ingress flows on an interface between the same source IP address and destination IP address aggregate into one entry.
- interface-full—The most-specific flow mask. The PFC creates and maintains a separate table entry for each IP flow on an interface. An interface-full entry includes the source IP address, destination IP address, protocol, and protocol ports.

The flow mask determines the granularity of the statistics gathered, which controls the size of the NetFlow table. The less-specific flow masks result in fewer entries in the NetFlow table and the most-specific flow masks result in the most NetFlow entries.

For example, if the flow mask is set to interface-source, the NetFlow table contains one entry per source IP address. (Assume that NetFlow is enabled on only one interface). The statistics for all flows from each source are accumulated in the one entry. However, if the flow mask is configured as interface-full, the NetFlow table contains one entry per full flow. Many entries may exist per source IP address, so the NetFlow table can become very large. See the "NetFlow Restrictions" section on page 63-4 for information about NetFlow table capacity.

NetFlow on the RP

The NetFlow feature on the RP captures statistics for flows routed in software.

For additional information about configuring NetFlow on the RP, see the *Cisco IOS NetFlow Configuration Guide*, Release 12.2SX.

NetFlow Features

NetFlow supports the following features:

- Per Interface NetFlow, page 63-3
- NetFlow Aggregation, page 63-4
- NetFlow for Multicast IP, page 63-4

Per Interface NetFlow

Cisco IOS Release 12.2(33)SXH and later releases support per-interface NetFlow, which enables PFC NetFlow data collection on a per-interface basis. With releases earlier than Release 12.2(33)SXH, NetFlow on the PFC could be only be enabled and disabled globally.

When you upgrade to a software release that supports the per-interface NetFlow feature, the system automatically enables per-interface NetFlow and configures the **ip flow ingress** command on every Layer 3 interface. This one-time action takes place on the first reload after the upgrade and maintains backward compatibility with the global NetFlow enable command. After the reload, you can configure the **no ip flow ingress** command on Layer 3 interfaces to selectively disable PFC and RP NetFlow data collection.

The per-interface NetFlow feature only applies to IPv4 unicast flows on Layer 3 interfaces. Flows for non-IPv4 protocols (such as IPv6 and MPLS) are not controlled by this feature.

NetFlow Aggregation

NetFlow supports aggregation for packets forwarded in hardware (PFC) or software (RP). See the *Cisco IOS NetFlow Configuration Guide*, Release 12.2SX for information about these features:

- NetFlow aggregation schemes
- Configuring NetFlow aggregation
- ToS-based router aggregation, which is supported by NetFlow on the RP

NetFlow for Multicast IP

NetFlow is supported for multicast IP packets forwarded in hardware (PFC) or software (RP).

NetFlow multicast provides ingress accounting and egress accounting. With ingress accounting, NetFlow creates one flow per source and includes information about how many packet replications occur. With egress accounting, NetFlow creates one flow for each outgoing interface.

Optionally, NetFlow multicast keeps statistics for multicast packets that fail the reverse path fail (RPF) check.

Note Disabling the mls netflow command globally will cause non-RPF multicast traffic to be dropped in software, as new non-RPF Netflow entries will not be created.

Default NetFlow Configuration

Table 63-1 shows the default NetFlow configuration.

Table 63-1 Default NetFlow Configuration

Feature	Default Value
NetFlow	Enabled
NetFlow of routed IP traffic	Disabled
NetFlow of ingress bridged IP traffic	Disabled
NetFlow Sampling	Disabled
NetFlow Aggregation	Disabled
Per-interface NDE	Enabled
Exclude ACL-denied traffic	Disabled (NetFlow creates entries for ACL-denied traffic)

NetFlow Restrictions

- General NetFlow Restrictions, page 63-5
- Flow Mask Conflicts, page 63-6
- Example Feature Configurations, page 63-7

General NetFlow Restrictions

- The CEF table (rather than the NetFlow table) implements Layer 3 switching in hardware.
- Except in PFC3A mode, NetFlow supports bridged IP traffic. PFC3A mode does not support NetFlow bridged IP traffic.
- NetFlow supports multicast IP traffic.
- In PFC3A mode, NAT requires a non-interface-full flow mask for fragmented packets because the PFC cannot provide hardware acceleration for them. Fragmented NAT traffic is sent to the RP to be processed in software, which requires additional fragment ACE entries in the ACL TCAM.

  Other PFC3 modes support the **mls ip nat netflow-frag-l4-zero** command, which removes the specific flow mask requirement and resolves NetFlow mask conflicts between NDE and NAT features. With the **mls ip nat netflow-frag-l4-zero** command configured, the PFC clears the initial fragmented packet L4 information before it gets NAT is applied.
- By default, NAT overload processes initial fragments in software on the RP because NAT for subsequent fragments depends on the Layer 4 information in the first fragment. To ensure that initial fragments do not get switched in hardware, two ACL entries that require a flowmask different from the one for NAT NetFlow are added to the NAT inside interface. Initial fragments hit one of the fragment ACL entries on the NAT inside interface and because it uses a different flowmask, they do not hit the NetFlow shortcut and so are not hardware switched. The two additional ACL entries added to the NAT inside interface could lead to a merge blowup if a big ACL is configured on the NAT inside interface.

  Except in PFC3A mode, you can configure the **mls ip nat netflow-frag-l4-zero** command to zero out the Layer 4 port information from the NetFlow lookup key for fragmented packets, which are then correctly sent to the RP for processing. In Layer 4 zero mode, fragmented packets (including initial fragments) do not match the NetFlow entries that have non-zero Layer 4 port information. In this mode, the 2 additional fragment entries for NAT are not required.

  This can alleviate possible merge failures if a big ACL is configured on the NAT inside interface, and avoids flowmask conflicts between NAT and other features like NDE that arise due to the NAT requirement for a non-interface-full flowmask for fragment entries.

  Note In this mode, fragmented packets are not counted correctly if NDE uses the full or interface-full flowmask. Similarly, initial fragments are not counted against the correct bucket with microflow policing that uses the full-flow mask.

- No statistics are available for flows that are switched when the NetFlow table is full.
- If the NetFlow table utilization exceeds the recommended utilization levels, there is an increased probability that there will be insufficient room to store statistics. Table 63-2 lists the recommended maximum utilization levels.

Table 63-2 NetFlow Table Utilization

PFC	Recommended NetFlow Table Utilization	Total NetFlow Table Capacity
PFC3CXL	235,520 (230 K) entries	262,144 (256 K) entries
PFC3C	117,760 (115 K) entries	131,072 (128 K) entries
PFC3BXL	235,520 (230 K) entries	262,144 (256 K) entries

Table 63-2 NetFlow Table Utilization

PFC	Recommended NetFlow Table Utilization	Total NetFlow Table Capacity
PFC3B	117,760 (115 K) entries	131,072 (128 K) entries
PFC3A	65,536 (64 K) entries	131,072 (128 K) entries

Flow Mask Conflicts

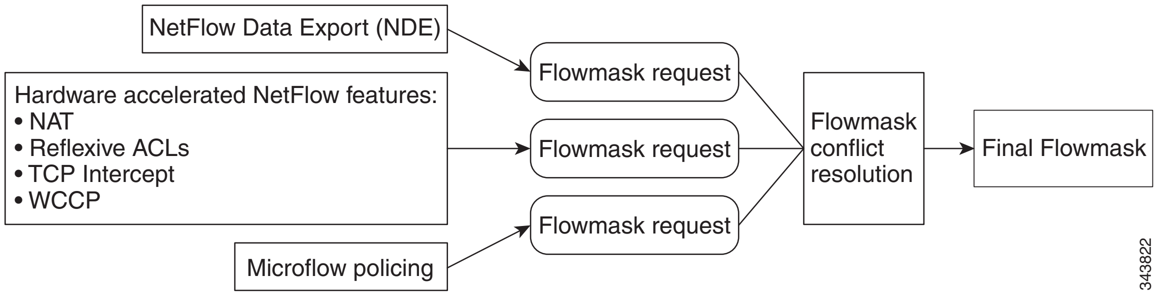
Several features use the NetFlow table. Table 63-3 lists the flow mask requirements for each feature.

Table 63-3 Feature Requirements for Flow Masks

Note "Min" indicates that the flowmask requirement is flexible: a more granular flowmask will also work. For example, "interface-source min" indicates that interface-source-destination can also be used.

"Exact" indicates that the flowmask requirement is not flexible.

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
Reflexive ACL							Exact	
TCP Intercept						Min		
Web Cache Redirect (WCCP)							Exact	
Server Load Balancing (SLB)						Min		
Network Address Translation (NAT)								
• Without **mls ip nat netflow-frag-l4-zero**:							Exact	Exact
• With **mls ip nat netflow-frag-l4-zero**:							Exact	
NetFlow Data Export (NDE)								
• With **mls flow ip interface-source**:		Min						
• With **mls flow ip interface-destination**:				Min				
• With **mls flow ip interface-destination-source**:					Min			
• With **mls flow ip interface-full**:							Min	
NetFlow Sampling							Min	
NetFlow Aggregation				Min				
Microflow Policing								
• With **police flow mask full-flow**:							Exact	
• With **police flow mask src-only**:	Exact							
• With **police flow mask dest-only**:			Exact					



NetFlow data export, hardware accelerated NetFlow features, and microflow policers are the three categories of features that request flowmasks. All of them can be configured at the same time, but depending on the flowmask that each requests, the configuration might be allowed or not. Typically, only one of the hardware accelerated NetFlow features is configured on an interface. If multiple hardware accelerated NetFlow features are configured, some of them might not be hardware accelerated if the flow is subject to more than one of these features.

Each of these features request a flowmask. The final flowmask depends on the other features.

Example Feature Configurations

Note

- "Min" indicates that the flowmask requirement is flexible: a more granular flowmask will also work. For example, "interface-source min" indicates that interface-source-destination can also be used.
- "Exact" indicates that the flowmask requirement is not flexible.
- You can use the information in Table 63-3 on page 63-6 to check for conflicts between features.

NAT "Non-interface Full" would conflict with NDE "Interface Full", but NAT "Interface Full" would not conflict with NDE "Interface Full":

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
Network Address Translation (NAT) without **mls ip nat netflow-frag-l4-zero**:							Exact	Exact
NetFlow Data Export (NDE) with **mls flow ip interface-full**:							Min	

NAT “Interface Full” would not conflict with NDE “Interface Full”:

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
Network Address Translation (NAT) with **mls ip nat netflow-frag-l4-zero**:							Exact	
NetFlow Data Export (NDE) with **mls flow ip interface-full**:							Min	

NAT “Interface Full” would not conflict with NDE “Interface Source”:

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
Network Address Translation (NAT) with **mls ip nat netflow-frag-l4-zero**:							Exact	
NetFlow Data Export (NDE) with **mls flow ip interface-source**:		Min						

WCCP would not conflict with NDE “Interface Destination Source”:

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
Web Cache Redirect (WCCP)							Exact	
NetFlow Data Export (NDE) With **mls flow ip interface-destination-source**:					Min			

WCCP would not conflict with NDE “Interface Full”:

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
Web Cache Redirect (WCCP)							Exact	
NetFlow Data Export (NDE) With **mls flow ip interface-full**:							Min	

NDE "Interface Full" would conflict with microflow policing "Destination":

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
NetFlow Data Export (NDE) with **mls flow ip interface-full**:							Min	
Microflow Policing with **police flow mask dest-only**:			Exact					

NDE "Interface Full" would not conflict with microflow policing "Interface Full":

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
NetFlow Data Export (NDE) with **mls flow ip interface-full**:							Min	
Microflow Policing with **police flow mask full-flow**:							Exact	

WCCP, NAT "Interface Full", and NDE "Interface Full" would not conflict:

Feature	Source	Interface Source	Destination	Interface Destination	Interface Destination Source	Full	Interface Full	Non-interface Full
Web Cache Redirect (WCCP)							Exact	
Network Address Translation (NAT) with **mls ip nat netflow-frag-l4-zero**:							Exact	
NetFlow Data Export (NDE) with **mls flow ip interface-full**:							Min	

Configuring NetFlow

These sections describe how to configure NetFlow:

- Configuring NetFlow on the PFC, page 63-10
- Configuring NetFlow Features, page 63-13

Configuring NetFlow on the PFC

These sections describe how to configure NetFlow statistics collection on the PFC:

NetFlow PFC Commands Summary

Table 63-4 shows a summary of the NetFlow commands available on the PFC.

Table 63-4 Summary of PFC NetFlow Commands

Command	Purpose
mls netflow	Enables NetFlow on the PFC.
mls flow ip	Sets the minimum flow mask.
mls aging	Sets the configurable aging parameters.
mls exclude acl-deny	Disables the creation of flows for ACL-denied traffic.
show mls netflow {...}	Displays NetFlow PFC information for unicast and multicast traffic.
show mls netflow aggregation flowmask	Displays the NetFlow aggregation flow mask.

Enabling NetFlow on the PFC

To enable NetFlow statistics collection globally on the PFC, perform this task:

Command	Purpose
`Router(config)# mls netflow`	Enables NetFlow on the PFC.

This example shows how to disable NetFlow statistics collection on the PFC (the default setting is enabled):

```
Router(config)# no mls netflow
```

Setting the Minimum IP MLS Flow Mask

You can set the minimum specificity of the flow mask for the NetFlow table on the PFC. The actual flow mask may be more specific than the level configured in the **mls flow** command, if other configured features need a more specific flow mask (see the "Flow Mask Conflicts" section on page 63-6).

To set the minimum IPv4 flow mask, perform this task:

Command	Purpose
Router(config)# **mls flow ip** {**interface-source** | **interface-destination** | **interface-destination-source** | **interface-full**}	Sets the minimum flow mask for IPv4 packets.

This example shows how to set the minimum flow mask:

```
Router(config)# mls flow ip interface-destination
```

To display the IP MLS flow mask configuration, perform this task:

Command	Purpose
Router# **show mls netflow flowmask**	Displays the flow mask configuration.

This example shows how to display the MLS flow mask configuration:

```
Router# show mls netflow flowmask
current ip flowmask for unicast: if-dst
Router#
```

Configuring the MLS Aging Time

The MLS aging time (default 300 seconds) applies to all NetFlow table entries. You can configure the normal aging time in the range of 32 to 4092 seconds. Flows can age as much as 4 seconds sooner or later than the configured interval. On average, flows age within 2 seconds of the configured value.

Other events might cause MLS entries to be purged, such as routing changes or a change in link state.

Note If the number of MLS entries exceeds the recommended utilization (see the "NetFlow Restrictions" section on page 63-4), only adjacency statistics might be available for some flows.

To keep the NetFlow table size below the recommended utilization, enable the following parameters when using the **mls aging** command:

- **normal**—Configures an inactivity timer. If no packets are received on a flow within the duration of the timer, the flow entry is deleted from the table.
- **fast aging**—Configures an efficient process to age out entries created for flows that only switch a few packets, and then are never used again. The **fast aging** parameter uses the **time** keyword value to check if at least the **threshold** keyword value of packets have been switched for each flow. If a flow has not switched the threshold number of packets during the time interval, then the entry is aged out.
- **long**—Configures entries for deletion that have been active for the specified value even if the entry is still in use. Long aging is used to prevent counter wraparound, which can cause inaccurate statistics.

A typical table entry that is removed by fast aging is the entry for flows to and from a Domain Name Server (DNS) or TFTP server.

If you need to enable MLS fast aging time, initially set the value to 128 seconds. If the size of the NetFlow table continues to grow over the recommended utilization, decrease the setting until the table size stays below the recommended utilization. If the table continues to grow over the recommended utilization, decrease the normal MLS aging time.

To configure the MLS aging time, perform this task:

Command	Purpose
Router(config)# **mls aging** {**fast** [**threshold** {*1-128*} | **time** {*1-128*}] | **long** *64-1920* | **normal** *32-4092*}	Configures the MLS aging time for a NetFlow table entry.

This example displays how to configure the MLS aging time:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls aging fast threshold 64 time 30
```

To display the MLS aging-time configuration, perform this task:

Command	Purpose
Router# **show mls netflow aging**	Displays the MLS aging-time configuration.

This example shows how to display the MLS aging-time configuration:

```
Router# show mls netflow aging
enable timeout packet threshold
------ ------- ----------------
normal aging true 300 N/A
fast aging true 32 100
long aging true 900 N/A
```

Configuring Exclude ACL-deny

By default, NetFlow table entries are created for ACL-denied flows. These flows can cause the NetFlow table to overflow. With Release 12.2(33)SXH and later releases, to exclude ACL-denied flows from the NetFlow table, perform this task:

Command	Purpose
Router# **mls exclude acl-deny**	Excludes ACL-denied flows from the NetFlow table.

This example shows how to exclude ACL-denied flows from the NetFlow table:

```
Router(config)# mls exclude acl-deny
```

Displaying PFC NetFlow Information

To display information about NetFlow on the PFC, perform this task:

Command	Purpose
Router(config)# **show mls netflow** {**aggregation** | **aging** | **creation** | **flowmask** | **ip** | **ipv6** | **mpls** | **table-contention** | **usage**}	Displays information about NetFlow on the PFC.

Configuring NetFlow Features

NetFlow features generally apply to packets forwarded in hardware (PFC) and software (RP). For the features to apply to PFC, you need to enable NetFlow on the PFC.

These sections describe how to configure NetFlow features:

- Configuring NetFlow on Layer 3 Interfaces, page 63-13
- Enabling NetFlow for Ingress-Bridged IP Traffic, page 63-14
- Configuring NetFlow Aggregation, page 63-14
- Configuring NetFlow for Multicast IP Traffic, page 63-15

Configuring NetFlow on Layer 3 Interfaces

The per-interface NDE feature allows you to enable or disable NetFlow collection on a per-interface basis for packets forwarded in hardware (PFC) or software (RP). This feature is automatically enabled in Release 12.2(33)SXH and later releases.

To enable or disable NetFlow for a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# **interface** {**vlan** *vlan_ID*} | {*type slot/port*} | {**port-channel** *port_channel_number*}	Selects a Layer 3 interface to configure.
Step 2	Router(config-if)# **ip flow ingress**	Enables NetFlow for the specified interface. NetFlow will collect statistics for packets forwarded in hardware (PFC) or software (RP).
Step 3	Router(config-if)# **no ip flow ingress**	Disables NetFlow for the specified interface. NetFlow will stop collecting statistics for packets forwarded in hardware (PFC) or software (RP).

When you upgrade for the first time to a software image that supports per-interface NetFlow on the PFC, the system automatically configures each Layer 3 interface to enable NetFlow (this ensures backward compatibility with the global **mls netflow** command). This one-time action occurs during the first system restart after the upgrade. After this action, you can configure Layer 3 interfaces to disable or enable NetFlow data collection.

Enabling NetFlow for Ingress-Bridged IP Traffic

Except in PFC3A mode, NetFlow supports ingress-bridged IP traffic. PFC3A mode does not support NetFlow for bridged IP traffic.

Note

- When you enable NetFlow for ingress-bridged IP traffic, the statistics are available to the NetFlow Flow Sampling feature (see the "NetFlow Sampling" section on page 64-7).
- To enable NetFlow for bridged IP traffic on a VLAN, you must create a corresponding VLAN interface and enter the **no shutdown** command. The **no shutdown** command can be followed, if necessary, by the **shutdown** command.
- For Layer 3 VLANs, enabling NetFlow for ingress-bridged IP traffic also enables NetFlow for Layer 3 flows on the specified VLANs.
- The exported bridged flows will have ingress and egress VLAN information and not the physical port information.

To enable NetFlow for ingress-bridged IP traffic in VLANs, perform this task:

Command	Purpose
Router(config)# **ip flow ingress layer2-switched vlan** *vlan_ID*[-*vlan_ID*] [**,** *vlan_ID*[-*vlan_ID*]]	Enables NetFlow for ingress-bridged IP traffic in the specified VLANs. **Note** NetFlow for ingress-bridged IP traffic in a VLAN requires that NetFlow on the PFC be enabled with the **mls netflow** command.

This example shows how to enable NetFlow for ingress-bridged IP traffic in VLAN 200:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip flow ingress layer2-switched vlan 200
```

Configuring NetFlow Aggregation

To configure NetFlow aggregation, use the procedures in the *Cisco IOS NetFlow Configuration Guide*, Release 12.2SX.

Note

- When you configure NetFlow aggregation, it is configured automatically for packets forwarded in hardware (PFC) or software (RP).
- The PFC and DFCs do not support NetFlow ToS-based router aggregation.

To display NetFlow Aggregation information for the PFC or DFCs, perform this task:

Command	Purpose
Router # **show ip cache flow aggregation** {**as** | **destination-prefix** | **prefix** | **protocol-port** | **source-prefix**) **module** *slot_num*	Displays the NetFlow Aggregation cache information.
Router # **show mls netflow aggregation flowmask**	Displays the NetFlow Aggregation flow mask information.

Note The PFC and DFCs do not support NetFlow ToS-based router Aggregation.

This example shows how to display the NetFlow Aggregation cache information:

```
Router# show ip cache flow aggregation destination-prefix module 1
IPFLOW_DST_PREFIX_AGGREGATION records and statistics for module :1
IP Flow Switching Cache, 278544 bytes
2 active, 4094 inactive, 6 added
236 ager polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
Dst If Dst Prefix Msk AS Flows Pkts B/Pk Active
Gi7/9 9.1.0.0 /16 0 3003 12M 64 1699.8
Gi7/10 11.1.0.0 /16 0 3000 9873K 64 1699.8
Router#
```

This example shows how to display the NetFlow Aggregation flow mask information:

```
Router# show mls netflow aggregation flowmask
 Current flowmask set for netflow aggregation : Vlan Full Flow
 Netflow aggregations configured/enabled :
        AS Aggregation
        PROTOCOL-PORT Aggregation
        SOURCE-PREFIX Aggregation
        DESTINATION-PREFIX Aggregation
Router
```

Configuring NetFlow for Multicast IP Traffic

To configure NetFlow for multicast IP traffic, perform this task:

	Command	Purpose
Step 1	Router(config)# **ip multicast netflow output-counters**	(Optional) Enables the calculation of output bytes/packets for an ingress flow.
Step 2	Router(config)# **ip multicast netflow rpf-failure**	(Optional) Enables NetFlow for multicast data that fails the RPF check.

	Command	Purpose
Step 3	`Router(config)#` **`interface`** `{`**`vlan`** *`vlan_ID`*`} | {`*`type slot/port`*`} | {`**`port-channel`** *`port_channel_number`*`}`	Selects a Layer 3 interface to configure.
Step 4	`Router(config-if)#` **`ip flow`** `{`**`ingress`** `|` **`egress`**`}`	Enables NetFlow multicast traffic on the specified interface (for RP and PFC). • Specify **ingress** to enable NetFlow multicast ingress accounting. • Specify **egress** to enable NetFlow multicast egress accounting.

For additional information about configuring NetFlow for multicast traffic, see the "Configuring NetFlow Multicast Accounting" section of the *Cisco IOS NetFlow Configuration Guide*, Release 12.2SX.

The "Configuring NetFlow Multicast Accounting" section specifies as a prerequisite that you need to configure multicast fast switching or multicast distributed fast switching (MDFS), but this prerequisite does not apply when configuring NetFlow multicast support with 12.2SX releases.

Note The Configuring NetFlow Multicast Accounting document describes new configuration commands for Cisco IOS Release 12.2(4) and newer releases. The 12.2SX releases support the new commands.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring NDE

This chapter describes how to configure NetFlow Data Export (NDE).

Note For complete syntax and usage information for the commands used in this chapter, see these publications:

- The Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/netflow/command/reference/nf_book.html

- The *Cisco IOS NetFlow Configuration Guide*, Release 12.2SX , which provides information about NetFlow version 9.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains the following sections:

Understanding NDE

These sections describe how NetFlow Data Export (NDE) works:

- NDE Overview, page 64-2
- NDE on the RP, page 64-2

NDE Overview

NetFlow collects traffic statistics by monitoring the packets that flow through the switch and storing the statistics in the NetFlow table. For more information about NetFlow, see Chapter 63, "Configuring NetFlow."

NetFlow Data Export (NDE) converts the NetFlow table statistics into records and exports the records to an external device, which is called a NetFlow collector.

In PFC3A mode, NDE exports statistics only for routed traffic. With modes other than PFC3A, you can configure NDE to export statistics for both routed and bridged traffic.

You can export IP unicast statistics using NDE record format versions 5, 7 or 9. Use NDE version 8 record format for NetFlow aggregation, and version 9 record format for IP multicast.

Exporting a large volume of statistics can significantly impact SP and RP CPU utilization. You can control the volume of records exported by configuring NDE flow filters to include or exclude flows from the NDE export. When you configure a filter, NDE exports only the flows that match the filter criteria.

You can configure up to two external data collector addresses. A second data collector improves the probability of receiving complete NetFlow data by providing redundant data streams.

NDE on the RP

The RP supports these features, which are documented in the *Cisco IOS NetFlow Configuration Guide*, Release 12.2SX:

- NDE for flows routed in software
- NetFlow aggregation
- NetFlow ToS-based router aggregation
- NetFlow flow sampling
- NetFlow version 9 export

NDE on the PFC

NDE on the PFC exports statistics for flows routed or bridged in hardware. These sections describe NDE on the PFC in more detail:

- NDE Flow Mask, page 64-3
- NDE Versions, page 64-3
- Exporting NetFlow Data, page 64-7
- NetFlow Sampling, page 64-7

NDE Flow Mask

You can configure the minimum NetFlow flow mask for NDE. The NetFlow flow mask determines the granularity of the statistics gathered, which controls the volume of statistics for NDE to export.

For more details about flow masks, see Chapter 63, "Configuring NetFlow."

Additional NDE Fields

You can configure NDE to populate the following additional fields in the NDE packets:

- IP address of the next hop router
- Egress interface SNMP ifIndex
- BGP AS

These fields are populated by the software looking up the FIB table entry before sending out the NDE record to the collector. These fields are empty when you use the **show** command to display the hardware NetFlow table.

NDE Versions

- NetFlow version 9 is described in this publication:

  http://www.cisco.com/en/US/docs/ios-xml/ios/netflow/configuration/12-2sx/cfg-nflow-data-expt.html
- NDE exports statistics for NetFlow aggregation flows using NDE version 8. The following document describes the version 8 header format:

  http://www.cisco.com/en/US/docs/ios-xml/ios/netflow/configuration/12-2sx/ios-netflow-ov.html
- With 12.2SX releases, NDE exports IP unicast traffic using NDE versions 5, 7 and 9.

  Some fields in the flow records might not have values, depending on the current flow mask. Unsupported fields contain a zero (0).

Note With the WCCP Layer 2 redirect, the nexthop field and the output field might not contain accurate information for all NetFlows. Therefore, the destination interface for traffic returned from the web server has a client interface instead of the cache interface or the ANCS interface.

  The following tables describe the supported fields for NDE versions 5 and 7:
  - Table 64-1—Version 5 header format
  - Table 64-2—Version 7 header format
  - Table 64-3—Version 5 flow record format
  - Table 64-4—Version 7 flow record format

Table 64-1 NDE Version 5 Header Format

Bytes	Content	Description
0–1	version	NetFlow export format version number
2–3	count	Number of flows exported in this packet (1–30)
4–7	SysUptime	Current time in milliseconds since router booted
8–11	unix_secs	Current seconds since 0000 UTC 1970
12–15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970
16–19	flow_sequence	Sequence counter of total flows seen
20–21	engine_type	Type of flow switching engine
21–23	engine_id	Slot number of the flow switching engine

Table 64-2 NDE Version 7 Header Format

Bytes	Content	Description
0–1	version	NetFlow export format version number
2–3	count	Number of flows exported in this packet (1–30)
4–7	SysUptime	Current time in milliseconds since router booted
8–11	unix_secs	Current seconds since 0000 UTC 1970
12–15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970
16–19	flow_sequence	Sequence counter of total flows seen
20–23	reserved	Unused (zero) bytes

Table 64-3 NDE Version 5 Flow Record Format

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field (see the "Populating Additional NDE Fields" section on page 64-12)					
			Source	Destination	Destination Source	Destination Source Interface	Full	Full Interface
0–3	srcaddr	Source IP address	X	0	X	X	X	X
4–7	dstaddr	Destination IP address	0	X	X	X	X	X
8–11	nexthop	Next hop router's IP address[1]	0	A[2]	A	A	A	A
12–13	input	Ingress interface SNMP ifIndex	0	0	0	X	0	X
14–15	output	Egress interface SNMP ifIndex[3]	0	A[2]	A	A	A	A
16–19	dPkts	Packets in the flow	X	X	X	X	X	X
20–23	dOctets	Octets (bytes) in the flow	X	X	X	X	X	X
24–27	first	SysUptime at start of the flow (milliseconds)	X	X	X	X	X	X
28–31	last	SysUptime at the time the last packet of the flow was received (milliseconds)	X	X	X	X	X	X
32–33	srcport	Layer 4 source port number or equivalent	0	0	0	0	X[4]	X[4]
34–35	dstport	Layer 4 destination port number or equivalent	0	0	0	0	X	X
36	pad1	Unused (zero) byte	0	0	0	0	0	0
37	tcp_flags	Cumulative OR of TCP flags[5]	0	0	0	0	0	0
38	prot	Layer 4 protocol (for example, 6=TCP, 17=UDP)	0	0	0	0	X	X
39	tos	IP type-of-service byte	X[6]	X[6]	X[6]	X[6]	X[6]	X[6]
40–41	src_as	Autonomous system number of the source, either origin or peer	X	0	X	X	X	X
42–43	dst_as	Autonomous system number of the destination, either origin or peer	0	X	X	X	X	X
44–45	src_mask	Source address prefix mask bits	X	0	X	X	X	X
46–47	dst_mask	Destination address prefix mask bits	0	X	X	X	X	X
48	pad2	Pad 2	0	0	0	0	0	0

1. Always zero when PBR, WCCP, or SLB is configured.
2. With the destination flow mask, the "Next hop router's IP address" field and the "Output interface's SNMP ifIndex" field might not contain information that is accurate for all flows.
3. Always zero when policy-based routing is configured.
4. Except in PFC3A mode, for ICMP traffic, contains the ICMP code and type values.
5. Always zero for hardware-switched flows.
6. Not populated in PFC3A mode.

Table 64-4 NDE Version 7 Flow Record Format

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field (see the "Populating Additional NDE Fields" section on page 64-12)					
			Source	Destination	Destination Source	Destination Source Interface	Full	Full Interface
0–3	srcaddr	Source IP address	X	0	X	X	X	X
4–7	dstaddr	Destination IP address	0	X	X	X	X	X
8–11	nexthop	Next hop router's IP address[1]	0	A[2]	A	A	A	A
12–13	input	Ingress interface SNMP ifIndex	0	0	0	X	0	X
14–15	output	Egress interface SNMP ifIndex[3]	0	A[2]	A	A	A	A
16–19	dPkts	Packets in the flow	X	X	X	X	X	X
20–23	dOctets	Octets (bytes) in the flow	X	X	X	X	X	X
24–27	First	SysUptime at start of the flow (milliseconds)	X	X	X	X	X	X
28–31	Last	SysUptime at the time the last packet of the flow was received (milliseconds)	X	X	X	X	X	X
32–33	srcport	Layer 4 source port number or equivalent	0	0	0	0	X[4]	X[4]
34–35	dstport	Layer 4 destination port number or equivalent	0	0	0	0	X	X
36	flags	Flow mask in use	X	X	X	X	X	X
37	tcp_flags	Cumulative OR of TCP flags[5]	0	0	0	0	0	0
38	prot	Layer 4 protocol (for example, 6=TCP, 17=UDP)	0	0	0	0	X	X
39	tos	IP type-of-service byte	X[6]	X[6]	X[6]	X[6]	X[6]	X[6]
40–41	src_as	Autonomous system number of the source, either origin or peer	X	0	X	X	X	X
42–43	dst_as	Autonomous system number of the destination, either origin or peer	0	X	X	X	X	X
44	src_mask	Source address prefix mask bits	X	0	X	X	X	X
45	dst_mask	Destination address prefix mask bits	0	X	X	X	X	X
46–47	pad2	Pad 2	0	0	0	0	0	0
48–51	MLS RP	IP address of MLS router	0	X	X	X	X	X

1. Always zero when PBR, WCCP, or SLB is configured.
2. With the destination flow mask, the "Next hop router's IP address" field and the "Output interface's SNMP ifIndex" field might not contain information that is accurate for all flows.
3. Always zero when policy-based routing is configured.
4. Except in PFC3A mode, for ICMP traffic, contains the ICMP code and type values.
5. Always zero for hardware-switched flows.
6. Not populated in PFC3A mode.

Exporting NetFlow Data

NetFlow maintains traffic statistics for each active flow in the NetFlow table and increments the statistics when packets within each flow are switched.

Periodically, NDE exports summarized traffic statistics for all expired flows, which the external data collector receives and processes.

Exported NetFlow data contains statistics for the flow entries in the NetFlow table that have expired since the last export. Flow entries in the NetFlow table expire and are flushed from the NetFlow table when one of the following conditions occurs:

- The entry ages out.
- The entry is cleared by the user.
- An interface goes down.
- Route flaps occur.

To ensure periodic reporting of continuously active flows, entries for continuously active flows expire at the end of the interval configured with the **mls aging long** command (default 32 minutes).

NDE packets go to the external data collector either when the number of recently expired flows reaches a predetermined maximum or after:

- 30 seconds for version 5 export.
- 10 seconds for version 9 export.

By default, all expired flows are exported unless they are filtered. If you configure a filter, NDE only exports expired and purged flows that match the filter criteria. NDE flow filters are stored in NVRAM and are not cleared when NDE is disabled. See the "Configuring NDE Flow Filters" section on page 64-16 for NDE filter configuration procedures.

NetFlow Sampling

NetFlow sampling is used when you want to report statistics for a subset of the traffic flowing through your network. The Netflow statistics can be exported to an external collector for further analysis.

There are two types of NetFlow sampling: NetFlow traffic sampling and NetFlow flow sampling. The configuration steps for configuring MSFC-based NetFlow traffic sampling for traffic switched in the software path and PFC/DFC-based NetFlow flow sampling for traffic switched in the hardware path on a Cisco 6500 series switch use different commands because they are mutually independent features.

The following sections provide additional information on the two types of NetFlow sampling supported by Cisco 6500 series switches:

- NetFlow Traffic Sampling, page 64-7
- NetFlow Flow Sampling, page 64-8

NetFlow Traffic Sampling

NetFlow traffic sampling provides NetFlow data for a subset of traffic forwarded by a Cisco router or switch by analyzing only one randomly selected packet out of n sequential packets (n is a user-configurable parameter) from the traffic that is processed by the router or switch. NetFlow traffic sampling is used on platforms that perform software-based NetFlow accounting, such as Cisco 7200 series routers and Cisco 6500 series MSFCs, to reduce the CPU overhead of running NetFlow by reducing the number of packets that are analyzed (sampled) by NetFlow. The reduction in the number of packets sampled by NetFlow on platforms that perform software based NetFlow accounting also reduces the number of packets that need to be exported to an external collector. Reducing the number of packets that need to be exported to an external collector by reducing the number of packets that are analyzed is useful when the volume of exported traffic created by analyzing every packet will overwhelm the collector, or result in an over-subscription of an outbound interface.

NetFlow traffic sampling and export for software-based NetFlow accounting behaves in the following manner:

- The flows are populated with statistics from a subset of the traffic that is seen by the router.
- The flows are expired.
- The statistics are exported.

On Cisco 6500 series switches, NetFlow traffic sampling is supported only on the MSFC for software switched packets. For more information on configuring NetFlow traffic sampling, see the *Cisco IOS NetFlow Configuration Guide*.

NetFlow Flow Sampling

NetFlow flow sampling does not limit the number of packets that are analyzed by NetFlow. NetFlow flow sampling is used to select a subset of the flows processed by the router for export. NetFlow flow sampling is not a solution to reduce oversubscribed CPUs or oversubscribed hardware NetFlow table usage. NetFlow flow sampling can help reduce CPU usage by reducing the amount of data that is exported. Using NetFlow flow sampling to reduce the number of packets that need to be exported to an external collector by reporting statistics on only a subset of the flows is useful when the volume of exported traffic created by reporting statistics for all of the flows will overwhelm the collector, or result in an over-subscription of an outbound interface.

NetFlow flow sampling is available on Cisco Catalyst 6500 series switches for hardware-based NetFlow accounting on the PFCs and DFCs installed in the router.

NetFlow flow sampling and export for hardware-based NetFlow accounting behaves in the following manner:

- Packets arrive at the switch and flows are created/updated to reflect the traffic seen.
- The flows are expired.
- The flows are sampled to select a subset of flows for exporting.
- The statistics for the subset of flows that have been selected by the NetFlow flow sampler are exported.

Note When NetFlow flow sampling is enabled, aging schemes such as fast, normal, long aging are disabled.

You can configure NetFlow flow sampling to use time-based sampling or packet-based sampling. With either the full-interface or destination-source-interface flow masks, you can enable or disable NetFlow Flow Sampling on each Layer 3 interface.

Packet-based NetFlow Flow Sampling

Packet-based NetFlow flow sampling uses a sampling-rate in packets and an interval in milliseconds to select a subset (sample) of flows from the total number of flows processed by the router. The values for the sampling-rate are: 64, 128, 256, 512, 1024, 2048, 4096, 8192. The interval is a user-configurable value in the range 8000-16000 milliseconds. The default for the interval is 16000 milliseconds. The interval value replaces the aging schemes such as fast, normal, long aging for expiring flows from the cache. The command syntax for configuring packet-based NetFlow flow sampling is:
mls sampling packet-based *rate* [*interval*].

Packet-based NetFlow flow sampling uses one of these two methods to select flows for sampling and export:

- **The number of packets in the expired flow exceeds the sampling rate**: If in a interval of X - where X is a value in the range of 8000-16000 (inclusive), a flow has a greater number of packets than the value configured for the sampling-rate, the flow is sampled (selected) and then exported.
- **The number of packets in the expired flow is less than the sampling rate**: If in a interval of X - where X is a value in the range of 8000-16000 (inclusive), a flow has a smaller number of packets than the value configured for the sampling-rate, the packet count for the flow is added to one of eight buckets based on the number of packets in the flow. The eight bucket sizes are $1/8^{th}$ increments of the sampling rate. The packet count for a flow that contains a quantity of packets that is $0–1/8^{th}$ of the sampling rate is assigned to the first bucket. The packet count for a flow that contains a quantity of packets that is $1/8^{th}–2/8^{th}$ of the sampling rate is assigned to the second bucket. And so on. When adding the packet count for a flow to a bucket causes the counter for the bucket to exceed the sampling rate, the last flow for which the counters were added to the bucket is sampled and exported. The bucket counter is changed to 0 and the process of increasing the bucket counter is started over. This method ensures that some flows for which the packet count never exceeds the sampling rate are selected for sampling and export.

Time-based Netflow Flow Sampling

Time-based Netflow flow sampling samples flows created in the first sampling time (in milliseconds) of the export interval time (in milliseconds). Each of the sampling rates that you can configure with the **mls sampling time-based** *rate* command has fixed values for the sampling time and export interval used by time-based NetFlow flow sampling. For example:

- If you configure a sampling rate of 64, NefFlow flow sampling selects flows created within the first 64 milliseconds (sampling time) of every 4096 millisecond export interval.
- If you configure a sampling rate of 2048, NefFlow flow sampling selects flows created within the first 4 milliseconds (sampling time) of every 8192 millisecond export interval.

Table 64-5 lists the sampling rates and export intervals for time-based NetFlow flow sampling.

Table 64-5 Time-Based Sampling Rates, Sampling Times, and Export Intervals

Sampling Rate (Configurable)	Sampling Time in Milliseconds (Not Configurable)	Export Interval Milliseconds (Not Configurable)
1 in 64	64	4096
1 in 128	32	4096
1 in 256	16	4096
1 in 512	8	4096
1 in 1024	4	4096
1 in 2048	4	8192
1 in 4096	4	16384
1 in 8192	4	32768

Default NDE Configuration

Table 64-6 shows the default NDE configuration.

Table 64-6 Default NDE Configuration

Feature	Default Value
NDE	Disabled
NDE of ingress bridged IP traffic	Disabled
NDE source addresses	None
NDE data collector address and UDP port	None
NDE filters	None
Populating additional NDE fields	Enabled

NDE Configuration Guidelines and Restrictions

When configuring NDE, follow these guidelines and restrictions:

- You must enable NetFlow on the PFC to export data for packets forwarded in hardware.
- When you configure NAT and NDE on an interface, the PFC sends all fragmented packets to the RP to be processed in software. (CSCdz51590)
- NDE supports IP multicast traffic only with NetFlow version 9.
- NetFlow aggregation must use NDE version 8 or version 9.
- Except in PFC3A mode, NDE supports bridged IP traffic. PFC3A mode does not support NDE for bridged IP traffic.
- NDE does not support Internetwork Packet Exchange (IPX) traffic or any other non-IP protocol.

The following IPv4 Netflow and NDE options are not available for IPv6 flows:

- Aggregation support (ip flow-aggregation cache command)
- Export of Layer 2 switched IPv6 flows
- Netflow and NDE sampling
- NDE filter support

Configuring NDE

These sections describe how to configure NDE:

- Configuring NDE on the PFC, page 64-11
- Configuring NDE on the RP, page 64-13
- Enabling NDE for Ingress-Bridged IP Traffic, page 64-14
- Displaying the NDE Address and Port Configuration, page 64-15
- Configuring NDE Flow Filters, page 64-16
- Displaying the NDE Configuration, page 64-17

Configuring NDE on the PFC

These sections describe how to configure NDE on the PFC:

- Enabling NDE From the PFC, page 64-11
- Populating Additional NDE Fields, page 64-12
- Configuring NetFlow Flow Sampling, page 64-12

Enabling NDE From the PFC

To enable NDE from the PFC, perform this task:

Command	Purpose
`Router(config)# mls nde sender [version {5 | 7}]`	Enables NDE from the PFC using version 7 records or version 5 records. If you enter the **mls nde sender** command without using the **version** {**5** | **7**} keywords version 7 records are enabled by default. **Note** If you are using NDE for direct export with WS-X6708-10GE, WS-X6716-10GE, or WS-X6716-10T ports, enter the **mls nde sender version 5** command.
`Router(config)# ip flow-export version 9`	(Optional) Enables the use of version 9 records[1]. If you want to enable the use of version 9 records for NDE, you must enter the **mls nde sender** command first. **Note** Enabling the use of version 9 records overrides the use of either version 5 records or version 7 records.

1. The **ip flow-export version 9** command was integrated into 12.2(18)SXF.

Note

- NDE from the PFC uses the source interface configured for the RP (see the "Configuring the RP NDE Source Layer 3 Interface" section on page 64-13).
- NetFlow version 9 is described at this URL:

  http://www.cisco.com/en/US/docs/ios-xml/ios/netflow/configuration/12-2sx/cfg-nflow-data-expt.html

This example shows how to enable NDE from the PFC:

```
Router(config)# mls nde sender
```

This example shows how to enable NDE from the PFC and configure NDE version 5:

```
Router(config)# mls nde sender version 5
```

Populating Additional NDE Fields

You can configure NDE to populate the following additional fields in the NDE packets:

- IP address of the next hop router
- Egress interface SNMP ifIndex
- BGP AS

Not all of the additional fields are populated with all flow masks. See the "NDE Versions" section on page 64-3 for additional information.

To populate the additional fields in NDE packets, perform this task:

Command	Purpose
`Router(config)# mls nde interface`	Populates additional fields in NDE packets.

This example shows how to populate the additional fields in NDE packets:

```
Router(config)# mls nde interface
```

Configuring NetFlow Flow Sampling

These sections describe how to configure NetFlow flow sampling on the PFC:

- Configuring NetFlow Flow Sampling Globally, page 64-12
- Configuring NetFlow Flow Sampling on a Layer 3 Interface, page 64-12

Configuring NetFlow Flow Sampling Globally

To configure NetFlow flow sampling globally, perform this task:

	Command	Purpose
Step 1	Router(config)# **mls sampling** {**time-based** *rate* | **packet-based** *rate* [*interval*]}	Enables NetFlow flow sampling and configures the rate. For packet-based sampling, optionally configures the export interval.
Step 2	Router(config)# **end**	Exits configuration mode.

When you configure NetFlow flow sampling globally, note the following information:

- The valid values for *rate* are 64, 128, 256, 512, 1024, 2048, 4096, and 8192.
- The valid values for the packet-based export *interval* are from 8,000 through 16,000.
- To export any data, you must also configure NetFlow flow sampling on a Layer 3 interface.

Configuring NetFlow Flow Sampling on a Layer 3 Interface

Note

- With the full-interface or destination-source-interface flow masks, you can enable or disable NetFlow flow sampling on individual Layer 3 interfaces. With all other flow masks, NetFlow flow sampling is enabled or disabled globally.

- The Layer 3 interface must be configured with an IP address.

To configure NetFlow flow sampling on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {vlan vlan_ID | type slot/port}`	Selects a Layer 3 interface to configure.
Step 2	`Router(config-if)# mls netflow sampling`	Enables NetFlow flow sampling on the Layer 3 interface.
Step 3	`Router(config)# end`	Exits configuration mode.

This example shows how to enable NetFlow flow sampling on Fast Ethernet port 5/12:

```
Router# configure terminal
Router(config)# interface fastethernet 5/12
Router(config-if)# mls netflow sampling
Router(config)# end
Router#
```

Configuring NDE on the RP

These sections describe how to configure NDE on the RP:

- Configuring the RP NDE Source Layer 3 Interface, page 64-13
- Configuring the NDE Destination, page 64-14
- Configuring NetFlow Sampling, page 64-14

Configuring the RP NDE Source Layer 3 Interface

To configure the Layer 3 interface used as the source of the NDE packets containing statistics from the RP, perform this task:

Command	Purpose
`Router(config)# ip flow-export source {{vlan vlan_ID} | {type slot/port} | {port-channel number} | {loopback number}}`	Configures the interface used as the source of the NDE packets containing statistics from the RP.

When configuring the RP NDE source Layer 3 interface, note the following information:

- You must select an interface configured with an IP address.
- You can use a loopback interface.

This example shows how to configure a loopback interface as the NDE flow source:

```
Router(config)# ip flow-export source loopback 0
Router(config)#
```

Configuring the NDE Destination

To configure the destination IP address and UDP port to receive the NDE statistics, perform this task:

Command	Purpose
`Router(config)# ip flow-export destination ip_address udp_port_number [vrf vrf_name]`	Configures the NDE destination IP address and UDP port. (Optional) Specify a VPN routing/forwarding table name.

Note NetFlow Multiple Export Destinations—To configure redundant NDE data streams, which improves the probability of receiving complete NetFlow data, you can enter the **ip flow-export destination** command twice and configure a different destination IP address in each command. Configuring two destinations increases the RP CPU utilization, as you are exporting the data records twice.

This example shows how to configure the NDE flow destination IP address and UDP port:

```
Router(config)# ip flow-export destination 172.20.52.37 200
```

Note The destination address and UDP port number are saved in NVRAM and are preserved if NDE is disabled and reenabled or if the switch is power cycled. If you are using the NetFlow FlowCollector application for data collection, verify that the UDP port number you configure is the same port number shown in the FlowCollector's /opt/csconfc/config/nfconfig.file file.

Configuring NetFlow Sampling

The RP supports NetFlow sampling for software-routed traffic.

For additional information, see the *Cisco IOS NetFlow Configuration Guide*.

Enabling NDE for Ingress-Bridged IP Traffic

Except in PFC3A mode, NDE supports ingress-bridged IP traffic. PFC3A mode does not support NDE for bridged IP traffic.

NDE is enabled by default when you enable NetFlow on the VLAN. For additional information, see "Configuring NetFlow on Layer 3 Interfaces" section on page 63-13.

To disable NDE for ingress-bridged IP traffic in VLANs, perform this task:

Command	Purpose
`Router(config)# ip flow export layer2-switched vlan vlan_ID[-vlan_ID] [, vlan_ID[-vlan_ID]]`	Enables NDE for ingress-bridged IP traffic in the specified VLANs (enabled by default when you enter the **ip flow ingress layer2-switched vlan** command). **Note** NDE for ingress-bridged IP traffic in a VLAN requires that NDE on the PFC be enabled with the **mls nde sender** command.

This example shows how to enable NDE for ingress bridged IP traffic in VLAN 200:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip flow export layer2-switched vlan 200
```

Displaying the NDE Address and Port Configuration

To display the NDE address and port configuration, perform these tasks:

Command	Purpose
Router# **show mls nde**	Displays NDE information for hardware flows including the NDE export flow IP address, UDP port, and the NDE source interface configuration.
Router# **show ip flow export**	Displays NDE information for software flows including the NDE export flow IP address, UDP port, and the NDE source interface configuration.

This example shows how to display the NDE export flow source IP address and UDP port configuration:

```
Router# show mls nde
Netflow Data Export enabled
 Exporting flows to  10.34.12.245 (9999)
 Exporting flows from 10.6.58.7 (55425)
 Version: 7
 Include Filter not configured
 Exclude Filter is:
   source:  ip address 11.1.1.0, mask 255.255.255.0
 Total Netflow Data Export Packets are:
    49 packets, 0 no packets, 247 records
 Total Netflow Data Export Send Errors:
        IPWRITE_NO_FIB = 0
        IPWRITE_ADJ_FAILED = 0
        IPWRITE_PROCESS = 0
        IPWRITE_ENQUEUE_FAILED = 0
        IPWRITE_IPC_FAILED = 0
        IPWRITE_OUTPUT_FAILED = 0
        IPWRITE_MTU_FAILED = 0
        IPWRITE_ENCAPFIX_FAILED = 0
 Netflow Aggregation Enabled
   source-prefix aggregation export is disabled
   destination-prefix aggregation exporting flows to 10.34.12.245 (9999)
10.34.12.246 (9909)
      exported 84 packets, 94 records
   prefix aggregation export is disabled
Router#
```

This example shows how to display the NDE export flow IP address, UDP port, and the NDE source interface configuration:

```
Router# show ip flow export
Flow export is enabled
  Exporting flows to 172.20.52.37 (200)
  Exporting using source interface FastEthernet5/8
  Version 1 flow records
  0 flows exported in 0 udp datagrams
  0 flows failed due to lack of export packet
  0 export packets were sent up to process level
```

```
  0 export packets were dropped due to no fib
  0 export packets were dropped due to adjacency issues
Router#
```

Configuring NDE Flow Filters

These sections describe NDE flow filters:

NDE Flow Filter Overview

By default, all expired flows are exported until you configure a filter. After you configure a filter, only expired and purged flows matching the specified filter criteria are exported. Filter values are stored in NVRAM and are not cleared when NDE is disabled.

To display the configuration of the NDE flow filters you configure, use the **show mls nde** command described in the "Displaying the NDE Configuration" section on page 64-17.

Configuring a Port Flow Filter

To configure a destination or source port flow filter, perform this task:

Command	Purpose
Router(config)# **mls nde flow** {**exclude** | **include**} {**dest-port** *number* | **src-port** *number*}	Configures a port flow filter for an NDE flow.

This example shows how to configure a port flow filter so that only expired flows to destination port 23 are exported (assuming the flow mask is set to full):

```
Router(config)# mls nde flow include dest-port 23
Router(config)#
```

Configuring a Host and Port Filter

To configure a host and TCP/UDP port flow filter, perform this task:

Command	Purpose
Router(config)# **mls nde flow** {**exclude** | **include**} {**destination** *ip_address mask* | **source** *ip_address mask* {**dest-port** *number* | **src-port** *number*}}	Configures a host and port flow filter for an NDE flow.

This example shows how to configure a source host and destination TCP/UDP port flow filter so that only expired flows from host 171.69.194.140 to destination port 23 are exported (assuming the flow mask is set to ip-flow):

```
Router(config)# mls nde flow include source 171.69.194.140 255.255.255.255 dest-port 23
```

Configuring a Host Flow Filter

To configure a destination or source host flow filter, perform this task:

Command	Purpose
Router(config)# **mls nde flow** {**exclude** | **include**} {**destination** *ip_address mask* | **source** *ip_address mask* | **protocol** {**tcp** {**dest-port** *number* | **src-port** *number*} | **udp** {**dest-port** *number* | **src-port** *number*}}	Configures a host flow filter for an NDE flow.

This example shows how to configure a host flow filter to export only flows to destination host 172.20.52.37:

```
Router(config)# mls nde flow include destination 172.20.52.37 255.255.255.225
Router(config)#
```

Configuring a Protocol Flow Filter

To configure a protocol flow filter, perform this task:

Command	Purpose
Router(config)# **mls nde flow** {**exclude** | **include**} **protocol** {**tcp** {**dest-port** *number* | **src-port** *number*} | **udp** {**dest-port** *number* | **src-port** *number*}}	Configures a protocol flow filter for an NDE flow.

This example shows how to configure a TCP protocol flow filter so that only expired flows from destination port 35 are exported:

```
Router(config)# mls nde flow include protocol tcp dest-port 35
Router(config)#
```

To display the status of the NDE flow filters, use the **show mls nde** command described in the "Displaying the NDE Configuration" section on page 64-17.

Displaying the NDE Configuration

To display the NDE configuration, perform this task:

Command	Purpose
Router# **show mls nde**	Displays the NDE configuration.

This example shows how to display the NDE configuration:

```
Router# show mls nde
```

```
 Netflow Data Export enabled
 Exporting flows to  10.34.12.245 (9988)  10.34.12.245 (9999)
 Exporting flows from 10.6.58.7 (57673)
 Version: 7
 Include Filter not configured
 Exclude Filter not configured
 Total Netflow Data Export Packets are:
    508 packets, 0 no packets, 3985 records
 Total Netflow Data Export Send Errors:
        IPWRITE_NO_FIB = 0
        IPWRITE_ADJ_FAILED = 0
        IPWRITE_PROCESS = 0
        IPWRITE_ENQUEUE_FAILED = 0
        IPWRITE_IPC_FAILED = 0
        IPWRITE_OUTPUT_FAILED = 0
        IPWRITE_MTU_FAILED = 0
        IPWRITE_ENCAPFIX_FAILED = 0
 Netflow Aggregation Enabled
Router#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Part 11

Network Management

Configuring Call Home

This chapter describes how to configure the Call Home feature in Cisco IOS Release 12.2SX. Release 12.2(33)SXH and later releases support the Call Home feature.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter includes the following sections:

Understanding Call Home

Call Home provides email-based and web-based notification of critical system events. A versatile range of message formats are available for optimal compatibility with pager services, standard email, or XML-based automated parsing applications. Common uses of this feature may include direct paging of a network support engineer, email notification to a Network Operations Center, XML delivery to a support website, and utilization of Cisco Smart Call Home services for direct case generation with the Cisco Systems Technical Assistance Center (TAC).

The Call Home feature can deliver alert messages containing information on configuration, diagnostics, environmental conditions, inventory, and syslog events.

The Call Home feature can deliver alerts to multiple recipients, referred to as *Call Home destination profiles*, each with configurable message formats and content categories. A predefined destination profile (CiscoTAC-1) is provided, and you also can define your own destination profiles. The CiscoTAC-1 profile is used to send alerts to the backend server of the Smart Call Home service, which can be used to create service requests to the Cisco TAC (depending on the Smart Call Home service support in place for your device and the severity of the alert).

Flexible message delivery and format options make it easy to integrate specific support requirements. If multiple destination profiles are configured, and one fails, the system will try every configured profile before sending a failure message.

The Call Home feature provides these functions:

- Multiple message-format options:
  - Short Text—Suitable for pagers or printed reports.
  - Plain Text—Full formatted message information suitable for human reading.
  - XML—Machine readable format using Extensible Markup Language (XML) and Adaptive Markup Language (AML) document type definitions (DTDs). The XML format enables communication with the Cisco Smart Call Home server.
- Multiple concurrent message destinations.
- Multiple message categories including configuration, diagnostics, environmental conditions, inventory, and syslog events.
- Filtering of messages by severity and pattern matching.
- Scheduling of periodic message sending.
- Continuous device health monitoring and real-time diagnostics alerts.
- Analysis of Call Home messages from your device and, where supported, Automatic Service Request generation, routed to the appropriate TAC team, including detailed diagnostic information to speed problem resolution.
- Secure message transport directly from your device or through a downloadable Transport Gateway (TG) aggregation point. You can use a TG aggregation point in cases requiring support for multiple devices or in cases where security requirements mandate that your devices may not be connected directly to the Internet.
- Web-based access to Call Home messages and recommendations, inventory and configuration information for all Call Home devices that provides access to associated Field Notices, Security Advisories and End-of-Life Information.

Obtaining Smart Call Home

If you have a service contract directly with Cisco Systems, you can register your Call Home devices for the Cisco Smart Call Home service. Smart Call Home provides fast resolution of system problems by analyzing Call Home messages sent from your devices and providing background information and recommendations. For issues that can be identified as known, particularly GOLD diagnostics failures, depending on the Smart Call Home service support in place for your device and the severity of the alert, Automatic Service Requests will be generated with the Cisco TAC.

You need the following items to register:

- The SMARTnet contract number for your switch.
- Your email address
- Your Cisco.com ID

For detailed information on Smart Call Home, see the Smart Call Home page at this location:

http://www.cisco.com/en/US/products/ps7334/serv_home.html

Default Settings

Table 65-1 lists the default Call Home settings.

Table 65-1 Default Call Home Settings

Parameters	Default
Call Home feature status	Disabled
User-defined profile status	Active
Predefined Cisco TAC profile status	Inactive
Transport method	Email
Message format type	XML
Destination message size for a message sent in long text, short text, or XML format	3,145,728
Alert group status	Enabled
Call Home message severity threshold	0 (debugging)
Message rate limit for messages per minute	20

Configuring Call Home

These sections provide an overview of Call Home configuration:

- Configuration Overview, page 65-4
- Configuring Customer Contact Information, page 65-4
- Configuring Destination Profiles, page 65-5
- Subscribing to Alert Groups, page 65-13
- Enabling Call Home, page 65-16

- Testing Call Home Communications, page 65-16
- Configuring and Enabling Smart Call Home, page 65-19

Configuration Overview

Consider these items before you configure Call Home:

- Obtain customer email, phone, and street address information for the Call Home contact to be configured so that the receiver can determine the origin of messages received.
- If using email message delivery, identify the name or IPv4 address of a primary Simple Mail Transfer Protocol (SMTP) server and any backup servers.
- If using secure HTTP (HTTPS) message delivery, configure a trustpoint certificate authority (CA) . For example, this procedure is required if you are using the HTTPS server for Cisco Smart Call Home Service in the CiscoTAC-1 profile for Call Home.
- Verify IP connectivity from the router to the email server(s) or the destination HTTP server.
- If servers are specified by name, the switch must have IP connectivity to a domain name server.
- If using Cisco Smart Call Home, verify that an active service contract exists for the device being configured.

Tip From the Smart Call Home web application, you can download a basic configuration script to assist you in the configuration of the Call Home feature for use with Smart Call Home and the Cisco TAC. The script will also assist in configuring the trustpoint CA for secure communications with the Smart Call Home service. The script, provided on an as-is basis, can be downloaded from this URL: http://www.cisco.com/en/US/products/ps7334/serv_home.html

Configuring Customer Contact Information

These are customer contact information items:

- Email address (required)
- Phone number (optional)
- Street address (optional)
- Contract ID (optional)
- Customer ID (optional)
- Site ID (optional)

To configure the customer contact information, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters configuration mode.
Step 2	`Router(config)#` **`call-home`**	Enters Call Home configuration mode.
Step 3	`Router(cfg-call-home)#` **`contact-email-addr`** *`email-address`*	Assigns the customer's email address. Enter up to 200 characters in email address format with no spaces.

	Command	Purpose
Step 4	`Router(cfg-call-home)# phone-number +phone-number`	(Optional) Assigns the customer's phone number. **Note** The number must begin with a plus (**+**) prefix, and may contain only dashes (-) and numbers. Enter up to 16 characters. If you include spaces, you must enclose your entry in quotes ("").
Step 5	`Router(cfg-call-home)# street-address street-address`	(Optional) Assigns the customer's street address where RMA equipment can be shipped. Enter up to 200 characters. If you include spaces, you must enclose your entry in quotes ("").
Step 6	`Router(cfg-call-home)# customer-id text`	(Optional) Identifies the customer ID. Enter up to 64 characters. If you include spaces, you must enclose your entry in quotes ("").
Step 7	`Router(cfg-call-home)# site-id text`	(Optional) Identifies the customer site ID. Enter up to 200 characters. If you include spaces, you must enclose your entry in quotes ("").
Step 8	`Router(cfg-call-home)# contract-id text`	(Optional) Identifies the customer's contract ID for the switch. Enter up to 64 characters. If you include spaces, you must enclose your entry in quotes ("").

This example shows the configuration of contact information:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# call-home
Router(cfg-call-home)# contact-email-addr username@example.com
Router(cfg-call-home)# phone-number +1-800-555-4567
Router(cfg-call-home)# street-address "1234 Picaboo Street, Any city, Any state, 12345"
Router(cfg-call-home)# customer-id Customer1234
Router(cfg-call-home)# site-id Site1ManhattanNY
Router(cfg-call-home)# contract-id Company1234
Router(cfg-call-home)# exit
Router(config)#
```

Configuring Destination Profiles

These sections describe destination profiles:

- Destination Profile Overview, page 65-6
- Configuring Call Home to Use VRF, page 65-6
- Configuring a Destination Profile to Send Email Messages, page 65-7
- Configuring a Destination Profile to Send HTTP Messages, page 65-9
- Configuring Call Home Traffic Rate Limiting, page 65-11
- Destination Profile Management, page 65-11

Destination Profile Overview

A destination profile contains the required delivery information for an alert notification. At least one destination profile is required. You can configure multiple destination profiles of one or more types.

You can use the predefined destination profile or define a desired profile. If you define a new destination profile, you must assign a profile name.

You can configure the following attributes for a destination profile:

- Profile name—A string that uniquely identifies each user-defined destination profile. The profile name is limited to 31 characters and is not case-sensitive. You cannot use **all** as a profile name.
- Transport method—The transport mechanism, either email or HTTP (including HTTPS), for delivery of alerts.
  - For user-defined destination profiles, email is the default, and you can enable either or both transport mechanisms. If you disable both methods, email will be enabled.
  - For the predefined Cisco TAC profile, you can enable either transport mechanism, but not both.
- Destination address—The actual address related to the transport method to which the alert should be sent.
- Message formatting—The message format used for sending the alert.
  - For user-defined destination profiles, the format options are long-text, short-text, or XML. The default is XML.
  - For the predefined Cisco TAC profile, only XML is allowed.
- Message size—The maximum destination message size. The valid range is 50 to 3,145,728 bytes and the default is 3,145,728 bytes.

Note

- The Call Home feature provides a predefined profile named CiscoTAC-1 that is inactive by default. The CiscoTAC-1 profile is intended for use with the Smart Call Home service, which requires certain additional configuration steps to enable the service with the Call Home feature. For more information about this profile, see the "Using the Predefined CiscoTAC-1 Destination Profile" section on page 65-13.
- If you use the Cisco Smart Call Home service, the destination profile must use the XML message format.

Configuring Call Home to Use VRF

To configure Call Home to use a VRF interface for Call Home email or for HTTP messages, perform this task:

	Command or Action	Purpose
Step 1	`Router#` **`configure terminal`**	Enters configuration mode.
Step 1	`Router(config)#` **`interface`** *`type`*	Selects an interface to configure.
Step 2	`Router(config-if)#` **`ip address`** *`ip_address mask`*	Assigns an IP address and subnet mask to the interface.

	Command or Action	Purpose
Step 3	`Router(config-if)# vrf forwarding call_home_vrf_name`	Associates the *call_home_vrf_name* VRF with the interface.
Step 4	`Router(config-if)# exit`	Exits interface configuration mode.

This example shows how to configure Call Home to use a VRF interface:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# ip address 10.10.10.10 0.0.0.0
Router(config-if)# vrf forwarding call_home_vrf
Router(config-if)# exit
Router(config)#
```

Configuring a Destination Profile to Send Email Messages

To configure Call Home to send email messages, complete the following tasks:

- Configuring Call Home to Use VRF for Email Messages, page 65-7 (optional)
- Configuring the Mail Server, page 65-8 (required)
- Configuring a Destination Profile for Email, page 65-8 (required)
- Configuring Other Email Options, page 65-9 (optional)

Note To send Call Home email messages through a VRF interface, configure Call Home to use VRF (see "Configuring Call Home to Use VRF" section on page 65-6).

Configuring Call Home to Use VRF for Email Messages

To configure Call Home to use a VRF instance for Call Home email messages, perform this task:

	Command or Action	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# call-home`	Enters Call Home configuration submode.
Step 3	`Router(cfg-call-home)# vrf call_home_vrf_name`	Specifies the VRF instance to use for Call Home email messages. **Note** Release 12.2(33)SXI1 and later releases support VRF configuration for Call Home email messages.

This example shows how to configure Call Home to use a VRF interface:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# call-home
Router(cfg-call-home)# vrf call_home_vrf
Router(cfg-call-home)# exit
Router(config)#
```

Configuring the Mail Server

To use the email message transport, perform this task:

	Command or Action	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	`Router(cfg-call-home)#` **`mail-server`** `{`*`ipv4-address`* `|` *`name`*`}` **`priority`** *`number`*	Specifies an email server and its relative priority among configured email servers, where: • *ipv4-address*—Specifies the IPv4 address of the mail server. • *name*—Specifies the mail server's fully qualified domain name (FQDN) of 64 characters or less. • *number*—Assigns a number between 1 (highest priority) and 100 (lowest priority). Higher priority (lower priority numbers) are tried first. • Repeat to define backup email servers (maximum four backup email servers, for a total of five email servers.

The following example shows the configuration of a primary mail server (named "smtp.example.com") and secondary mail server at IP address 192.168.0.1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# call-home
Router(cfg-call-home)# mail-server smtp.example.com priority 1
Router(cfg-call-home)# mail-server 192.168.0.1 priority 2
Router(cfg-call-home)# exit
Router(config)#
```

Configuring a Destination Profile for Email

To configure a destination profile for email transport, complete this task:

	Command or Action	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	`Router(config-call-home)#` **`profile`** *`name`*	Enters call home destination profile configuration mode for the specified destination profile name. If the specified destination profile does not exist, it is created.
Step 4	`Router(cfg-call-home-profile)#` **`destination transport-method email`**	Configures the message transport method for email. (This is the default.)
Step 5	`Router(cfg-call-home-profile)#` **`destination address email`** *`email_address`*	Configures the destination email address to which Call Home messages are sent.
Step 6	`Router(cfg-call-home-profile)#` **`destination preferred-msg-format`** `{`**`long-text`** `|` **`short-text`** `|` **`xml`**`}`	(Optional) Configures a preferred message format. The default is XML.

	Command or Action	Purpose
Step 7	`Router(cfg-call-home-profile)#` **`destination message-size`** *`bytes`*	(Optional) Configures a maximum destination message size (from 50 to 3145728 bytes) for the destination profile. The default is 3145728 bytes.
Step 8	`Router(cfg-call-home-profile)#` **`active`**	(Optional) Enables the destination profile. By default, a user-defined profile is enabled when it is created.
Step 9	`Router(cfg-call-home-profile)#` **`exit`**	Exits call home destination profile configuration mode and returns to call home configuration mode.
Step 10	`Router(cfg-call-home)#` **`end`**	Returns to privileged EXEC mode.

Configuring Other Email Options

To configure other email options, perform this task:

	Command or Action	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	`Router(cfg-call-home)#` **`sender from`** *`email-address`*	(Optional) Assigns the email address that will appear in the from field in Call Home email messages. If no address is specified, the contact email address is used.
Step 4	`Router(cfg-call-home)#` **`sender reply-to`** *`email-address`*	(Optional) Assigns the email address that will appear in the reply-to field in Call Home email messages.
Step 5	`Router(cfg-call-home)#` **`source-ip-address`** *`ip_address`*	(Optional; supported in Release 12.2(33)SXI and later releases) Assigns the source IP address that will be used for Call Home email messages.

This example shows how to configure the email options:

```
Router(cfg-call-home)# sender from username@example.com
Router(cfg-call-home)# sender reply-to username@example.com
Router(cfg-call-home)# source-ip-address 10.10.10.10
```

Configuring a Destination Profile to Send HTTP Messages

To configure Call Home to send HTTP (or HTTPS) messages, complete the following tasks:

- Configuring the HTTP Source Interface, page 65-10
- Configuring a Destination Profile for HTTP, page 65-10
- Configuring a Trustpoint Certificate Authority, page 65-11 (required for HTTPS)

Configuring the HTTP Source Interface

To configure an HTTP client source interface, perform this task:

	Command or Action	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`ip http client source-interface`** *`type number`*	Configures the source interface for the HTTP client. If the interface is associated with a VRF instance, the HTTP messages use the VRF instance.

Configuring a Destination Profile for HTTP

To configure a destination profile for HTTP transport, perform this task:

	Command or Action	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`call-home`**	Enters call home configuration mode.
Step 3	`Router(config-call-home)#` **`profile`** *`name`*	Enters call home destination profile configuration mode for the specified destination profile. If the specified destination profile does not exist, it is created.
Step 4	`Router(cfg-call-home-profile)#` **`destination transport-method http`**	Enables the HTTP message transport method.
Step 5	`Router(cfg-call-home-profile)#` **`destination address http`** *`url`*	Configures the destination URL to which Call Home messages are sent. **Note** When entering a destination URL, include either **http://** or **https://**, depending on whether the server is a secure server. If the destination is a secure server, you must also configure a trustpoint CA.
Step 6	`Router(cfg-call-home-profile)#` **`destination preferred-msg-format`** {**`long-text`** | **`short-text`** | **`xml`**}	(Optional) Configures a preferred message format. The default is XML.
Step 7	`Router(cfg-call-home-profile)#` **`destination message-size`** *`bytes`*	(Optional) Configures a maximum destination message size for the destination profile.
Step 8	`Router(cfg-call-home-profile)#` **`active`**	Enables the destination profile. By default, a profile is enabled when it is created.
Step 9	`Router(cfg-call-home-profile)#` **`exit`**	Exits call home destination profile configuration mode and returns to call home configuration mode.
Step 10	`Router(cfg-call-home)#` **`end`**	Returns to privileged EXEC mode.

This example shows how to configure a destination profile for HTTP transport:

```
Router# configure terminal
Router(config)# call-home
Router(config-call-home)# profile test
Router(cfg-call-home-profile)# destination transport-method http
Router(cfg-call-home-profile)# destination address http https://example.url.com
Router(cfg-call-home-profile)# destination preferred-msg-format xml
Router(cfg-call-home-profile)# destination message-size 3,145,728
Router(cfg-call-home-profile)# active
```

```
Router(cfg-call-home-profile)# exit
Router(cfg-call-home)# end
```

Configuring a Trustpoint Certificate Authority

If you are using the HTTP transport method and specifying an HTTPS destination URL, then you will also need to configure a trustpoint certificate authority (CA). See the "Declare and Authenticate a CA Trustpoint" section on page 65-21.

Configuring Call Home Traffic Rate Limiting

To configure Call Home traffic rate limiting, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# call-home`	Enters Call Home configuration submode.
Step 3	`Router(cfg-call-home)# rate-limit number`	(Optional) Specifies a limit on the number of messages sent per minute, from 1 to 60. The default is 20.

This example shows how to configure Call Home traffic rate limiting:

```
Router# configure terminal
Router(config)# call-home
Router(config-call-home)# profile test
Router(cfg-call-home-profile)# rate-limit 20
```

Destination Profile Management

These sections describe destination profile management:

Activating and Deactivating a Destination Profile

Except for the predefined CiscoTAC-1 profile, all Call Home destination profiles are automatically activated when you create them. If you do not want to use a profile right way, you can deactivate the profile. The CiscoTAC-1 profile is inactive by default and must be activated to be used.

To activate or deactivate a destination profile, perform this task:

	Command or Action	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# call-home`	Enters call home configuration mode.

	Command or Action	Purpose
Step 3	`Router(config-call-home)# profile name`	Enters call home destination profile configuration mode for the specified destination profile. If the specified destination profile does not exist, it is created.
Step 4	`Router(cfg-call-home-profile)# active`	Enables the destination profile. By default, a new profile is enabled when it is created.
Step 5	`Router(cfg-call-home-profile)# no active`	Disables the destination profile.
Step 6	`Router(cfg-call-home)# end`	Exits call home destination profile configuration mode and returns to privileged EXEC mode.

This example shows how to activate a destination profile:

```
Router# configure terminal
Router(config)# call-home
Router(config-call-home)# profile test
Router(cfg-call-home-profile)# active
Router(cfg-call-home)# end
```

This example shows how to deactivate a destination profile:

```
Router# configure terminal
Router(config)# call-home
Router(config-call-home)# profile test
Router(cfg-call-home-profile)# no active
Router(cfg-call-home)# end
```

Copying a Destination Profile

To create a new destination profile by copying an existing profile, perform this task:

	Command or Action	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# call-home`	Enters call home configuration mode.
Step 3	`Router(cfg-call-home)# copy profile source_profile target_profile`	Creates a new destination profile with the same configuration settings as the existing destination profile, where: • *source_profile*—Specifies the existing name of the profile. • *target_profile*—Specifies a name for the new copy of the profile.

This example shows how to activate a destination profile:

```
Router# configure terminal
Router(config)# call-home
Router(config-call-home)# profile test
Router(cfg-call-home-profile)# copy profile profile1 profile2
```

Renaming a Destination Profile

To change the name of an existing profile, perform this task:

	Command or Action	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# call-home`	Enters call home configuration mode.
Step 3	`Router(cfg-call-home)# rename profile source_profile target_profile`	Renames an existing source file, where: • *source_profile*—Specifies the existing name of the profile. • *target_profile*—Specifies a new name for the existing profile.

This example shows how to activate a destination profile:

```
Router# configure terminal
Router(config)# call-home
Router(config-call-home)# profile test
Router(cfg-call-home-profile)# rename profile profile1 profile2
```

Using the Predefined CiscoTAC-1 Destination Profile

The CiscoTAC-1 profile is automatically configured in the Call Home feature for your use with the Cisco Smart Call Home service. This profile includes certain information, such as the destination email address and HTTPS URL, and default alert groups for communication with the Smart Call Home service. Some of these attributes, such as the destination email address, HTTPS URL, and message format cannot be modified.

You can use either email or http transport to communicate with the Smart Call Home service backend server. By default, the CiscoTAC-1 profile is inactive and uses email as the default transport method. To use email transport, you only need to enable the profile. However, to use this profile with the Cisco Smart Call Home service secure server (via HTTPS), you not only must enable the profile, but you must also change the transport method to HTTP as shown in the following example:

```
Router# configure terminal
Router(config)# call-home
Router(config-call-home)# profile CiscoTAC-1
Router(cfg-call-home-profile)# destination transport-method http
Router(cfg-call-home-profile)# active
```

For more information about additional requirements for Configuring the Smart Call Home service, see the "Configuring and Enabling Smart Call Home" section on page 65-19.

Verifying the Call Home Profile Configuration

To verify the profile configuration for Call Home, use the **show call-home profile** command. See the "Displaying Call Home Configuration Information" section on page 65-23 for more information and examples.

Subscribing to Alert Groups

These sections describe subscribing to alert groups:

Overview of Alert Group Subscription

An alert group is a predefined subset of Call Home alerts supported in all switches. Different types of Call Home alerts are grouped into different alert groups depending on their type. These alert groups are available:

Configuration

Diagnostic

- Environment
- Inventory
- Syslog

The triggering events for each alert group are listed in the "Alert Group Trigger Events and Commands" section on page 65-26, and the contents of the alert group messages are listed in the "Message Contents" section on page 65-33.

You can select one or more alert groups to be received by a destination profile.

> **Note** A Call Home alert is only sent to destination profiles that have subscribed to the alert group containing that Call Home alert. In addition, the alert group must be enabled.

Configuring Alert Group Subscription

To subscribe a destination profile to an alert group, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters configuration mode.
Step 2	`Router(config)#` **`call-home`**	Enters Call Home configuration submode.
Step 3	`Router(cfg-call-home)#` **`alert-group`** `{`**`all`** `|` **`configuration`** `|` **`diagnostic`** `|` **`environment`** `|` **`inventory`** `|` **`syslog`**`}`	Enables the specified alert group. Use the keyword **all** to enable all alert groups. By default, all alert groups are enabled.
Step 4	`Router(cfg-call-home)#` **`profile`** *`name`*	Enters the Call Home destination profile configuration submode for the specified destination profile.
Step 5	`Router(cfg-call-home-profile)#` **`subscribe-to-alert-group configuration`** `[`**`periodic`** `{`**`daily`** *`hh:mm`* `|` **`monthly`** *`date hh:mm`* `|` **`weekly`** *`day hh:mm`*`}]`	Subscribes this destination profile to the Configuration alert group. The Configuration alert group can be configured for periodic notification, as described in the "Configuring Periodic Notification" section on page 65-15.
Step 6	`Router(cfg-call-home-profile)#` **`subscribe-to-alert-group all`**	Subscribes to all available alert groups.

	Command	Purpose
Step 7	`Router(cfg-call-home-profile)#` **`subscribe-to-alert-group diagnostic`** `[`**`severity`** `{`**`catastrophic`** `|` **`critical`** `|` **`debugging`** `|` **`disaster`** `|` **`fatal`** `|` **`major`** `|` **`minor`** `|` **`normal`** `|` **`notification`** `|` **`warning`**`}]`	Subscribes this destination profile to the Diagnostic alert group. The Diagnostic alert group can be configured to filter messages based on severity, as described in the "Configuring Message Severity Threshold" section on page 65-15.
Step 8	`Router(cfg-call-home-profile)#` **`subscribe-to-alert-group environment`** `[`**`severity`** `{`**`catastrophic`** `|` **`critical`** `|` **`debugging`** `|` **`disaster`** `|` **`fatal`** `|` **`major`** `|` **`minor`** `|` **`normal`** `|` **`notification`** `|` **`warning`**`}]`	Subscribes this destination profile to the Environment alert group. The Environment alert group can be configured to filter messages based on severity, as described in the "Configuring Message Severity Threshold" section on page 65-15.
Step 9	`Router(cfg-call-home-profile)#` **`subscribe-to-alert-group inventory`** `[`**`periodic`** `{`**`daily`** *`hh:mm`* `|` **`monthly`** *`date hh:mm`* `|` **`weekly`** *`day hh:mm`*`}]`	Subscribes this destination profile to the Inventory alert group. The Inventory alert group can be configured for periodic notification, as described in the "Configuring Periodic Notification" section on page 65-15.
Step 10	`Router(cfg-call-home-profile)#` **`subscribe-to-alert-group syslog`** `[`**`severity`** `{`**`catastrophic`** `|` **`disaster`** `|` **`fatal`** `|` **`critical`** `|` **`major`** `|` **`minor`** `|` **`warning`** `|` **`notification`** `|` **`normal`** `|` **`debugging`**`} [`**`pattern`** *`string`*`]]`	Subscribes this destination profile to the Syslog alert group. The Syslog alert group can be configured to filter messages based on severity, as described in the "Configuring Message Severity Threshold" section on page 65-15. You can specify a pattern to be matched in the syslog message, as described in the "Configuring Syslog Pattern Matching" section on page 65-16. If the pattern contains spaces, you must enclose it in quotes ("").
Step 11	`Router(cfg-call-home-profile)#` **`exit`**	Exits the Call Home destination profile configuration submode.

Configuring Periodic Notification

When you subscribe a destination profile to either the Configuration or the Inventory alert group (see the "Configuring Alert Group Subscription" section on page 65-14), you can choose to receive the alert group messages asynchronously or periodically at a specified time. The sending period can be one of the following:

- Daily—Specify the time of day to send, using an hour:minute format *hh:mm*, with a 24-hour clock (for example, 14:30).
- Weekly—Specify the day of the week and time of day in the format *day hh:mm*, where the day of the week is spelled out (for example, monday).
- Monthly—Specify the numeric date, from 1 to 31, and the time of day, in the format *date hh:mm*.

Configuring Message Severity Threshold

When you subscribe a destination profile to the Diagnostic, Environment, or Syslog alert group (see the "Configuring Alert Group Subscription" section on page 65-14), you can set a threshold for the sending of alert group messages based on the message's level of severity. Any message with a value lower than the destination profile's specified threshold is not sent to the destination.

The severity threshold is configured using the keywords in Table 65-2, and ranges from catastrophic (level 9, highest level of urgency) to debugging (level 0, lowest level of urgency). If no severity threshold is configured, the default is debugging (level 0).

Note Call Home severity levels are not the same as system message logging severity levels.

Table 65-2 Severity and Syslog Level Mapping

Level	Keyword	Syslog Level	Description
9	**catastrophic**	N/A	Network-wide catastrophic failure.
8	**disaster**	N/A	Significant network impact.
7	**fatal**	Emergency (0)	System is unusable.
6	**critical**	Alert (1)	Critical conditions, immediate attention needed.
5	**major**	Critical (2)	Major conditions.
4	**minor**	Error (3)	Minor conditions.
3	**warning**	Warning (4)	Warning conditions.
2	**notification**	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	**normal**	Information (6)	Normal event signifying return to normal state.
0	**debugging**	Debug (7)	Debugging messages.

Configuring Syslog Pattern Matching

When you subscribe a destination profile to the Syslog alert group (see the "Configuring Alert Group Subscription" section on page 65-14), you can optionally specify a text pattern to be matched within each syslog message. If you configure a pattern, a Syslog alert group message will be sent only if it contains the specified pattern and meets the severity threshold. If the pattern contains spaces, you must enclose it in quotes ("") when configuring it. You can specify up to five patterns for each destination profile.

Enabling Call Home

To enable or disable the Call Home feature, perform this task:

	Command	Purpose
	Command	**Purpose**
Step 1	`Router#` **`configure terminal`**	Enters configuration mode.
Step 2	`Router(config)#` **`service call-home`**	Enables the Call Home feature.

Testing Call Home Communications

You can test Call Home communications by sending messages manually using two command types. To send a user-defined Call Home test message, use the **call-home test** command. To send a specific alert group message, use the **call-home send** command.

These sections describe Call Home communication:

- Sending a Call Home Test Message Manually, page 65-17
- Sending a Call Home Alert Group Message Manually, page 65-17

- Sending a Request for an Analysis and Report, page 65-18
- Sending the Output of a Command, page 65-19

Sending a Call Home Test Message Manually

To manually send a Call Home test message, perform this task:

Command	Purpose
Router# **call-home test** ["*test-message*"] **profile** *name*	Sends a test message to the specified destination profile. The user-defined test message text is optional, but must be enclosed in quotes (“”) if it contains spaces. If no user-defined message is configured, a default message will be sent.

Sending a Call Home Alert Group Message Manually

To manually trigger a Call Home alert group message, perform this task:

	Command	Purpose
Step 1	Router# **call-home send alert-group configuration** [**profile** *name*]	Sends a configuration alert group message to one destination profile if specified, or to all subscribed destination profiles.
Step 2	Router# **call-home send alert-group diagnostic** {**module** *number* | *slot/subslot* | *slot/bay_number* | **switch** *x* **module** *number*} [**profile** *name*]	Sends a diagnostic alert group message to the configured destination profile if specified, or to all subscribed destination profiles. You must specify the module or port whose diagnostic information should be sent. If a virtual switching system (VSS) is used, you must specify the switch and module.
Step 3	Router# **call-home send alert-group inventory** [**profile** *name*]	Sends an inventory alert group message to one destination profile if specified, or to all subscribed destination profiles.

When manually sending Call Home alert group messages, note the following guidelines:

- Only the configuration, diagnostic, and inventory alert groups can be sent manually.
- When you manually trigger a configuration, diagnostic, or inventory alert group message and you specify a destination profile name, a message is sent to the destination profile regardless of the profile’s active status, subscription status, or severity setting.
- When you manually trigger a configuration or inventory alert group message and do not specify a destination profile name, a message is sent to all active profiles that have either a normal or periodic subscription to the specified alert group.
- When you manually trigger a diagnostic alert group message and do not specify a destination profile name, the command will cause the following actions:
  - For any active profile that subscribes to diagnostic events with a severity level of less than minor, a message is sent regardless of whether the module or interface has observed a diagnostic event.

- For any active profile that subscribes to diagnostic events with a severity level of minor or higher, a message is sent only if the specified module or interface has observed a diagnostic event of at least the subscribed severity level; otherwise, no diagnostic message is sent to the destination profile.

Sending a Request for an Analysis and Report

Release 12.2(33)SXI and later releases support Call Home requests. You can use the **call-home request** command to submit information about your system to Cisco in order to receive helpful information specific to your system. You can request a variety of reports, including security alerts, known bugs, best practices, and command references.

To submit a request for report and analysis information from the Cisco Output Interpreter tool, perform this task:

	Command	Purpose
Step 1	`Router#` **`call-home request output-analysis "`***`show-command`***`"`** `[`**`profile`** *`name`*`] [`**`ccoid`** *`user-id`*`]`	Sends the output of the specified show command for analysis. The show command must be contained in quotes ("").
Step 2	`Router#` **`call-home request`** `{`**`config-sanity`** `|` **`bugs-list`** `|` **`command-reference`** `|` **`product-advisory`**`} [`**`profile`** *`name`*`] [`**`ccoid`** *`user-id`*`]`	Sends the output of a predetermined set of commands such as the **show running-config all**, **show version**, and **show module** (standalone) or **show module switch all** (VS system) commands, for analysis. Specifies the type of report requested.

When manually sending a Call Home report and analysis request, note the following guidelines:

- If a **profile** *name* is specified, the request will be sent to the profile. If no profile is specified, the request will be sent to the Cisco TAC profile. The recipient profile does not need to be enabled for the call-home request. The profile should specify the email address where the transport gateway is configured so that the request message can be forwarded to the Cisco TAC and the user can receive the reply from the Smart Call Home service.
- The **ccoid** *user-id* is the registered identifier of the Smart Call Home user. If the *user-id* is specified, the response will be sent to the email address of the registered user. If no *user-id* is specified, the response will be sent to the contact email address of the device.
- Based on the keyword specifying the type of report requested, the following information will be returned:
  - **config-sanity**—Information on best practices as related to the current running configuration.
  - **bugs-list**—Known bugs in the running version and in the currently applied features.
  - **command-reference**—Reference links to all commands in the running configuration.
  - **product-advisory**—Product Security Incident Response Team (PSIRT) notices, End of Life (EOL) or End of Sales (EOS) notices, or field notices (FN) that may affect devices in your network.

This example shows a request for analysis of a user-specified show command:

```
Router# call-home request output-analysis "show diagnostic result module all" profile TG
```

Sending the Output of a Command

You can use the **call-home send** command to execute a CLI command and email the command output to Cisco or to an email address that you specify.

To execute a CLI command and email the command output, perform this task:

Command	Purpose
Router# **call-home send "***command***"** [**email** *email-addr*] [**tac-service-request** *SR*]	Executes the specified CLI command and emails the output.

When sending the output of a command, note the following guidelines:

- The specified CLI command can be any run command, including commands for all modules. The command must be contained in quotes (“”).
- If an email address is specified, the command output will be sent to that address. If no email address is specified, the output will be sent to the Cisco TAC (attach@cisco.com). The email will be sent in long text format with the service number, if specified, in the subject line.
- The service number is required only if no email address is specified, or if a Cisco TAC email address is specified.

This example shows how to send the output of a CLI command to a user-specified email address:

```
Router# call-home send "show diagnostic result module all" email support@example.com
```

Configuring and Enabling Smart Call Home

For application and configuration information of the Cisco Smart Call Home service, see the “Quick Start for Smart Call Home” section in Chapter 1 of the *Smart Call Home User Guide*:

http://www.cisco.com/en/US/docs/switches/lan/smart_call_home/book.html

The user guide includes configuration examples for sending Smart Call Home messages directly from your device or through a transport gateway (TG) aggregation point. You can use a TG aggregation point in cases requiring support for multiple devices or in cases where security requirements mandate that your devices may not be connected directly to the Internet.

Because the Smart Call Home service uses HTTPS as the transport method, you must also configure its CA as a trustpoint, as described in the *Smart Call Home User Guide*.

Configuring the Smart Call Home Service

Tip From the Smart Call Home website, you can download a basic configuration script to assist you in the configuration of the Call Home feature for use with Smart Call Home service and the Cisco TAC. The script also assists in configuring the trustpoint CA for secure communications with the Smart Call Home service. The script, provided on an as-is basis, can be downloaded from a link under the "Smart Call Home Resources" heading at: http://www.cisco.com/en/US/products/ps7334/serv_home.html

This section provides an overview of the minimum steps required to configure the Call Home feature on a Cisco device, and other required supporting configuration to communicate securely with the Smart Call Home service using HTTPS:

- Enabling the Smart Call Home Service, page 65-20
- Declare and Authenticate a CA Trustpoint, page 65-21
- Start Smart Call Home Registration, page 65-22

Enabling the Smart Call Home Service

Note Before you start to configure the Smart Call Home Service, be sure that you have completed the following prerequisites:

- Verify that you have an active Cisco Systems service contract for the device being configured.
- Verify that you have IP connectivity to the Cisco HTTPS server.
- Obtain the latest Cisco Systems server security certificate.

The CiscoTAC-1 profile is predefined in the Call Home feature to communicate using email to the backend server for the Smart Call Home service. The URL to the Cisco HTTPS backend server is also predefined. This profile is inactive by default.

Unlike other profiles that you can configure in Call Home to support both transport methods, the CiscoTAC-1 profile can only use one transport method at a time. To use this profile with the Cisco Smart Call Home HTTPS server, you must change the transport method from email to HTTP and enable the profile. In addition, you must minimally specify a contact email address and enable the Call Home feature.

To enable the Smart Call Home service, perform this task:

	Command or Action	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# call-home`	Enters call home configuration mode.
Step 3	`Router(config-call-home)# profile CiscoTAC-1`	Enters call home destination profile configuration mode for the CiscoTAC-1 destination profile.
Step 4	`Router(cfg-call-home-profile)# destination transport-method http`	(Required for HTTPS) Configures the message transport method for http.
Step 5	`Router(cfg-call-home-profile)# active`	Enables the destination profile.

	Command or Action	Purpose
Step 6	`Router(cfg-call-home-profile)# **exit**`	Exits call home destination profile configuration mode and returns to call home configuration mode.
Step 7	`Router(cfg-call-home)# **contact-email-addr** *customer_email_address*`	Assigns the customer's email address. Enter up to 200 characters in email address format with no spaces.
Step 8	`Router(cfg-call-home)# **exit**`	Exits call home configuration mode and returns to global configuration mode.
Step 9	`Router(config)# **service call-home**`	Enables the Call Home feature.
Step 10	`Router(config)# **exit**`	Exits global configuration mode and returns to privileged EXEC mode.
Step 11	`Router# **copy running-config startup-config**`	Saves the configuration.

This example shows how to enable the Smart Call Home service:

```
Router(cfg-call-home-profile)# destination transport-method http
Router(cfg-call-home-profile)# active
Router(cfg-call-home-profile)# exit
Router(cfg-call-home)# contact-email-addr username@example.com
Router(cfg-call-home)# exit
Router(config)# service call-home
Router(config)# exit
Router# copy running-config startup-config
```

Declare and Authenticate a CA Trustpoint

To declare and authenticate the Cisco server security certificate and establish communication with the Cisco HTTPS server for Smart Call Home service, perform this task:

	Command or Action	Purpose
Step 1	`Router# **configure terminal**`	Enters global configuration mode.
Step 2	`Router(config)# **crypto pki trustpoint** *trustpoint_name*`	Declares a CA trustpoint on your router and enters CA trustpoint configuration mode.
Step 3	`Router(ca-trustpoint)# **enrollment terminal**`	Specifies a manual cut-and-paste method of certificate enrollment.
Step 4	`Router(ca-trustpoint)# **exit**`	Exits CA trustpoint configuration mode and returns to global configuration mode.
Step 5	`Router(config)# **crypto pki authenticate** *trustpoint_name*` `Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself`	Authenticates the named CA. The CA name should match the *trustpoint_name* specified in the **crypto pki trustpoint** command. At the prompt, paste the security certificate text.
Step 6	`**quit**` `% Do you accept this certificate? [yes/no]:`	Specifies the end of the security certificate text.
Step 7	`**yes**`	Confirms acceptance of the entered security certificate.
Step 8	`Router(config)# **end**`	Exits global configuration mode and returns to privileged EXEC mode.
Step 9	`Router# **copy running-config startup-config**`	Saves the configuration.

The example shows how to declare and authenticate the Cisco server security certificate and establish communication with the Cisco HTTPS server for Smart Call Home service:

```
Router# configure terminal
Router(config)# crypto pki trustpoint cisco
Router(ca-trustpoint)# enrollment terminal
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate cisco

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
```

(CA certificate text not shown)

```
quit
Certificate has the following attributes:
       Fingerprint MD5: A2339B4C 747873D4 6CE7C1F3 8DCB5CE9
      Fingerprint SHA1: 85371CA6 E550143D CE280347 1BDE3A09 E8F8770F

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported

Router(config)# end
Router# copy running-config startup-config
```

Start Smart Call Home Registration

To start the Smart Call Home registration process, perform this task:

Command or Action	Purpose
Router# **call-home send alert-group inventory profile CiscoTAC-1**	Manually sends an inventory alert group message to the CiscoTAC-1 destination profile.

After the Smart Call Home service is registered, you will receive an email from Cisco Systems. Follow the instructions in the email. The instructions include these procedures:

- To complete the device registration, launch the Smart Call Home web application at the following URL:

  http://tools.cisco.com/sch/

- Accept the Legal Agreement.
- Confirm device registration for Call Home devices with pending registration.

For more information about using the Smart Call Home web application, see the *Smart Call Home User Guide*. This user guide also includes configuration examples for sending Smart Call Home messages directly from your device or through a transport gateway (TG) aggregation point. You can use a TG aggregation point in cases requiring support for multiple devices or in cases where security requirements mandate that your devices must not be connected directly to the Internet.

Displaying Call Home Configuration Information

To display the configured Call Home information, perform these tasks:

Command	Purpose
Router# **show call-home**	Displays the Call Home configuration in summary.
Router# **show call-home detail**	Displays the Call Home configuration in detail.
Router# **show call-home alert-group**	Displays the available alert groups and their status.
Router# **show call-home mail-server status**	Checks and displays the availability of the configured email server(s).
Router# **show call-home profile** {**all** | *name*}	Displays the configuration of the specified destination profile. Use the keyword **all** to display the configuration of all destination profiles.
Router# **show call-home statistics**	Displays the statistics of Call Home events.

Examples 65-1 to 65-7 show the results when using different options of the **show call-home** command.

Example 65-1 Configured Call Home Information

```
Router# show call-home
Current call home settings:
    call home feature : disable
    call home message's from address: switch@example.com
    call home message's reply-to address: support@example.com

    contact person's email address: technical@example.com

    contact person's phone number: +1-408-555-1234
    street address: 1234 Picaboo Street, Any city, Any state, 12345
    customer ID: ExampleCorp
    contract ID: X123456789
    site ID: SantaClara
    Mail-server[1]: Address: smtp.example.com Priority: 1
    Mail-server[2]: Address: 192.168.0.1 Priority: 2
    Rate-limit: 20 message(s) per minute

Available alert groups:
    Keyword                  State   Description
    ------------------------ ------- -------------------------------
    configuration            Disable configuration info
    diagnostic               Disable diagnostic info
    environment              Disable environmental info
    inventory                Enable  inventory info
    syslog                   Disable syslog info

Profiles:
    Profile Name: campus-noc
    Profile Name: CiscoTAC-1

Router#
```

Example 65-2 Configured Call Home Information in Detail

```
Router# show call-home detail
Current call home settings:
    call home feature : disable
    call home message's from address: switch@example.com
    call home message's reply-to address: support@example.com

    contact person's email address: technical@example.com

    contact person's phone number: +1-408-555-1234
    street address: 1234 Picaboo Street, Any city, Any state, 12345
    customer ID: ExampleCorp
    contract ID: X123456789
    site ID: SantaClara
    Mail-server[1]: Address: smtp.example.com Priority: 1
    Mail-server[2]: Address: 192.168.0.1 Priority: 2
    Rate-limit: 20 message(s) per minute

Available alert groups:
    Keyword                  State   Description
    ------------------------ ------- -------------------------------
    configuration            Disable configuration info
    diagnostic               Disable diagnostic info
    environment              Disable environmental info
    inventory                Enable  inventory info
    syslog                   Disable syslog info

Profiles:

Profile Name: campus-noc
    Profile status: ACTIVE
    Preferred Message Format: long-text
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): noc@example.com
    HTTP  address(es): Not yet set up

    Alert-group               Severity
    ------------------------  ------------
    inventory                 normal

    Syslog-Pattern            Severity
    ------------------------  ------------
    N/A                       N/A

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): callhome@cisco.com
    HTTP  address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

    Periodic configuration info message is scheduled every 1 day of the month at 09:27

    Periodic inventory info message is scheduled every 1 day of the month at 09:
12

    Alert-group               Severity
    ------------------------  ------------
    diagnostic                minor
    environment               minor

    Syslog-Pattern            Severity
    ------------------------  ------------
    .*                        major

Router#
```

Example 65-3 Available Call Home Alert Groups

```
Router# show call-home alert-group
Available alert groups:
    Keyword                  State   Description
    ------------------------ ------- -------------------------------
    configuration            Disable configuration info
    diagnostic               Disable diagnostic info
    environment              Disable environmental info
    inventory                Enable  inventory info
    syslog                   Disable syslog info

Router#
```

Example 65-4 Email Server Status Information

```
Router# show call-home mail-server status
Please wait. Checking for mail server status ...

Translating "smtp.example.com"
    Mail-server[1]: Address: smtp.example.com Priority: 1 [Not Available]
    Mail-server[2]: Address: 192.168.0.1 Priority: 2 [Not Available]

Router#
```

Example 65-5 Information for All Destination Profiles (Predefined and User-Defined)

```
Router# show call-home profile all

Profile Name: campus-noc
    Profile status: ACTIVE
    Preferred Message Format: long-text
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): noc@example.com
    HTTP  address(es): Not yet set up

    Alert-group               Severity
    ------------------------  ------------
    inventory                 normal

    Syslog-Pattern            Severity
    ------------------------  ------------
    N/A                       N/A

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): callhome@cisco.com
    HTTP  address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

    Periodic configuration info message is scheduled every 1 day of the month at 09:27

    Periodic inventory info message is scheduled every 1 day of the month at 09:12

    Alert-group               Severity
    ------------------------  ------------
    diagnostic                minor
    environment               minor

    Syslog-Pattern            Severity
    ------------------------  ------------
    .*                        major

Router#
```

Example 65-6 Information for a User-Defined Destination Profile

```
Router# show call-home profile campus-noc

Profile Name: campus-noc
    Profile status: ACTIVE
    Preferred Message Format: long-text
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): noc@example.com
    HTTP  address(es): Not yet set up

    Alert-group               Severity
    ------------------------  ------------
    inventory                 normal

    Syslog-Pattern            Severity
    ------------------------  ------------
    N/A                       N/A

Router#
```

Example 65-7 Call Home Statistics

```
Router# show call-home statistics
Successful Call-Home Events: 1
Dropped Call-Home Events due to Rate Limiting: 0
Last call-home message sent time: 2007-04-25 11:07:04 GMT+00:00
```

Alert Group Trigger Events and Commands

Call Home trigger events are grouped into alert groups, with each alert group assigned CLI commands to execute when an event occurs. The CLI command output is included in the transmitted message. These tables list the trigger events included in each alert group, including the severity level of each event and the executed CLI commands for the alert group:

- Call Home Syslog Alert Group Events and Actions, Table 65-3 on page 65-27
- Call Home Environmental Alert Group Events and Actions, Table 65-4 on page 65-27
- Call Home Inventory Alert Group Events and Actions, Table 65-5 on page 65-30
- Call Home Diagnostic Failure Alert Group Events and Actions, Table 65-6 on page 65-31
- Call Home Test Alert Group Events and Actions, Table 65-7 on page 65-32
- Call Home License Alert Group Events and Actions, Table 65-8 on page 65-32
- Call Home Configuration Alert Group Events and Actions, Table 65-9 on page 65-33

Table 65-3 Call Home Syslog Alert Group Events and Actions

Alert Group Description:	Event logged to syslog		
Send to TAC:	No		
Executed Commands:	**show logging**, **show inventory**		
Call Home Trigger Event	**Syslog Event**	**Sev**	**Description**
SYSLOG	LOG_EMERG	0	system is unusable
	LOG_ALERT	1	action must be taken immediately
	LOG_CRIT	2	critical conditions
	LOG_ERR	3	error conditions
	LOG_WARNING	4	warning conditions
	LOG_NOTICE	5	normal but signification condition
	LOG_INFO	6	informational
	LOG_DEBUG	7	debug-level messages

Table 65-4 Call Home Environmental Alert Group Events and Actions

Alert Group Description:	Events related to power, fan and environment sensing elements such as temperature alarms		
Send to TAC:	Yes		
Executed Commands:	**show module**, **show environment**, **show logging**, **show inventory**, **show power**		
Call Home Trigger Event	**Syslog Event**	**Sev**	**Description**
FAN_FAILURE	FANPSINCOMPAT	4	Fan tray and power supply %d are incompatible
	ALARMCLR	4	The specified alarm condition has been cleared, and shutdown has been cancelled.
	FANHIOUTPUT	4	Version %d high-output fan-tray is in effect
	FANLOOUTPUT	4	Version %d low-output fan-tray is in effect
	FANVERCHK	4	Power-supply %d inserted is only compatible with Version %d fan-tray.
	FANTRAYFAILED	4	fan tray failed
	FANTRAYOK	4	fan tray OK
	FANCOUNTFAILED	4	Required number of fan trays is not present
	FANCOUNTOK	4	Required number of fan trays is present
	PSFANFAIL	4	the fan in power supply has failed
	PSFANOK	4	the fan in power supply is OK

Table 65-4 Call Home Environmental Alert Group Events and Actions (continued)

Alert Group Description:	Events related to power, fan and environment sensing elements such as temperature alarms		
Send to TAC:	Yes		
Executed Commands:	**show module**, **show environment**, **show logging**, **show inventory**, **show power**		
Call Home Trigger Event	**Syslog Event**	**Sev**	**Description**
TEMPERATURE_ALARM	MAJORTEMPALARM	2	It has exceeded allowed operating temperature range.
	MAJORTEMPALARMRECOVER	4	It has returned to allowed operating temperature range.
	MINORTEMPALARM	4	It has exceeded normal operating temperature range.
	MINORTEMPALARMRECOVER	4	It has returned to normal operating temperature range.
VTT_FAILED	VTTFAILED	4	VTT %d failed.
	VTTOK	4	VTT %d operational.
	VTTMAJFAILED	0	Too many VTT failures to continue system operation.
	VTTMAJRECOVERED	2	Enough VTTs operational to continue system operation.
CLOCK_FAILED	CLOCKFAILED	4	clock failed
	CLOCKOK	4	clock operational
	CLOCKMAJFAILED	0	too many clocks failed to continue system operation
	CLOCKMAJRECOVERED	2	enough clocks operational to continue system operation
	SHUTDOWN-SCHEDULED	2	shutdown for %s scheduled in %d seconds
	SHUTDOWN_NOT_SCHEDULED	2	Major sensor alarm for %s is ignored, %s will not be shutdown
	SHUTDOWN-CANCELLED	2	shutdown for cancelled
	SHUTDOWN	2	shutdown %s now because of %s
	SHUTDOWN-DISABLED	1	need to shutdown %s now but shutdown action is disabled!
	RESET_SCHEDULED	2	System reset scheduled in seconds
	CLOCK_SWITCHOVER	2	changing system switching clock
	CLOCK_A_MISSING	4	cannot detect clock A in the system
	CLOCK_B_MISSING	4	cannot detect clock B in the system
	USE_RED_CLOCK	4	system is using the redundant clock (clock B).
	ENABLED	4	power to module in slot %d set on
	DISABLED	4	power to module in slot %d set %s
	PSOK	4	power supply %d turned on.

Table 65-4 Call Home Environmental Alert Group Events and Actions (continued)

Alert Group Description:	Events related to power, fan and environment sensing elements such as temperature alarms		
Send to TAC:	Yes		
Executed Commands:	**show module**, **show environment**, **show logging**, **show inventory**, **show power**		
Call Home Trigger Event	**Syslog Event**	**Sev**	**Description**
POWER_SUPPLY_FAILURE	PSFAIL	4	power supply %d output failed.
	PSREDUNDANTMODE	4	power supplies set to redundant mode.
	PSCOMBINEDMODE	4	power supplies set to combined mode.
	PSREDUNDANTMISMATCH	4	power supplies rated outputs do not match.
	PSMISMATCH	4	power supplies rated outputs do not match.
	PSNOREDUNDANCY	4	Power supplies are not in full redundancy, power usage exceed lower capacity supply
	PSOCPSHUTDOWN	2	Power usage exceeds power supply %d allowable capacity.
	PSREDUNDANTONESUPPLY	4	in power-redundancy mode, system is operating on one power supply
	PSREDUNDANTBOTHSUPPLY	4	in power-redundancy mode, system is operating on both power supplies
	UNDERPOWERED	4	insufficient power to operate all FRUs in system.
	COULDNOTREPOWER	4	wanted to re-power FRU (slot %d) but could not.
	POWERDENIED	4	insufficient power, module in slot %d power denied.
	UNSUPPORTED	4	unsupported module in slot %d, power not allowed: %s.
	INSUFFICIENTPOWER	2	Powering down all linecards as there is not enough power to operate all critical cards
	INPUTCHANGE	4	Power supply %d input has changed. Power capacity adjusted to %sW
	PSINPUTDROP	4	Power supply %d input has droppe

Table 65-5 Call Home Inventory Alert Group Events and Actions

Alert Group Description:	Inventory status should be provided whenever a unit is cold-booted, or when FRUs are inserted or removed. This is considered a non-critical event, and the information is used for status and entitlement.		
Send to TAC:	Yes		
Executed Commands:	**show module**, **show version**, **show install running** (software modularity images only), **show inventory oid**, **show idprom all**, **remote command switch show version**, **show diagbus**, **show power**		
Call Home Trigger Event	**Syslog Event**	**Sev**	**Description**
HARDWARE_INSERTION	INSPS	6	Power supply inserted in slot %d
HARDWARE_REMOVAL	REMPS	6	Power supply removed from slot %d
	REMCARD	6	Card removed from slot %d, interfaces disabled
	STDBY_REMCARD	6	The OIR facility on Standby Supervior was notifed by the Active that a processor from slot[n] has been removed
HARDWARE_INSERTION	INSCAR	6	Card inserted in slot %d, interfaces are now online
	STDBY_INSCARD	6	Standby was notified, card online in slot %d
	SEQ_MISMATCH	6	SCP seq mismatch for card in slot %d : %s
HARDWARE_REMOVAL	UNKNOWN	3	Unknown card in slot %d, card is being disabled
	STDBY_UNKNOWN	3	Standby was notified, Unknown card in slot %d
	UNSUPPORTED	3	Card in slot %d is unsupported. %s
	PWRCYCLE	3	Card in module %d, is being power-cycled %s
	STDBY_PWRCYCLE	3	Standby was notified, Card in module %d is being power-cycled %s
	CONSOLE	6	Changing console ownership to %s processor
	RUNNING_CONFIG	6	During switchover, the OIR facility is unable to clean up running-config processor.
	DISALLOW	6	Supervisor attempting to come up as secondary in EHSA mode, will not be allowed
	REMFAN	6	Fan %d removed
HARDWARE_INSERTION	INSFAN	6	Fan %d inserted
	PSINSERTED	4	power supply inserted in slot %d.

Table 65-6 Call Home Diagnostic Failure Alert Group Events and Actions

Alert Group Description:	Events related to standard or intelligent line cards	
Send to TAC:	Yes	
Executed Commands:	**show module**, **show diagnostic result Module <#> detail**, **show version**, **show install running** (software modularity images only), **show inventory**, **show buffers**, **show logging**, **show diagnostic result module all**, **remote command switch show version**, **show logging system last 100**	
Call Home Trigger Event:	DIAGNOSTICS_FAILURE	
Syslog Event	**Sev**	**Description**
C2PLUSWITHNODB	2	The constellation 2 plus module in slot %d has no forwarding daughter board. Power denied
DFCMISMATCH	2	Module %d DFC incompatible with Supervisor DFC. Power denied
BADFLOWCTRL	2	Module %d not at an appropriate hardware revision level to support DFC. Power denied
BADFLOWCTRL_WARN	2	WARNING: Module %d not at an appropriate hardware revision level to support DFC3
BADPINN1	2	Module %d not at an appropriate hardware revision level to coexist with PFC3 system. Power denied
FANUPGREQ	2	Module %d not supported without fan upgrade
INSUFFCOO	4	Module %d cannot be adequately cooled
PROVISION	6	Module %d does not meet the provisioning requirements, power denied
PWRFAILURE	6	Module %d is being disabled due to power convertor failure
LC_FAILURE	3	Module %d has Major online diagnostic failure, %s
HARD_RESET	3	Module %d is being hard reset as a part of swichover error recovery
SOFT_RESET	3	Module %d is being soft reset as a part of swichover error recovery
DOWNGRADE	6	Fabric capable module %d not at an appropriate hardware revision level, and can only run in flowthrough mode
DIAG_OK		
DIAG_BYPASS		
DIAG_ERROR		
DIAG_MINOR_ERROR		
DIAG_MAJOR_ERROR		
DIAG_LINE_CARD_NOT_PRESENT		
DIAG_LINE_CARD_REMOVED		
DIAG_INVALID_TEST_ID_RANGE		
DIAG_INVALID_PORT_RANGE		
DIAG_IS_BUSY		
DIAG_IS_IDLE		
DIAG_NO_SCHEDULE		
DIAG_SCHEDULE_EXIST		

Table 65-6 Call Home Diagnostic Failure Alert Group Events and Actions (continued)

Alert Group Description:	Events related to standard or intelligent line cards	
Send to TAC:	Yes	
Executed Commands:	**show module**, **show diagnostic result Module <#> detail**, **show version**, **show install running** (software modularity images only), **show inventory**, **show buffers**, **show logging**, **show diagnostic result module all**, **remote command switch show version**, **show logging system last 100**	
Call Home Trigger Event:	DIAGNOSTICS_FAILURE	
Syslog Event	**Sev**	**Description**
DIAG_NO_TEST		
DIAG_UNKNOWN		
DIAG_NOT_AVAILABLE		
DIAG_EXIT_ON_ERROR		
DIAG_EXIT_ON_FAIL_LIMIT_REACHED		
DIAG_INVALID_SCHEDULE		
DIAG_PF_DIAG_NOT_SUPORTED		
DIAG_IS_STOPPED		
DIAG_INVALID_DEVICE_RANGE		

Table 65-7 Call Home Test Alert Group Events and Actions

Alert Group Description:	—	
Send to TAC:	Yes	
Executed Commands:	**show version**, **show module**, **show inventory**, **test message**	
Call Home Trigger Event:	—	
Syslog Event	**Sev**	**Description**
TEST	2	User-generated test message.

Table 65-8 Call Home License Alert Group Events and Actions

Alert Group Description:	For future use.	
Send to TAC:	Yes	
Executed Commands:	**show license all**, **show running-config**	
Call Home Trigger Event:	—	
Syslog Event	**Sev**	**Description**
—	—	Events related to unlicensed use of licensed features, or inconsistent license information.

Table 65-9 Call Home Configuration Alert Group Events and Actions

Alert Group Description:	User generated request for configuration.	
Send to TAC:	Yes	
Executed Commands:	**show module**, **show version**, **show install running** (software modularity images only), **show running-config all**, **show startup-config**, **show inventory**, **remote command switch show version**	
Call Home Trigger Event:	—	
Syslog Event	**Sev**	**Description**
—	—	—

Message Contents

The following tables display the content formats of alert group messages:

- Table 65-10 describes the content fields of a short text message.
- Table 65-11 describes the content fields that are common to all long text and XML messages. The fields specific to a particular alert group message are inserted after the common fields.
- Table 65-12 describes the content fields for reactive messages (system failures that require a TAC case) and proactive messages (issues that might result in degraded system performance).
- Table 65-13 describes the content fields for an inventory message.

Table 65-10 Format for a Short Text Message

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to a system message

Table 65-11 Common Fields for All Long Text and XML Messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Time stamp	Date and time stamp of event in ISO time notation: *YYYY-MM-DD*T*HH:MM:SS*	CallHome/EventTime
Message name	Name of message. Specific event names are listed in the "Alert Group Trigger Events and Commands" section on page 65-26.	(for short text message only)
Message type	Specifically Call Home.	CallHome/Event/Type
Message subtype	Specific type of message: full, delta, or test.	CallHome/Event/SubType

Table 65-11 Common Fields for All Long Text and XML Messages (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Message group	Specifically reactive or proactive.	(for long text message only)
Severity level	Severity level of message (see Table 65-2 on page 65-16).	Body/Block/Severity
Source ID	Product type for routing. Specifically Catalyst 6500.	(for long text message only)
Device ID	Unique device identifier (UDI) for end device generating message. This field should be empty if the message is nonspecific to a fabric switch. The format is *type@Sid@serial*. • *type* is the product model number from backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number· • *serial* is the number identified by the Sid field. Example: WS-C6509@C@12345678	CallHome/CustomerData/ContractData/DeviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	CallHome/CustomerData/ContractData/CustomerId
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	CallHome/CustomerData/ContractData/ContractId
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	CallHome/CustomerData/ContractData/SiteId
Server ID	If the message is generated from the fabric switch, this is the unique device identifier (UDI) of the switch. The format is *type@Sid@serial*. • *type* is the product model number from backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number· • *serial* is the number identified by the Sid field. Example: WS-C6509@C@12345678	(for long text message only)
Message description	Short text describing the error.	CallHome/MessageDescription

Table 65-11 Common Fields for All Long Text and XML Messages (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Device name	Node that experienced the event. This is the host name of the device.	CallHome/CustomerData/SystemInfo/Name
Contact name	Name of person to contact for issues associated with the node experiencing the event.	CallHome/CustomerData/SystemInfo/Contact
Contact email	Email address of person identified as contact for this unit.	CallHome/CustomerData/SystemInfo/ContactEmail
Contact phone number	Phone number of the person identified as the contact for this unit.	CallHome/CustomerData/SystemInfo/ContactPhoneNumber
Street address	Optional field containing street address for RMA part shipments associated with this unit.	CallHome/CustomerData/SystemInfo/StreetAddress
Model name	Model name of the switch. This is the specific model as part of a product family name.	CallHome/Device/Cisco_Chassis/Model
Serial number	Chassis serial number of the unit.	CallHome/Device/Cisco_Chassis/SerialNumber
Chassis part number	Top assembly number of the chassis.	CallHome/Device/Cisco_Chassis/AdditionalInformation/ AD@name="PartNumber"/
System Object ID	The System ObjectID that uniquely identifies the system.	CallHome/Device/Cisco_Chassis/AdditionalInformation/ AD@name="sysObjectID"
SysDesc	System description for the managed element.	CallHome/Device/Cisco_Chassis/AdditionalInformation/ AD@name="sysDescr"
The following fields may be repeated if multiple CLI commands are executed for this alert group.		
Command output name	The exact name of the issued CLI command.	/aml/Attachments/Attachment/Name
Attachment type	Type (usually inline).	/aml/Attachments/Attachment@type
MIME type	Normally text/plain or encoding type.	/aml/attachments/attachment/Data@encoding
Command output text	Output of command automatically executed (see the "Alert Group Trigger Events and Commands" section on page 65-26).	/aml/attachments/attachment/atdata

Table 65-12 Fields for a Reactive or Proactive Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	CallHome/Device/Cisco_Chassis/HardwareVersion
Supervisor module software version	Top-level software version.	CallHome/Device/Cisco_Chassis/AdditionalInformation/ AD@name="SoftwareVersion"

Table 65-12 Fields for a Reactive or Proactive Event Message (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Affected FRU name	Name of the affected FRU generating the event message.	CallHome/Device/Cisco_Chassis/Cisco_Card/Model
Affected FRU serial number	Serial number of affected FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/SerialNumber
Affected FRU part number	Part number of affected FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/PartNumber
FRU slot	Slot number of FRU generating the event message.	CallHome/Device/Cisco_Chassis/Cisco_Card/ LocationWithinContainer
FRU hardware version	Hardware version of affected FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/HardwareVersion
FRU software version	Software version(s) running on affected FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/SoftwareIdentity/VersionString
Process name	Name of process.	/aml/body/process/name
Process ID	Unique process ID.	/aml/body/process/id
Process state	State of process (for example, running or halted).	/aml/body/process/processState
Process exception	Exception or reason code.	/aml/body/process/exception

Table 65-13 Fields for an Inventory Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	CallHome/Device/Cisco_Chassis/HardwareVersion
Supervisor module software version	Top-level software version.	CallHome/Device/Cisco_Chassis/AdditionalInformation/AD@name="SoftwareVersion"
FRU name	Name of the affected FRU generating the event message.	CallHome/Device/Cisco_Chassis/Cisco_Card/Model
FRU s/n	Serial number of FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/SerialNumber
FRU part number	Part number of FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/PartNumber
FRU slot	Slot number of FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/LocationWithinContainer

Table 65-13 Fields for an Inventory Event Message (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
FRU hardware version	Hardware version of FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/HardwareVersion
FRU software version	Software version(s) running on FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/SoftwareIdentity/VersionString

Sample Syslog Alert Notification in Long-Text Format

```
source:MDS9000
Switch Priority:7
Device Id:WS-C6509@C@FG@07120011
Customer Id:Example.com
Contract Id:123
Site Id:San Jose
Server Id:WS-C6509@C@FG@07120011
Time of Event:2004-10-08T11:10:44
Message Name:SYSLOG_ALERT
Message Type:Syslog
Severity Level:2
System Name:10.76.100.177
Contact Name:User Name
Contact Email:admin@yourcompany.com
Contact Phone:+1 408 555-1212
Street Address:#1234 Picaboo Street, Any city, Any state, 12345
Event Description:2006 Oct  8 11:10:44 10.76.100.177 %PORT-5-IF_TRUNK_UP: %$VSAN 1%$
Interface fc2/5, vsan 1 is up

syslog_facility:PORT
start chassis information:
Affected Chassis:WS-C6509
Affected Chassis Serial Number:FG@07120011
Affected Chassis Hardware Version:0.104
Affected Chassis Software Version:3.1(1)
Affected Chassis Part No:73-8607-01
end chassis information:
```

Sample Syslog Alert Notification in XML Format

```
From: example
Sent: Wednesday, April 25, 2007 7:20 AM
To: User (user)
Subject: System Notification From Router - syslog - 2007-04-25 14:19:55
GMT+00:00

<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session xmlns:aml-session="http://www.example.com/2004/01/aml-session"
soap-env:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.example.com/services/DDCEService</aml-session:To>
<aml-session:Path>
<aml-session:Via>http://www.example.com/appliance/uri</aml-session:Via>
```

```
</aml-session:Path>
<aml-session:From>http://www.example.com/appliance/uri</aml-session:From>
<aml-session:MessageId>M2:69000101:C9D9E20B</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block xmlns:aml-block="http://www.example.com/2004/01/aml-block">
<aml-block:Header>
<aml-block:Type>http://www.example.com/2005/05/callhome/syslog</aml-block:Type>
<aml-block:CreationDate>2007-04-25 14:19:55 GMT+00:00</aml-block:CreationDate>
<aml-block:Builder>
<aml-block:Name>Cat6500</aml-block:Name>
<aml-block:Version>2.0</aml-block:Version>
</aml-block:Builder>
<aml-block:BlockGroup>
<aml-block:GroupId>G3:69000101:C9F9E20C</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>2</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
<ch:CallHome xmlns:ch="http://www.example.com/2005/05/callhome" version="1.0">
<ch:EventTime>2007-04-25 14:19:55 GMT+00:00</ch:EventTime>
<ch:MessageDescription>03:29:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by
console</ch:MessageDescription>
<ch:Event>
<ch:Type>syslog</ch:Type>
<ch:SubType></ch:SubType>
<ch:Brand>Cisco Systems</ch:Brand>
<ch:Series>Catalyst 6500 Series Switches</ch:Series>
</ch:Event>
<ch:CustomerData>
<ch:UserData>
<ch:Email>user@example.com</ch:Email>
</ch:UserData>
<ch:ContractData>
<ch:CustomerId>12345</ch:CustomerId>
<ch:SiteId>building 1</ch:SiteId>
<ch:ContractId>abcdefg12345</ch:ContractId>
<ch:DeviceId>WS-C6509@C@69000101</ch:DeviceId>
</ch:ContractData>
<ch:SystemInfo>
<ch:Name>Router</ch:Name>
<ch:Contact></ch:Contact>
<ch:ContactEmail>user@example.com</ch:ContactEmail>
<ch:ContactPhoneNumber>+1 408 555-1212</ch:ContactPhoneNumber>
<ch:StreetAddress>270 E. Tasman Drive, San Jose, CA</ch:StreetAddress>
</ch:SystemInfo>
</ch:CustomerData>
<ch:Device>
<rme:Chassis xmlns:rme="http://www.example.com/rme/4.0">
<rme:Model>WS-C6509</rme:Model>
<rme:HardwareVersion>1.0</rme:HardwareVersion>
<rme:SerialNumber>69000101</rme:SerialNumber>
<rme:AdditionalInformation>
<rme:AD name="PartNumber" value="73-3438-03 01" />
<rme:AD name="SoftwareVersion" value="12.2(20070421:012711)" />
</rme:AdditionalInformation>
</rme:Chassis>
</ch:Device>
</ch:CallHome>
```

```
</aml-block:Content>
<aml-block:Attachments>
<aml-block:Attachment type="inline">
<aml-block:Name>show logging</aml-block:Name>
<aml-block:Data encoding="plain">
<![CDATA[
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 53 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 53 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 72 message lines logged

Log Buffer (8192 bytes):

00:00:54: curr is 0x20000

00:00:54: RP: Currently running ROMMON from F2 region
00:01:05: %SYS-5-CONFIG_I: Configured from memory by console
00:01:09: %SYS-5-RESTART: System restarted --
Cisco IOS Software, s72033_rp Software (s72033_rp-ADVENTERPRISEK9_DBG-VM), Experimental
Version 12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 15:54 by xxx

Firmware compiled 11-Apr-07 03:34 by integ Build [100]


00:01:01: %PFREDUN-6-ACTIVE: Initializing as ACTIVE processor for this switch

00:01:01: %SYS-3-LOGGER_FLUSHED: System was paused for 00:00:00 to ensure console
debugging output.

00:03:00: SP: SP: Currently running ROMMON from F1 region
00:03:07: %C6K_PLATFORM-SP-4-CONFREG_BREAK_ENABLED: The default factory setting for config
register is 0x2102.It is advisable to retain 1 in 0x2102 as it prevents returning to
ROMMON when break is issued.

00:03:18: %SYS-SP-5-RESTART: System restarted --
Cisco IOS Software, s72033_sp Software (s72033_sp-ADVENTERPRISEK9_DBG-VM), Experimental
Version 12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 18:00 by xxx
00:03:18: %SYS-SP-6-BOOTTIME: Time taken to reboot after reload =  339 seconds
00:03:18: %OIR-SP-6-INSPS: Power supply inserted in slot 1
00:03:18: %C6KPWR-SP-4-PSOK: power supply 1 turned on.
00:03:18: %OIR-SP-6-INSPS: Power supply inserted in slot 2
00:01:09: %SSH-5-ENABLED: SSH 1.99 has been enabled
00:03:18: %C6KPWR-SP-4-PSOK: power supply 2 turned on.
00:03:18: %C6KPWR-SP-4-PSREDUNDANTMISMATCH: power supplies rated outputs do not match.
00:03:18: %C6KPWR-SP-4-PSREDUNDANTBOTHSUPPLY: in power-redundancy mode, system is
operating on both power supplies.
00:01:10: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
00:01:10: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
00:03:20: %C6KENV-SP-4-FANHIOUTPUT: Version 2 high-output fan-tray is in effect
00:03:22: %C6KPWR-SP-4-PSNOREDUNDANCY: Power supplies are not in full redundancy, power
usage exceeds lower capacity supply
00:03:26: %FABRIC-SP-5-FABRIC_MODULE_ACTIVE: The Switch Fabric Module in slot 6 became
active.
```

```
00:03:28: %DIAG-SP-6-RUN_MINIMUM: Module 6: Running Minimal Diagnostics...
00:03:50: %DIAG-SP-6-DIAG_OK: Module 6: Passed Online Diagnostics
00:03:50: %OIR-SP-6-INSCARD: Card inserted in slot 6, interfaces are now online
00:03:51: %DIAG-SP-6-RUN_MINIMUM: Module 3: Running Minimal Diagnostics...
00:03:51: %DIAG-SP-6-RUN_MINIMUM: Module 7: Running Minimal Diagnostics...
00:03:51: %DIAG-SP-6-RUN_MINIMUM: Module 9: Running Minimal Diagnostics...
00:01:51: %MFIB_CONST_RP-6-REPLICATION_MODE_CHANGE: Replication Mode Change Detected.
Current system replication mode is Ingress
00:04:01: %DIAG-SP-6-DIAG_OK: Module 3: Passed Online Diagnostics
00:04:01: %OIR-SP-6-DOWNGRADE: Fabric capable module 3 not at an appropriate hardware
revision level, and can only run in flowthrough mode
00:04:02: %OIR-SP-6-INSCARD: Card inserted in slot 3, interfaces are now online
00:04:11: %DIAG-SP-6-DIAG_OK: Module 7: Passed Online Diagnostics
00:04:14: %OIR-SP-6-INSCARD: Card inserted in slot 7, interfaces are now online
00:04:35: %DIAG-SP-6-DIAG_OK: Module 9: Passed Online Diagnostics
00:04:37: %OIR-SP-6-INSCARD: Card inserted in slot 9, interfaces are now online
00:00:09: DaughterBoard (Distributed Forwarding Card 3)

Firmware compiled 11-Apr-07 03:34 by integ Build [100]


00:00:22: %SYS-DFC4-5-RESTART: System restarted --
Cisco IOS Software, c6lc2 Software (c6lc2-SPDBG-VM), Experimental Version
12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 17:20 by xxx
00:00:23: DFC4: Currently running ROMMON from F2 region
00:00:25: %SYS-DFC2-5-RESTART: System restarted --
Cisco IOS Software, c6slc Software (c6slc-SPDBG-VM), Experimental Version
12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 16:40 by username1
00:00:26: DFC2: Currently running ROMMON from F2 region
00:04:56: %DIAG-SP-6-RUN_MINIMUM: Module 4: Running Minimal Diagnostics...
00:00:09: DaughterBoard (Distributed Forwarding Card 3)

Firmware compiled 11-Apr-07 03:34 by integ Build [100]

slot_id is 8

00:00:31: %FLASHFS_HES-DFC8-3-BADCARD: /bootflash:: The flash card seems to be corrupted
00:00:31: %SYS-DFC8-5-RESTART: System restarted --
Cisco IOS Software, c6lc2 Software (c6lc2-SPDBG-VM), Experimental Version
12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 17:20 by username1
00:00:31: DFC8: Currently running ROMMON from S (Gold) region
00:04:59: %DIAG-SP-6-RUN_MINIMUM: Module 2: Running Minimal Diagnostics...
00:05:12: %DIAG-SP-6-RUN_MINIMUM: Module 8: Running Minimal Diagnostics...
00:05:13: %DIAG-SP-6-RUN_MINIMUM: Module 1: Running Minimal Diagnostics...
00:00:24: %SYS-DFC1-5-RESTART: System restarted --
Cisco IOS Software, c6slc Software (c6slc-SPDBG-VM), Experimental Version
12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 16:40 by username1
00:00:25: DFC1: Currently running ROMMON from F2 region
00:05:30: %DIAG-SP-6-DIAG_OK: Module 4: Passed Online Diagnostics
00:05:31: %SPAN-SP-6-SPAN_EGRESS_REPLICATION_MODE_CHANGE: Span Egress HW Replication Mode
Change Detected. Current replication mode for unused asic session 0 is Centralized
00:05:31: %SPAN-SP-6-SPAN_EGRESS_REPLICATION_MODE_CHANGE: Span Egress HW Replication Mode
Change Detected. Current replication mode for unused asic session 1 is Centralized
00:05:31: %OIR-SP-6-INSCARD: Card inserted in slot 4, interfaces are now online
00:06:02: %DIAG-SP-6-DIAG_OK: Module 1: Passed Online Diagnostics
00:06:03: %OIR-SP-6-INSCARD: Card inserted in slot 1, interfaces are now online
```

```
00:06:31: %DIAG-SP-6-DIAG_OK: Module 2: Passed Online Diagnostics
00:06:33: %OIR-SP-6-INSCARD: Card inserted in slot 2, interfaces are now online
00:04:30: %XDR-6-XDRIPCNOTIFY: Message not sent to slot 4/0 (4) because of IPC error
timeout. Disabling linecard. (Expected during linecard OIR)
00:06:59: %DIAG-SP-6-DIAG_OK: Module 8: Passed Online Diagnostics
00:06:59: %OIR-SP-6-DOWNGRADE_EARL: Module 8 DFC installed is not identical to system PFC
and will perform at current system operating mode.
00:07:06: %OIR-SP-6-INSCARD: Card inserted in slot 8, interfaces are now online

Router#]]></aml-block:Data>
</aml-block:Attachment>
</aml-block:Attachments>
</aml-block:Block>
</soap-env:Body>
</soap-env:Envelope>
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Using the System Event Archive

This chapter describes how to use the System Event Archive (SEA). Release 12.2(33)SXH and later releases support the SEA.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- SEA is supported with Supervisor Engine 32, Supervisor Engine 720-10GE, and Supervisor Engine 720 with a CompactFlash adapter and a CompactFlash card (WS-CF-UPG=).

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding the System Event Archive

The primary method of discovering the cause of system failure is system messages. When system messages do not provide the information needed to determine the cause of a failure, you can enable debug traces and attempt to recreate the failure. However, there are several situations in which neither of the above methods provides an optimum solution:

- Reviewing a large number of system messages can be an inefficient method of determing the cause of a failure.
- Debug trace is usually not configured by default.
- You cannot recreate the failure while using debug trace.

- Using debug trace is not an option if the switch on which the failure has occurred is part of your critical network.

The SEA enables each of the CPUs on a switch to report events to the management processor using an out-of-band interface. Each event is logged in nonvolatile memory with a time stamp. You can retrieve the event log by accessing the bootflash on the device, or you can copy the log to another location such as a removable storage device.

The SEA maintains two files in the bootdisk, using up to 32 MB. These files contain the most recent messages recorded to the log:

- sea_log.dat—Applications store the most recent system events in this file.
- sea_console.dat—The most recent console messages are stored in this file.

These files are for system use and should not be removed.

Displaying the SEA Logging System

To display the SEA logging system, perform this task:

Command	Purpose
`Router# show logging system [disk	size]`	Displays the contents of the SEA. (Optional) Use the keyword **disk** to display the location where the SEA is stored. Use the keyword **size** to display the current size of the SEA.
`Router# clear logging system`	Removes the event records stored in the SEA.

The following example shows how to display the SEA:

```
Router# show logging system
SEQ: MM/DD/YY HH:MM:SS MOD/SUB: SEV, COMP,    MESSAGE
=====================================================

1: 01/24/07 15:38:40   6/-1 : MAJ, GOLD, syndiagSyncPinnacle failed in slot 6

2: 01/24/07 15:38:40   6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in sw_mode 1

3: 01/24/07 15:38:40   6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in sw_mode 1

4: 01/24/07 15:38:40   6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in sw_mode 1

5: 01/24/07 15:38:40   6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in sw_mode 1

6: 01/24/07 15:38:40   6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in sw_mode 1

7: 01/24/07 15:38:39   6/-1 : MAJ, GOLD, queryHyperionSynched[6]: Hyperion out of sync in sw_mode 1
```

The following example shows how to display the SEA logging system disk:

```
Router# show logging system disk
SEA log disk: sup-bootdisk:
```

The following example shows how to display the current size of the SEA:

```
Router# show logging system size
SEA log size: 33554432 bytes
```

The following example shows how to clear the SEA:

```
Router# clear logging system
Clear logging system operation will take a while.
Do you want to continue? [no]: yes
Router#
```

Copying the SEA To Another Device

To copy the SEA to another device, such as a removeable memory device, perform this task:

Command	Purpose
`Router# copy logging system file_system`	Copies the contents of the SEA to the specified destination file system or process.

The valid values for *file_system* are:

- **bootflash:**
- **disk0:**
- **disk1:**
- **ftp:**
- **http:**
- **https:**
- **rcp:**
- **slavebootflash:**
- **slavedisk0:**
- **slavedisk1:**
- **slavesup-bootdisk:**
- **slavesup-bootflash:**
- **sup-bootdisk:**
- **sup-bootflash:**
- **tftp:**

The following example shows how to copy the SEA to the disk0 file system:

```
Router# copy logging system disk0:
Destination filename [sea_log.dat]?
Copy in
progress...CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
33554432 bytes copied in 112.540 secs (298156 bytes/sec)
```

The following example shows how to copy the SEA using the remote file copy function (rcp):

```
Router# copy logging system rcp:
Address or name of remote host []? 192.0.2.1
Destination username [Router]? username1
Destination filename [sea_log.dat]? /auto/tftpboot-users/username1/sea_log.dat
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
33554432 bytes copied in 48.172 secs (696555 bytes/sec)
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Backplane Traffic Monitoring

This chapter describes how to configure the Backplane Traffic Monitoring feature in Cisco IOS Software Release 12.2SX. Cisco IOS Release 12.2(33)SXI and later releases support the Backplane Traffic Monitoring feature.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Traffic Monitoring

Backplane Traffic Monitoring can monitor the backplane and fabric-channel traffic utilization at a configured interval and threshold.

Traffic monitoring allows the switch to monitor the backplane and fabric channel traffic utilization at a configured interval and threshold. Syslog messages are generated when the traffic utilization is above or below the configured threshold. The following examples show several types of syslog messages:

- 00:08:03: %TRAFFIC_UTIL-SP-4-MONITOR_BACKPLANE_REACH_THR: Backplane traffic utilization is 26%, reached threshold(20%) within 10 second interval

- 00:08:13: %TRAFFIC_UTIL-SP-4-MONITOR_BACKPLANE_BELOW_THR: Backplane traffic utilization is 18%, below threshold(20%) within 10 second interval
- 00:08:44: %TRAFFIC_UTIL-SP-4-MONITOR_FABRIC_IG_REACH_THR: Module 1, Channel 0 ingress traffic utilization is 5%, reached threshold(3%) within 30 second interval
- 00:08:44: %TRAFFIC_UTIL-SP-4-MONITOR_FABRIC_EG_REACH_THR: Module 1, Channel 0 egress traffic utilization is 5%, reached threshold(3%) within 30 second interval
- 00:09:14: %TRAFFIC_UTIL-SP-4-MONITOR_FABRIC_IG_BELOW_THR: Module 1, Channel 0 ingress traffic utilization is 1%, below threshold(3%) within 30 second interval
- 00:09:14: %TRAFFIC_UTIL-SP-4-MONITOR_FABRIC_EG_BELOW_THR: Module 1, Channel 0 egress traffic utilization is 1%, below threshold(3%) within 30 second interval

Default Configuration

- The default threshold is 80 percent.
- Traffic monitor is off by default.

Configuration Guidelines and Restrictions

The syslog message buffer is limited in size. To reduce false alarms and the number of syslog messages, use the following guidelines:

- Traffic can occur in bursts.

  If a small amount of bursts occur in a monitoring interval, it does not represent a capacity overload issue for the system; the hardware buffers are able to absorb the effects and not cause packet drops. For an example, if you set a monitoring interval to 10 seconds and the threshold to 80 percent, there are a total of 10 traffic utilization readings. Assume only 2 of the readings reached 90 percent and the other 8 readings are 20 percent. If the peak threshold of 90 percent is used to compare with the threshold, an unwanted warning syslog message is generated. It is better to use the average 34 percent of the 10 readings to compare with the threshold and not generate warning messages in this case. If the peak value comparison is really needed, you can set the interval to 1 second. Setting the interval to 1 second compares the reading directly with the threshold.
- The number of syslog messages that generate syslog messages are from below the threshold and above the threshold.

Configuring Backplane Traffic Monitoring

To configure the Backplane Traffic Monitoring feature, perform one or more of the following tasks:

Command	Purpose
`Router(config)#` **`monitor traffic-util backplane interval`** *`interval`* **`threshold`** *`percentage`*	Configures the backplane utilization traffic monitoring.
`Router(config)#` **`monitor traffic-util fabric module`** {*`mod-num`* | **`all`**} {**`channel`** {**`0`**|**`1`**|**`both`**}} {**`direction`** {**`egress`** | **`ingress`** | **`both`**}} [**`interval`** *`interval`* **`threshold`** *`percentage`*]	Configures the fabric channel utilization traffic monitoring.

Command	Purpose
`Router(config)# monitor traffic-util fabric logging interval second`	Configures the fabric channel utilization traffic monitor SYSLOG interval when the traffic utilization is in the crossed state.
`Router(config)# monitor traffic-util backplane logging interval second`	Configures the traffic monitor backplane SYSLOG interval when the traffic utilization is in the crossed state.
`Router# show catalyst6000 traffic-meter`	Displays the percentage of the backplane (shared bus) utilization and traffic monitor status information.

When configuring a range of interfaces, you can enter the *mod-num* as a list or a range. Separate each entry with a comma and each range with a hyphen (-). For example, 1,3,5-9,12.

The following example shows how to enable backplane traffic utilization monitoring:

```
Router(config)# monitor traffic-util backplane logging interval 50 threshold 100
```

The following example shows how to disable backplane traffic utilization monitoring:

```
Router(config)# no monitor traffic-util backplane
```

The following example shows how to specify the fabric channel traffic utilization monitor interval and threshold for a fabric channel on a specific module:

```
Router(config)# monitor traffic-util fabric module 8 channel both direction both interval
50 threshold 60
```

The following example shows how to specify the fabric channel traffic utilization monitor threshold for a specific fabric channel and for egress traffic only:

```
Router(config)# monitor traffic-util fabric module 6 channel 0 direction egress interval
100 threshold 90
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring Local SPAN, RSPAN, and ERSPAN

This chapter describes how to configure local Switched Port Analyzer (SPAN), remote SPAN (RSPAN), and Encapsulated RSPAN (ERSPAN) in Cisco IOS Release 12.2SX.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- SPA ports and FlexWAN ports do not support SPAN, RSPAN, or ERSPAN.

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Local SPAN, RSPAN, and ERSPAN

These sections describe how local SPAN, RSPAN, and ERSPAN work:

Local SPAN, RSPAN, and ERSPAN Overview

SPAN copies traffic from one or more CPUs, one or more ports, one or more EtherChannels, or one or more VLANs, and sends the copied traffic to one or more destinations for analysis by a network analyzer such as a SwitchProbe device or other Remote Monitoring (RMON) probe. Traffic can also be sent to the processor for packet capture by the Mini Protocol Analyzer, as described in Chapter 72, "Using the Mini Protocol Analyzer."

SPAN does not affect the switching of traffic on sources. You must dedicate the destination for SPAN use. The SPAN-generated copies of traffic compete with user traffic for switch resources.

These sections provide an overview of local SPAN, RSPAN, and ERSPAN:

- Local SPAN Overview, page 68-2
- RSPAN Overview, page 68-3
- ERSPAN Overview, page 68-4
- Understanding the Traffic Monitored at SPAN Sources, page 68-4

Local SPAN Overview

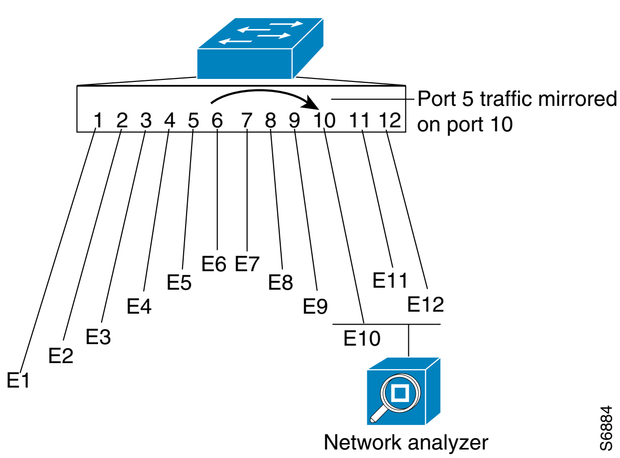
A local SPAN session is an association of source ports and source VLANs with one or more destinations. You configure a local SPAN session on a single switch. Local SPAN does not have separate source and destination sessions.

Local SPAN sessions do not copy locally sourced RSPAN VLAN traffic from source trunk ports that carry RSPAN VLANs. Local SPAN sessions do not copy locally sourced RSPAN GRE-encapsulated traffic from source ports.

Each local SPAN session can have either ports or VLANs as sources, but not both.

Local SPAN copies traffic from one or more source ports in any VLAN or from one or more VLANs to a destination for analysis (see Figure 68-1). For example, as shown in Figure 68-1, all traffic on Ethernet port 5 (the source port) is copied to Ethernet port 10. A network analyzer on Ethernet port 10 receives all traffic from Ethernet port 5 without being physically attached to Ethernet port 5.

Figure 68-1 Example SPAN Configuration



RSPAN Overview

RSPAN supports source ports, source VLANs, and destinations on different switches, which provides remote monitoring of multiple switches across your network (see Figure 68-2). RSPAN uses a Layer 2 VLAN to carry SPAN traffic between switches.

RSPAN consists of an RSPAN source session, an RSPAN VLAN, and an RSPAN destination session. You separately configure RSPAN source sessions and destination sessions on different switches. To configure an RSPAN source session on one switch, you associate a set of source ports or VLANs with an RSPAN VLAN. To configure an RSPAN destination session on another switch, you associate the destinations with the RSPAN VLAN.

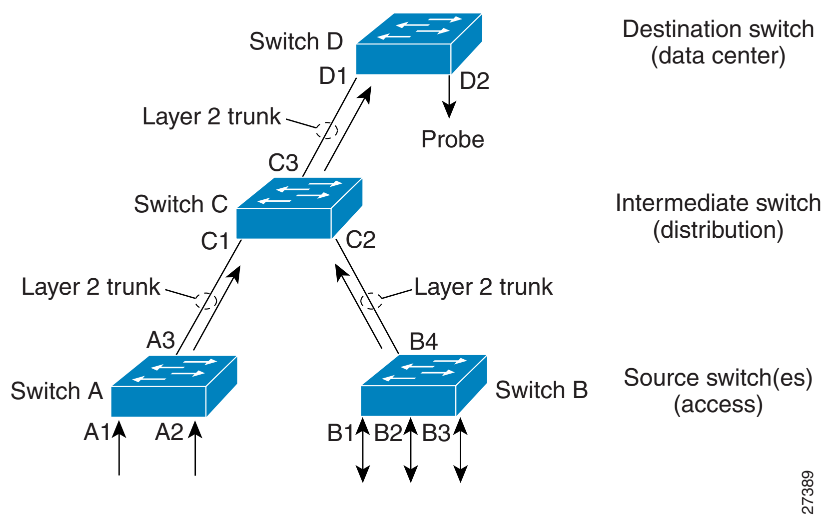
The traffic for each RSPAN session is carried as Layer 2 nonroutable traffic over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches. All participating switches must be trunk-connected at Layer 2.

RSPAN source sessions do not copy locally sourced RSPAN VLAN traffic from source trunk ports that carry RSPAN VLANs. RSPAN source sessions do not copy locally sourced RSPAN GRE-encapsulated traffic from source ports.

Each RSPAN source session can have either ports or VLANs as sources, but not both.

The RSPAN source session copies traffic from the source ports or source VLANs and switches the traffic over the RSPAN VLAN to the RSPAN destination session. The RSPAN destination session switches the traffic to the destinations.

Figure 68-2 RSPAN Configuration



ERSPAN Overview

ERSPAN supports source ports, source VLANs, and destinations on different switches, which provides remote monitoring of multiple switches across your network (see Figure 68-3). ERSPAN uses a GRE tunnel to carry traffic between switches.

ERSPAN consists of an ERSPAN source session, routable ERSPAN GRE-encapsulated traffic, and an ERSPAN destination session. You separately configure ERSPAN source sessions and destination sessions on different switches.

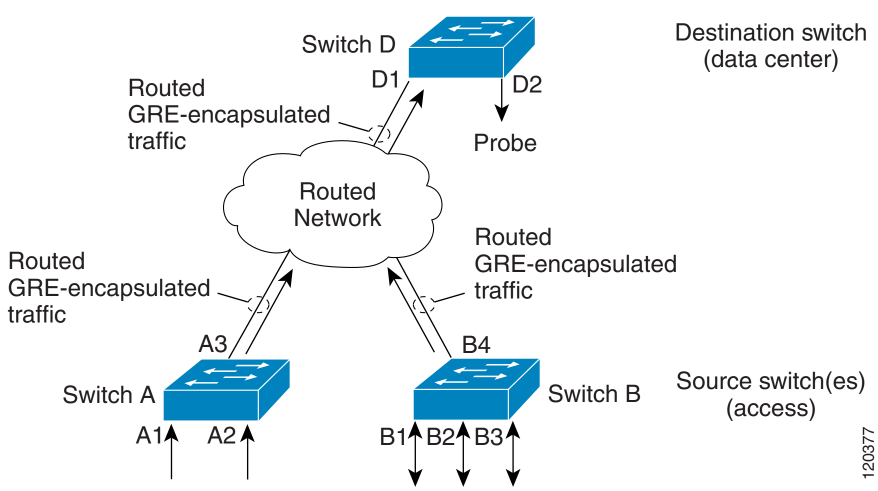
To configure an ERSPAN source session on one switch, you associate a set of source ports or VLANs with a destination IP address, ERSPAN ID number, and optionally with a VRF name. To configure an ERSPAN destination session on another switch, you associate the destinations with the source IP address, ERSPAN ID number, and optionally with a VRF name.

ERSPAN source sessions do not copy locally sourced RSPAN VLAN traffic from source trunk ports that carry RSPAN VLANs. ERSPAN source sessions do not copy locally sourced ERSPAN GRE-encapsulated traffic from source ports.

Each ERSPAN source session can have either ports or VLANs as sources, but not both.

The ERSPAN source session copies traffic from the source ports or source VLANs and forwards the traffic using routable GRE-encapsulated packets to the ERSPAN destination session. The ERSPAN destination session switches the traffic to the destinations.

Figure 68-3 ERSPAN Configuration



Understanding the Traffic Monitored at SPAN Sources

These sections describe the traffic that local SPAN, RSPAN, and ERSPAN sources can monitor:

- Monitored Traffic Direction, page 68-5
- Monitored Traffic Type, page 68-5
- Duplicate Traffic, page 68-5

Monitored Traffic Direction

You can configure local SPAN sessions, RSPAN source sessions, and ERSPAN source sessions to monitor the following traffic:

- Ingress traffic
  - Called ingress SPAN.
  - Copies traffic received by the sources (ingress traffic).
  - Ingress traffic is sent to the supervisor engine SPAN ASIC to be copied.
- Egress traffic
  - Called egress SPAN.
  - Copies traffic transmitted from the sources (egress traffic).
  - Distributed egress SPAN mode—With Release 12.2(33)SXH and later releases, on some fabric-enabled switching modules, egress traffic can be copied locally by the switching module SPAN ASIC and then sent to the SPAN destinations. See the "Distributed Egress SPAN Mode Guidelines and Restrictions" section on page 68-14 for information about switching modules that support distributed egress SPAN mode.
  - Centralized egress SPAN mode—On all other switching modules, egress traffic is sent to the supervisor engine SPAN ASIC to be copied and is then sent to the SPAN destinations.
- Both
  - Copies both the received traffic and the transmitted traffic (ingress and egress traffic).
  - Both ingress traffic and egress traffic is sent to the supervisor engine SPAN ASIC to be copied.

Monitored Traffic Type

By default, local SPAN and ERSPAN monitor all traffic, including multicast and bridge protocol data unit (BPDU) frames. RSPAN does not support BPDU monitoring.

Duplicate Traffic

In some configurations, SPAN sends multiple copies of the same source traffic to the destination. For example, in a configuration with a bidirectional SPAN session (both ingress and egress) for two SPAN sources, called s1 and s2, to a SPAN destination, called d1, if a packet enters the switch through s1 and is sent for egress from the switch to s2, ingress SPAN at s1 sends a copy of the packet to SPAN destination d1 and egress SPAN at s2 sends a copy of the packet to SPAN destination d1. If the packet was Layer 2 switched from s1 to s2, both SPAN packets would be the same. If the packet was Layer 3 switched from s1 to s2, the Layer 3 rewrite would alter the source and destination Layer 2 addresses, in which case the SPAN packets would be different.

Local SPAN, RSPAN, and ERSPAN Sources

These sections describe local SPAN, RSPAN, and ERSPAN sources:

- Source CPUs, page 68-6
- Source Ports and EtherChannels, page 68-6
- Source VLANs, page 68-6

Source CPUs

A source CPU is a CPU monitored for traffic analysis. With Release 12.2(33)SXH and later releases, you can configure both the SP CPU and the RP CPU as SPAN sources. These are examples of what you can do with the data generated by CPU monitoring:

- Develop baseline information about CPU traffic.
- Develop information to use when creating control plane policing (CoPP) policies.
- Troubleshoot CPU-related issues (for example, high CPU utilization).

Note
- CPU SPAN monitors CPU traffic from the perspective of the ASICs that send and receive the CPU traffic, rather than from onboard the CPUs themselves.
- Traffic to and from the CPU is tagged with VLAN IDs. You can configure source VLAN filtering of the CPU traffic.

Source Ports and EtherChannels

A source port or EtherChannel is a port or EtherChannel monitored for traffic analysis. You can configure both Layer 2 and Layer 3 ports and EtherChannels as SPAN sources. SPAN can monitor one or more source ports or EtherChannels in a single SPAN session. You can configure ports or EtherChannels in any VLAN as SPAN sources. Trunk ports or EtherChannels can be configured as sources and mixed with nontrunk sources.

Note SPAN does not copy the encapsulation from trunk sources. You can configure SPAN destinations as trunks to tag the monitored traffic before it is transmitted for analysis.

Source VLANs

A source VLAN is a VLAN monitored for traffic analysis. VLAN-based SPAN (VSPAN) uses a VLAN as the SPAN source. All the ports and EtherChannels in the source VLANs become sources of SPAN traffic.

Note Layer 3 VLAN interfaces on source VLANs are not sources of SPAN traffic. Traffic that enters a VLAN through a Layer 3 VLAN interface is monitored when it is transmitted from the switch through an egress port or EtherChannel that is in the source VLAN.

Local SPAN, RSPAN, and ERSPAN Destinations

A SPAN destination is a Layer 2 or Layer 3 port or, with Release 12.2(33)SXH and later releases, an EtherChannel, to which local SPAN, RSPAN, or ERSPAN sends traffic for analysis. When you configure a port or EtherChannel as a SPAN destination, it is dedicated for use only by the SPAN feature.

Destination EtherChannels do not support the Port Aggregation Control Protocol (PAgP) or Link Aggregation Control Protocol (LACP) EtherChannel protocols; only the on mode is supported, with all EtherChannel protocol support disabled.

There is no requirement that the member links of a destination EtherChannel be connected to a device that supports EtherChannels. For example, you can connect the member links to separate network analyzers. See Chapter 19, "Configuring EtherChannels," for more information about EtherChannel.

Destinations, by default, cannot receive any traffic. With Release 12.2(33)SXH and later releases, you can configure Layer 2 destinations to receive traffic from any attached devices.

Destinations, by default, do not transmit anything except SPAN traffic. Layer 2 destinations that you have configured to receive traffic can be configured to learn the Layer 2 address of any devices attached to the destination and transmit traffic that is addressed to the devices.

You can configure trunks as destinations, which allows trunk destinations to transmit encapsulated traffic. You can use allowed VLAN lists to configure destination trunk VLAN filtering.

Local SPAN, RSPAN, and ERSPAN Configuration Guidelines and Restrictions

These sections describe local SPAN, RSPAN, and ERSPAN configuration guidelines and restrictions:

General Guidelines and Restrictions

Use SPAN for troubleshooting. Except in carefully planned topologies, SPAN consumes too many switch and network resources to enable permanently.

Exercise all possible care when enabling and configuring SPAN. The traffic copied by SPAN can impose a significant load on the switch and the network.

To minimize the load, configure SPAN to copy only the specific traffic that you want to analyze. Select sources that carry as little unwanted traffic as possible. For example, a port as a SPAN source might carry less unwanted traffic than a VLAN.

Note To monitor traffic that can be matched with an ACL, consider using VACL capture.

Before enabling SPAN, carefully evaluate the SPAN source traffic rates, and consider the performance implications and possible oversubscription points, which include these:

- SPAN destination
- Fabric channel
- Rewrite/replication engine
- Forwarding engine (PFC/DFC)

To avoid disrupting traffic, do not oversubscribe any of these points in your SPAN topology. Some oversubscription and performance considerations are:

- SPAN doubles traffic internally
- SPAN adds to the traffic being processed by the switch fabric
- SPAN doubles forwarding engine load
- The supervisor engine handles the entire load imposed by egress SPAN (also called transmit SPAN).

  Note Egress SPAN should only be enabled for short periods of time during active troubleshooting.

  Release 12.2(33)SXH and later releases support distributed egress SPAN, which reduces the load on the supervisor engine.

- The ingress modules handle the load imposed by ingress SPAN sources (also called receive SPAN) on each module. Ingress SPAN adds to rewrite/replication engine load.

Feature Incompatibilities

These feature incompatibilities exist with local SPAN, RSPAN, and ERSPAN:

- In releases where CSCth62957 is not resolved, in PFC3B mode or PFC3BXL mode, the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command might cause traffic to loop continuously with these SPAN configurations:
  - If the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command is configured on a physical interface, the CLI prevents configuration of that port as part of a SPAN session.
  - If a SPAN session is configured on a physical port and you attempt to configure the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command, the CLI prints a warning that recommends against the configuration.

- If the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command is configured on a physical interface, you should not configure **source cpu** {**rp** | **sp**} in any SPAN session, but the CLI does not enforce any restriction.
- If a SPAN session is configured with **source cpu** {**rp** | **sp**} and you attempt to configure the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command, the CLI does not enforce any restriction.

- In releases where CSCth62957 is resolved, to avoid a configuration that might cause traffic to loop continuously, the CLI enforces these restrictions in PFC3B mode or PFC3BXL mode:
  - If the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command is configured on a physical interface, the CLI prevents configuration of that port as part of a SPAN session.
  - If a SPAN session is configured on a physical port and you attempt to configure the the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command command on that port, the CLI prints a warning that recommends against the configuration.
  - If the the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command command is configured on a physical interface, you cannot configure **source cpu** {**rp** | **sp**} in any SPAN session.
  - If a SPAN session is configured with **source cpu** {**rp** | **sp**} and you attempt to configure the **xconnect** *target_ip_address vc_value* **encapsulation mpls** command, the CLI prints a warning that recommends against the configuration.
- Egress SPAN is not supported in egress multicast mode. (CSCsa95965)
- Unknown unicast flood blocking (UUFB) ports cannot be RSPAN or local SPAN egress-only destinations. (CSCsj27695)
- Except in PFC3C mode or PFC3CXL mode, Ethernet over MultiProtocol Label Switching (EoMPLS) ports cannot be SPAN sources. (CSCed51245)
- A port-channel interface (an EtherChannel) can be a SPAN source, but you cannot configure active member ports of an EtherChannel as SPAN source ports. Inactive member ports of an EtherChannel can be configured as SPAN sources but they are put into the suspended state and carry no traffic.
- These features are incompatible with SPAN destinations:
  - Private VLANs
  - IEEE 802.1X port-based authentication
  - Port security
  - Spanning Tree Protocol (STP) and related features (PortFast, PortFast BPDU filtering, BPDU Guard, UplinkFast, BackboneFast, EtherChannel Guard, Root Guard, Loop Guard)
  - VLAN trunk protocol (VTP)
  - Dynamic trunking protocol (DTP)
  - IEEE 802.1Q tunneling

Note

- SPAN destinations can participate in IEEE 802.3Z flow control.
- IP multicast switching using egress packet replication is not compatible with SPAN. In some cases, egress replication can result in multicast packets not being sent to the SPAN destination port. If you are using SPAN and your switching modules are capable of egress replication, enter the **mls ip multicast replication-mode ingress** command to force ingress replication.

Local SPAN, RSPAN, and ERSPAN Session Limits

With Release 12.2(33)SXH and later releases, these are the PFC3 local SPAN, RSPAN, and ERSPAN session limits:

	Local and Source Sessions		Destination Sessions	
Total Sessions	**Local SPAN, RSPAN Source, ERSPAN Source Ingress or Egress or Both**	**Local SPAN Egress-Only**	**RSPAN**	**ERSPAN**
80	2	14	64	23

Local SPAN, RSPAN, and ERSPAN Interface Limits

With Release 12.2(33)SXH and later releases, these are the PFC3 local SPAN, RSPAN, and ERSPAN source and destination interface limits:

	In Each Local SPAN Session	In Each RSPAN Source Session	In Each ERSPAN Source Session	In Each RSPAN Destination Session	In Each ERSPAN Destination Session
Egress or "both" sources	128	128	128	—	—
Ingress sources	128	128	128	—	—
RSPAN and ERSPAN destination session sources	—	—	—	1 RSPAN VLAN	1 IP address
Destinations per session	64	1 RSPAN VLAN	1 IP address	64	64

Local SPAN, RSPAN, and ERSPAN Guidelines and Restrictions

These guidelines and restrictions apply to local SPAN, RSPAN, and ERSPAN:

- A SPAN destination that is copying traffic from a single egress SPAN source port sends only egress traffic to the network analyzer. If you configure more than one egress SPAN source port, the traffic that is sent to the network analyzer also includes these types of ingress traffic that were received from the egress SPAN source ports:
  - Any unicast traffic that is flooded on the VLAN
  - Broadcast and multicast traffic

  This situation occurs because an egress SPAN source port receives these types of traffic from the VLAN but then recognizes itself as the source of the traffic and drops it instead of sending it back to the source from which it was received. Before the traffic is dropped, SPAN copies the traffic and sends it to the SPAN destination. (CSCds22021)
- Entering additional **monitor session** commands does not clear previously configured SPAN parameters. You must enter the **no monitor session** command to clear configured SPAN parameters.
- Connect a network analyzer to the SPAN destinations.

- Within a SPAN session, all of the SPAN destinations receive all of the traffic from all of the SPAN sources, except when source-VLAN filtering is configured on the SPAN source.
- You can configure destination trunk VLAN filtering to select which traffic is transmitted from the SPAN destination.
- You can configure both Layer 2 LAN ports (LAN ports configured with the **switchport** command) and Layer 3 LAN ports (LAN ports not configured with the **switchport** command) as sources or destinations.
- You cannot mix individual source ports and source VLANs within a single session.
- If you specify multiple ingress source ports, the ports can belong to different VLANs.
- Within a session, you cannot configure both VLANs as SPAN sources and do source VLAN filtering. You can configure VLANs as SPAN sources or you can do source VLAN filtering of traffic from source ports and EtherChannels, but not both in the same session.
- You cannot configure source VLAN filtering for internal VLANs.
- When enabled, local SPAN, RSPAN, and ERSPAN use any previously entered configuration.
- When you specify sources and do not specify a traffic direction (ingress, egress, or both), “both” is used by default.
- SPAN copies Layer 2 Ethernet frames, but SPAN does not copy source trunk port ISL or 802.1Q tags. You can configure destinations as trunks to send locally tagged traffic to the traffic analyzer.

Note A destination configured as a trunk tags traffic from a Layer 3 LAN source with the internal VLAN used by the Layer 3 LAN source.

- Local SPAN sessions, RSPAN source sessions, and ERSPAN source sessions do not copy locally sourced RSPAN VLAN traffic from source trunk ports that carry RSPAN VLANs.
- Local SPAN sessions, RSPAN source sessions, and ERSPAN source sessions do not copy locally sourced ERSPAN GRE-encapsulated traffic from source ports.
- With Release 12.2(33)SXH and later, SPAN sessions can share destinations.
- SPAN destinations cannot be SPAN sources.
- Destinations never participate in any spanning tree instance. Local SPAN includes BPDUs in the monitored traffic, so any BPDUs seen on the destination are from the source. RSPAN does not support BPDU monitoring.
- All packets forwarded through the switch for transmission from a port that is configured as an egress SPAN source are copied to the SPAN destination, including packets that do not exit the switch through the egress port because STP has put the egress port into the blocking state, or on an egress trunk port because STP has put the VLAN into the blocking state on the trunk port.

VSPAN Guidelines and Restrictions

Note Local SPAN, RSPAN, and ERSPAN all support VSPAN.

These are VSPAN guidelines and restrictions:

- VSPAN sessions do not support source VLAN filtering.
- For VSPAN sessions with both ingress and egress configured, two packets are forwarded from the destination to the analyzer if the packets get switched on the same VLAN (one as ingress traffic from the ingress port and one as egress traffic from the egress port).
- VSPAN only monitors traffic that leaves or enters Layer 2 ports in the VLAN.
  - If you configure a VLAN as an ingress source and traffic gets routed into the monitored VLAN, the routed traffic is not monitored because it never appears as ingress traffic entering a Layer 2 port in the VLAN.
  - If you configure a VLAN as an egress source and traffic gets routed out of the monitored VLAN, the routed traffic is not monitored because it never appears as egress traffic leaving a Layer 2 port in the VLAN.

RSPAN Guidelines and Restrictions

These are RSPAN guidelines and restrictions:

- All participating switches must be connected by Layer 2 trunks.
- Any network device that supports RSPAN VLANs can be an RSPAN intermediate device.
- Networks impose no limit on the number of RSPAN VLANs that the networks carry.
- Intermediate network devices might impose limits on the number of RSPAN VLANs that they can support.
- You must configure the RSPAN VLANs in all source, intermediate, and destination network devices. If enabled, the VLAN Trunking Protocol (VTP) can propagate configuration of VLANs numbered 1 through 1024 as RSPAN VLANs. You must manually configure VLANs numbered higher than 1024 as RSPAN VLANs on all source, intermediate, and destination network devices.
- If you enable VTP and VTP pruning, RSPAN traffic is pruned in the trunks to prevent the unwanted flooding of RSPAN traffic across the network.
- RSPAN VLANs can be used only for RSPAN traffic.
- Do not configure a VLAN used to carry management traffic as an RSPAN VLAN.
- Do not assign access ports to RSPAN VLANs. RSPAN puts access ports in an RSPAN VLAN into the suspended state.
- Do not configure any ports in an RSPAN VLAN except trunk ports selected to carry RSPAN traffic.
- MAC address learning is disabled in the RSPAN VLAN.
- You can use output access control lists (ACLs) on the RSPAN VLAN in the RSPAN source switch to filter the traffic sent to an RSPAN destination.
- RSPAN does not support BPDU monitoring.
- Do not configure RSPAN VLANs as sources in VSPAN sessions.
- You can configure any VLAN as an RSPAN VLAN as long as all participating network devices support configuration of RSPAN VLANs and you use the same RSPAN VLAN for each RSPAN session in all participating network devices.

ERSPAN Guidelines and Restrictions

These are ERSPAN guidelines and restrictions:

- A WS-SUP720 (a Supervisor Engine 720 manufactured with a PFC3A) can only support ERSPAN if it has hardware version 3.2 or higher. Enter the **show module version | include WS-SUP720-BASE** command to display the hardware version. For example:

```
Router# show module version | include WS-SUP720-BASE
 7    2  WS-SUP720-BASE     SAD075301SZ Hw :3.2
```

- For ERSPAN packets, the "protocol type" field value in the GRE header is 0x88BE.
- The payload of a Layer 3 ERSPAN packet is a copied Layer 2 Ethernet frame, excluding any ISL or 802.1Q tags.
- ERSPAN adds a 50-byte header to each copied Layer 2 Ethernet frame and replaces the 4-byte cyclic redundancy check (CRC) trailer.
- ERSPAN supports jumbo frames that contain Layer 3 packets of up to 9,202 bytes. If the length of the copied Layer 2 Ethernet frame is greater than 9,170 (9,152-byte Layer 3 packet), ERSPAN truncates the copied Layer 2 Ethernet frame to create a 9,202-byte ERSPAN Layer 3 packet.

Note The Layer 3 IP header in truncated packets retains the nontruncated Layer 3 packet size. The length consistency check between the Layer 2 frame and the Layer 3 packet on ERSPAN destinations that are 6500 switches drops truncated ERSPAN packets unless you configure the **no mls verify ip length consistent** global configuration command on the ERSPAN destination 6500 switch.

- Regardless of any configured MTU size, ERSPAN creates Layer 3 packets that can be as long as 9,202 bytes. ERSPAN traffic might be dropped by any interface in the network that enforces an MTU size smaller than 9,202 bytes.
- With the default MTU size (1,500 bytes), if the length of the copied Layer 2 Ethernet frame is greater than 1,468 bytes (1,450-byte Layer 3 packet), the ERSPAN traffic is dropped by any interface in the network that enforces the 1,500-byte MTU size.

Note The **mtu** interface command and the **system jumbomtu** command (see the "Configuring Jumbo Frame Support" section on page 8-10) set the maximum Layer 3 packet size (default is 1,500 bytes, maximum is 9,216 bytes).

- All participating switches must be connected at Layer 3 and the network path must support the size of the ERSPAN traffic.
- ERSPAN does not support packet fragmentation. The "do not fragment" bit is set in the IP header of ERSPAN packets. ERSPAN destination sessions cannot reassemble fragmented ERSPAN packets.
- ERSPAN traffic is subject to the traffic load conditions of the network. You can set the ERSPAN packet IP precedence or DSCP value to prioritize ERSPAN traffic for QoS.
- The only supported destination for ERSPAN traffic is an ERSPAN destination session on a PFC3.
- All ERSPAN source sessions on a switch must use the same origin IP address, configured with the **origin ip address** command (see the "Configuring ERSPAN Source Sessions" section on page 68-30).

- All ERSPAN destination sessions on a switch must use the same IP address on the same destination interface. You enter the destination interface IP address with the **ip address** command (see the "Configuring ERSPAN Destination Sessions" section on page 68-32).
- The ERSPAN source session's destination IP address, which must be configured on an interface on the destination switch, is the source of traffic that an ERSPAN destination session sends to the destinations. You configure the same address in both the source and destination sessions with the **ip address** command.
- The ERSPAN ID differentiates the ERSPAN traffic arriving at the same destination IP address from various different ERSPAN source sessions.

Distributed Egress SPAN Mode Guidelines and Restrictions

These are distributed egress SPAN mode guidelines and restrictions:

- These switching modules disable distributed egress SPAN mode:
  - WS-X6502-10GE
  - WS-X6816-GBIC
  - WS-X6516-GBIC
  - WS-X6516-GE-TX
  - WS-X6524-100FX-MM
  - WS-X6548-RJ-45
  - WS-X6548-RJ-21

  With any of these switching modules installed, the egress SPAN mode is centralized.

  Enter the **show monitor session egress replication-mode | include Operational|slot** command to display any switching modules that disable distributed egress SPAN mode. If there are no modules installed that disable distributed egress SPAN mode, the command displays only the egress SPAN operational mode.
- Some switching modules have ASICs that do not support distributed egress SPAN mode for ERSPAN sources.

  Enter the **show monitor session egress replication-mode | include Distributed.*Distributed.*Centralized** command to display the slot number of any switching modules that do not support distributed egress SPAN mode for ERSPAN sources.

  Enter the **show asic-version slot** *slot_number* command to display the versions of the ASICs on the switching module in the slot where distributed egress SPAN mode is not supported for ERSPAN sources.

  Hyperion ASIC revision levels 5.0 and higher and all versions of the Metropolis ASIC support distributed egress SPAN mode for ERSPAN sources. Switching modules with Hyperion ASIC revision levels lower than 5.0 do not support distributed egress SPAN mode for ERSPAN sources.

Configuring Local SPAN, RSPAN, and ERSPAN

These sections describe how to configure local SPAN, RSPAN, and ERSPAN:

Local SPAN, RSPAN, and ERSPAN Default Configuration

This section describes the local SPAN, RSPAN, and ERSPAN default configuration:

Feature		Default Value
Local SPAN		Disabled
RSPAN		Disabled
ERSPAN		Disabled
Default operating mode for egress SPAN sessions:		
	Releases earlier than Release 12.2(33)SXH:	Centralized
	Release 12.2(33)SXH:	Distributed
	Release 12.2(33)SXH1:	Distributed
	Release 12.2(33)SXH2:	Distributed
	Release 12.2(33)SXH2a:	Centralized

Configuring a Destination as an Unconditional Trunk (Optional)

To tag the monitored traffic as it leaves a destination, configure the destination as a trunk before you configure it as a destination.

To configure the destination as a trunk, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface {type[1] slot/port | port-channel number}`	Selects the interface to configure.
Step 3	`Router(config-if)# switchport`	Configures the interface for Layer 2 switching (required only if the interface is not already configured for Layer 2 switching).
Step 4	`Router(config-if)# switchport trunk encapsulation {isl | dot1q}`	Configures the encapsulation, which configures the interface as either an ISL or 802.1Q trunk.
Step 5	`Router(config-if)# switchport mode trunk`	Configures the interface to trunk unconditionally.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure a port as an unconditional IEEE 802.1Q trunk:

```
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
```

Note Releases earlier than Release 12.2(33)SXH required you to enter the **switchport nonegotiate** command when you configured a destination port as an unconditional trunk. This requirement has been removed in Release 12.2(33)SXH and later releases.

Configuring Destination Trunk VLAN Filtering (Optional)

Note

- In addition to filtering VLANs on a trunk, you can also apply the allowed VLAN list to access ports.
- Destination trunk VLAN filtering is applied at the destination. Destination trunk VLAN filtering does not reduce the amount of traffic being sent from the SPAN sources to the SPAN destinations.

When a destination is a trunk, you can use the list of VLANs allowed on the trunk to filter the traffic transmitted from the destination. (CSCeb01318)

Destination trunk VLAN filtering removes the restriction that, within a SPAN session, all destinations receive all the traffic from all the sources. Destination trunk VLAN filtering allows you to select, on a per-VLAN basis, the traffic that is transmitted from each destination trunk to the network analyzer.

To configure destination trunk VLAN filtering on a destination trunk, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface`** *`type`*[1] *`slot/port`*	Selects the destination trunk port to configure.
Step 3	`Router(config-if)#` **`switchport trunk allowed vlan`** {**`add`** | **`except`** | **`none`** | **`remove`**} *`vlan`* [,*`vlan`*[,*`vlan`*[,...]]	Configures the list of VLANs allowed on the trunk.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When configuring the list of VLANs allowed on a destination trunk port, note the following information:

- The *vlan* parameter is either a single VLAN number from 1 through 4094, or a range of VLANs described by two VLAN numbers, the lesser one first, separated by a dash. Do not enter any spaces between comma-separated *vlan* parameters or in dash-specified ranges.
- All VLANs are allowed by default.
- To remove all VLANs from the allowed list, enter the **switchport trunk allowed vlan none** command.
- To add VLANs to the allowed list, enter the **switchport trunk allowed vlan add** command.
- You can modify the allowed VLAN list without removing the SPAN configuration.

This example shows the configuration of a local SPAN session that has several VLANs as sources and several trunk ports as destinations, with destination trunk VLAN filtering that filters the SPAN traffic so that each destination trunk port transmits the traffic from one VLAN:

```
interface GigabitEthernet1/1
description SPAN destination interface for VLAN 10
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/2
description SPAN destination interface for VLAN 11
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 11
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/3
description SPAN destination interface for VLAN 12
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 12
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/4
description SPAN destination interface for VLAN 13
no ip address
switchport
```

```
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 13
switchport mode trunk
switchport nonegotiate
!
monitor session 1 source vlan 10 - 13
monitor session 1 destination interface Gi1/1 - 4
```

Configuring Destination Port Permit Lists (Optional)

To prevent accidental configuration of ports as destinations, you can create a permit list of the ports that are valid for use as destinations. With a destination port permit list configured, you can only configure the ports in the permit list as destinations.

To configure a destination port permit list, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`monitor permit-list`**	Enables use of the destination port permit list.
Step 3	`Router(config)#` **`monitor permit-list destination interface`** *`type`*[1] *`slot/port`*`[-`*`port`*`] [,` *`type`*[1] *`slot/port`* `-` *`port`*`]`	Configures a destination port permit list or adds to an existing destination port permit list.
Step 4	`Router(config)#` **`do show monitor permit-list`**	Verifies the configuration.

1. *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure a destination port permit list that includes Gigabit Ethernet ports 5/1 through 5/4 and 6/1:

```
Router# configure terminal
Router(config)# monitor permit-list
Router(config)# monitor permit-list destination interface gigabitethernet 5/1-4,
gigabitethernet 6/1
```

This example shows how to verify the configuration:

```
Router(config)# do show monitor permit-list
 SPAN Permit-list       :Admin Enabled
 Permit-list ports      :Gi5/1-4,Gi6/1
```

Configuring the Egress SPAN Mode (Optional)

With Release 12.2(33)SXH, Release 12.2(33)SXH1, and Release 12.2(33)SXH2, distributed egress SPAN mode is the default if there are no switching modules installed that disable it.

With Release 12.2(33)SXH2a and later releases, centralized egress SPAN mode is the default.

See the "Distributed Egress SPAN Mode Guidelines and Restrictions" section on page 68-14 for information about switching modules that support distributed egress SPAN mode.

With Release 12.2(33)SXH2a and later releases, to configure the egress SPAN mode, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session egress replication-mode distributed`	Enables distributed egress SPAN mode. **Note** Enter the **no monitor session egress replication-mode distributed** command to enable centralized egress SPAN mode.
Step 3	`Router(config)# end`	Exits configuration mode.

This example shows how to enable distributed egress SPAN mode:

```
Router# configure terminal
Router(config)# monitor session egress replication-mode distributed
Router(config)# end
```

With Release 12.2(33)SXH, Release 12.2(33)SXH1, and Release 12.2(33)SXH2, to configure the egress SPAN mode, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session egress replication-mode centralized`	Enables centralized egress SPAN mode. **Note** Enter the **no monitor session egress replication-mode centralized** command to enable distributed egress SPAN mode.
Step 3	`Router(config)# end`	Exits configuration mode.

This example shows how to disable distributed egress SPAN mode:

```
Router# configure terminal
Router(config)# monitor session egress replication-mode centralized
Router(config)# end
```

This example shows how to display the configured egress SPAN mode:

```
Router# show monitor session egress replication-mode | include Configured
Configured mode   : Centralized
```

Configuring Local SPAN

These sections describe how to configure local SPAN sessions:

- Configuring Local SPAN (SPAN Configuration Mode), page 68-20
- Configuring Local SPAN (Global Configuration Mode), page 68-22

Configuring Local SPAN (SPAN Configuration Mode)

Note To tag the monitored traffic as it leaves a destination, you must configure the destination to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination as an Unconditional Trunk (Optional)" section on page 68-16).

To configure a local SPAN session in SPAN configuration mode, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`monitor session`** *`local_SPAN_session_number`* **`type`** `[`**`local`** `|` **`local-tx`**`]`	Configures a local SPAN session number and enters local SPAN session configuration mode. **Note** • Enter the **local** keyword to configure ingress or egress or both SPAN sessions. • Enter the **local-tx** keyword to configure egress-only SPAN sessions.
Step 3	`Router(config-mon-local)#` **`description`** *`session_description`*	(Optional) Describes the local SPAN session.
Step 4	`Router(config-mon-local)#` **`source`** `{{`**`cpu`** `{`**`rp`** `|` **`sp`**`}} |` *`single_interface`* `|` *`interface_list`* `|` *`interface_range`* `|` *`mixed_interface_list`* `|` *`single_vlan`* `|` *`vlan_list`* `|` *`vlan_range`* `|` *`mixed_vlan_list`*`} [`**`rx`** `|` **`tx`** `|` **`both`**`]`	Associates the local SPAN session number with the CPU, source ports, or VLANs, and selects the traffic direction to be monitored. **Note** • When you enter the **local-tx** keyword, the **rx** and **both** keywords are not available and the **tx** keyword is required. • To make best use of the available SPAN sessions, it is always preferable to configure **local-tx** sessions instead of **local** sessions with the **tx** keyword.
Step 5	`Router(config-mon-local)#` **`filter`** *`single_vlan`* `|` *`vlan_list`* `|` *`vlan_range`* `|` *`mixed_vlan_list`*	(Optional) Configures source VLAN filtering when the local SPAN source is a trunk port.
Step 6	`Router(config-mon-local)#` **`destination`** `{`*`single_interface`* `|` *`interface_list`* `|` *`interface_range`* `|` *`mixed_interface_list`*`} [`**`ingress`** `[`**`learning`**`]]`	Associates the local SPAN session number with the destinations.
Step 7	`Router(config-mon-local)#` **`no shutdown`**	Activates the local SPAN session. **Note** The **no shutdown** command and **shutdown** commands are not supported for local-tx egress-only SPAN sessions.
Step 8	`Router(config-mon-local)#` **`end`**	Exits configuration mode.

When configuring monitor sessions, note the following information:

- *session_description* can be up to 240 characters and cannot contain special characters; with Release 12.2(33)SXH and later releases, the description can contain spaces.

  Note You can enter 240 characters after the **description** command.

- *local_span_session_number* can range from 1 to 80.
- **cpu rp** is the route processor (RP).
- **cpu sp** is the switch processor (SP).
- *single_interface* is as follows:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

    Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 19-8.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the switch to transmit traffic that is addressed to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:
  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 17-6.
  - If the destination is a trunk and the attached device transmits tagged traffic back to the switch, you can use either ISL or 802.1Q trunking.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the switch, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.
  - Do not configure the destinations with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.

- Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

This example shows how to configure session 1 to monitor ingress traffic from Gigabit Ethernet port 1/1 and configure Gigabit Ethernet port 1/2 as the destination:

```
Router(config)# monitor session 1 type local
Router(config-mon-local)# source interface gigabitethernet 1/1 rx
Router(config-mon-local)# destination interface gigabitethernet 1/2
```

For additional examples, see the "Configuration Examples" section on page 68-35.

Configuring Local SPAN (Global Configuration Mode)

Note

- To tag the monitored traffic as it leaves a destination, you must configure the destination to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination as an Unconditional Trunk (Optional)" section on page 68-16).
- You can configure up to two local SPAN sessions in global configuration mode.
- You can use SPAN configuration mode for all SPAN configuration tasks.
- You must use SPAN configuration mode to configure the supported maximum number of SPAN sessions.

Local SPAN does not use separate source and destination sessions. To configure a local SPAN session, configure local SPAN sources and destinations with the same session number. To configure a local SPAN session, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **monitor session** *local_span_session_number* **source** {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list* | *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*} [**rx** | **tx** | **both**]	Associates the local SPAN source session number with the source ports or VLANs and selects the traffic direction to be monitored.
Step 3	Router(config)# **monitor session** *local_span_session_number* **destination** {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list*} [**ingress** [**learning**]]	Associates the local SPAN session number and the destinations.

When configuring local SPAN sessions, note the following information:

- *local_span_session_number* can range from 1 to 80.
- *single_interface* is as follows:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 19-8.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID* - *last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the switch to transmit traffic that is addressed to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:

  – Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 17-6.

  – If the destination is a trunk and the attached device transmits tagged traffic back to the switch, you can use either ISL or 802.1Q trunking.

  – If the destination is a trunk and the attached device transmits untagged traffic back to the switch, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.

  – Do not configure the destinations with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.

  – Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

This example shows how to configure Fast Ethernet port 5/1 as a bidirectional source for session 1:

```
Router(config)# monitor session 1 source interface fastethernet 5/1
```

This example shows how to configure Fast Ethernet port 5/48 as the destination for SPAN session 1:

```
Router(config)# monitor session 1 destination interface fastethernet 5/48
```

For additional examples, see the "Configuration Examples" section on page 68-35.

Configuring RSPAN

RSPAN uses a source session on one switch and a destination session on a different switch. These sections describe how to configure RSPAN sessions:

Configuring RSPAN VLANs

To configure a VLAN as an RSPAN VLAN, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# vlan vlan_ID[{-|,}vlan_ID]`	Creates or modifies an Ethernet VLAN, a range of Ethernet VLANs, or several Ethernet VLANs specified in a comma-separated list (do not enter space characters).
Step 3	`Router(config-vlan)# remote-span`	Configures the VLAN as an RSPAN VLAN.
Step 4	`Router(config-vlan)# end`	Updates the VLAN database and returns to privileged EXEC mode.

Configuring RSPAN Sessions (SPAN Configuration Mode)

These sections describe how to configure RSPAN sessions in SPAN configuration mode:

- Configuring RSPAN Source Sessions in SPAN Configuration Mode, page 68-24
- Configuring RSPAN Destination Sessions in SPAN Configuration Mode, page 68-25

Configuring RSPAN Source Sessions in SPAN Configuration Mode

To configure an RSPAN source session in SPAN configuration mode, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session RSPAN_source_session_number type rspan-source`	Configures an RSPAN source session number and enters RSPAN source session configuration mode for the session.
Step 3	`Router(config-mon-rspan-src)# description session_description`	(Optional) Describes the RSPAN source session.
Step 4	`Router(config-mon-rspan-src)# source {{cpu {rp | sp}} | single_interface | interface_list | interface_range | mixed_interface_list | single_vlan | vlan_list | vlan_range | mixed_vlan_list} [rx | tx | both]`	Associates the RSPAN source session number with the CPU, source ports, or VLANs, and selects the traffic direction to be monitored.
Step 5	`Router(config-mon-rspan-src)# filter single_vlan | vlan_list | vlan_range | mixed_vlan_list`	(Optional) Configures source VLAN filtering when the RSPAN source is a trunk port.
Step 6	`Router(config-mon-rspan-src)# destination remote vlan rspan_vlan_ID`	Associates the RSPAN source session number session number with the RSPAN VLAN.
Step 7	`Router(config-mon-rspan-src)# no shutdown`	Activates the RSPAN source session.
Step 8	`Router(config-mon-rspan-src)# end`	Exits configuration mode.

When configuring RSPAN source sessions, note the following information:

- *session_description* can be up to 240 characters and cannot contain special characters; with Release 12.2(33)SXH and later releases, the description can contain spaces.

  Note You can enter 240 characters after the **description** command.

- *RSPAN_source_span_session_number* can range from 1 to 80.
- **cpu rp** is the route processor (RP).
- **cpu sp** is the switch processor (SP).
- *single_interface* is as follows:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- See the "Configuring RSPAN VLANs" section on page 68-24 for information about the RSPAN VLAN ID.

This example shows how to configure session 1 to monitor bidirectional traffic from Gigabit Ethernet port 1/1:

```
Router(config)# monitor session 1 type rspan-source
Router(config-mon-rspan-src)# source interface gigabitethernet 1/1
Router(config-mon-rspan-src)# destination remote vlan 2
```

For additional examples, see the "Configuration Examples" section on page 68-35.

Configuring RSPAN Destination Sessions in SPAN Configuration Mode

Note

- To tag the monitored traffic, you must configure the port to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination as an Unconditional Trunk (Optional)" section on page 68-16).
- You can configure an RSPAN destination session on the RSPAN source session switch to monitor RSPAN traffic locally.

To configure an RSPAN destination session, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session RSPAN_destination_session_number type rspan-destination`	Configures an RSPAN destination session number and enters RSPAN destination session configuration mode for the session.
Step 3	`Router(config-mon-rspan-dst)# description session_description`	(Optional) Describes the RSPAN destination session.
Step 4	`Router(config-mon-rspan-dst)# source remote vlan rspan_vlan_ID`	Associates the RSPAN destination session number with the RSPAN VLAN.
Step 5	`Router(config-mon-rspan-dst)# destination {single_interface | interface_list | interface_range | mixed_interface_list} [ingress [learning]]`	Associates the RSPAN destination session number with the destinations.
Step 6	`Router(config-mon-rspan-dst)# end`	Exits configuration mode.

When configuring RSPAN destination sessions, note the following information:

- *RSPAN_destination_span_session_number* can range from 1 to 80.
- *single_interface* is as follows:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

    Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 19-8.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the switch to transmit traffic that is addressed to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:
  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 17-6.
  - If the destination is a trunk and the attached device transmits tagged traffic back to the switch, you can use either ISL or 802.1Q trunking.

- If the destination is a trunk and the attached device transmits untagged traffic back to the switch, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.
- Do not configure the destinations with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.
- Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

- The **no shutdown** command and **shutdown** commands are not supported for RSPAN destination sessions.

This example shows how to configure RSPAN VLAN 2 as the source for session 1 and Gigabit Ethernet port 1/2 as the destination:

```
Router(config)# monitor session 1 type rspan-destination
Router(config-rspan-dst)# source remote vlan 2
Router(config-rspan-dst)# destination interface gigabitethernet 1/2
```

For additional examples, see the "Configuration Examples" section on page 68-35.

Configuring RSPAN Sessions (Global Configuration Mode)

These sections describe how to configure RSPAN sessions in global configuration mode:

Configuring RSPAN Source Sessions in Global Configuration Mode

To configure an RSPAN source session in global configuration mode, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **monitor session** *RSPAN_source_session_number* **source** {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list* | *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*} [**rx** | **tx** | **both**]	Associates the RSPAN source session number with the source ports or VLANs, and selects the traffic direction to be monitored.
Step 3	Router(config)# **monitor session** *RSPAN_source_session_number* **destination remote vlan** *rspan_vlan_ID*	Associates the RSPAN source session number session number with the RSPAN VLAN.

When configuring RSPAN source sessions, note the following information:

- To configure RSPAN VLANs, see the "Configuring RSPAN VLANs" section on page 68-24.
- *RSPAN_source_span_session_number* can range from 1 to 80.

- *single_interface* is as follows:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- See the "Configuring RSPAN VLANs" section on page 68-24 for information about the RSPAN VLAN ID.

This example shows how to configure Fast Ethernet port 5/2 as the source for session 2:

```
Router(config)# monitor session 2 source interface fastethernet 5/2
```

This example shows how to configure RSPAN VLAN 200 as the destination for session 2:

```
Router(config)# monitor session 2 destination remote vlan 200
```

For additional examples, see the "Configuration Examples" section on page 68-35.

Configuring RSPAN Destination Sessions in Global Configuration Mode

Note
- To tag the monitored traffic, you must configure the port to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination as an Unconditional Trunk (Optional)" section on page 68-16).
- You can configure an RSPAN destination session on the RSPAN source session switch to monitor RSPAN traffic locally.

To configure an RSPAN destination session in global configuration mode, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **monitor session** *RSPAN_destination_session_number* **source remote vlan** *rspan_vlan_ID*	Associates the RSPAN destination session number with the RSPAN VLAN.
Step 3	Router(config)# **monitor session** *RSPAN_destination_session_number* **destination** {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list*} [**ingress** [**learning**]]	Associates the RSPAN destination session number with the destinations.

When configuring monitor sessions, note the following information:

- *RSPAN_destination_span_session_number* can range from 1 to 80.
- See the "Configuring RSPAN VLANs" section on page 68-24 for information about the RSPAN VLAN ID.
- *single_interface* is as follows:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

  Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 19-8.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the switch to transmit traffic that is addressed to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:
  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 17-6.
  - If the destination is a trunk and the attached device transmits tagged traffic back to the switch, you can use either ISL or 802.1Q trunking.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the switch, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.
  - Do not configure the destinations with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.
  - Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

This example shows how to configure RSPAN VLAN 200 as the source for session 3:

```
Router(config)# monitor session 3 source remote vlan 200
```

This example shows how to configure Fast Ethernet port 5/47 as the destination for session 3:

```
Router(config)# monitor session 3 destination interface fastethernet 5/47
```

For additional examples, see the "Configuration Examples" section on page 68-35.

Configuring ERSPAN

ERSPAN uses separate source and destination sessions. You configure the source and destination sessions on different switches. These sections describe how to configure ERSPAN sessions:

- Configuring ERSPAN Source Sessions, page 68-30
- Configuring ERSPAN Destination Sessions, page 68-32

Configuring ERSPAN Source Sessions

To configure an ERSPAN source session, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **monitor session** *ERSPAN_source_session_number* **type erspan-source**	Configures an ERSPAN source session number and enters ERSPAN source session configuration mode for the session.
Step 3	Router(config-mon-erspan-src)# **description** *session_description*	(Optional) Describes the ERSPAN source session.
Step 4	Router(config-mon-erspan-src)# **source** {{{**cpu** {**rp** | **sp**}} | *single_interface* | *interface_list* | *interface_range* | *mixed_interface_list* | *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*} [**rx** | **tx** | **both**]}	Associates the ERSPAN source session number with the CPU, source ports, or VLANs, and selects the traffic direction to be monitored.
Step 5	Router(config-mon-erspan-src)# **filter** *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*	(Optional) Configures source VLAN filtering when the ERSPAN source is a trunk port.
Step 6	Router(config-mon-erspan-src)# **destination**	Enters ERSPAN source session destination configuration mode.
Step 7	Router(config-mon-erspan-src-dst)# **ip address** *ip_address*	Configures the ERSPAN flow destination IP address, which must also be configured on an interface on the destination switch and be entered in the ERSPAN destination session configuration (see the "Configuring ERSPAN Destination Sessions" section on page 68-32, Step 6).
Step 8	Router(config-mon-erspan-src-dst)# **erspan-id** *ERSPAN_flow_id*	Configures the ID number used by the source and destination sessions to identify the ERSPAN traffic, which must also be entered in the ERSPAN destination session configuration (see the "Configuring ERSPAN Destination Sessions" section on page 68-32, Step 7).
Step 9	Router(config-mon-erspan-src-dst)# **origin ip address** *ip_address* [**force**]	Configures the IP address used as the source of the ERSPAN traffic.
Step 10	Router(config-mon-erspan-src-dst)# **ip ttl** *ttl_value*	(Optional) Configures the IP time-to-live (TTL) value of the packets in the ERSPAN traffic.
Step 11	Router(config-mon-erspan-src-dst)# **ip prec** *ipp_value*	(Optional) Configures the IP precedence value of the packets in the ERSPAN traffic.
Step 12	Router(config-mon-erspan-src-dst)# **ip dscp** *dscp_value*	(Optional) Configures the IP DSCP value of the packets in the ERSPAN traffic.

	Command	Purpose
Step 13	`Router(config-mon-erspan-src-dst)# vrf vrf_name`	(Optional) Configures the VRF name to use instead of the global routing table.
Step 14	`Router(config-mon-erspan-src)# no shutdown`	Activates the ERSPAN source session.
Step 15	`Router(config-mon-erspan-src-dst)# end`	Exits configuration mode.

When configuring monitor sessions, note the following information:

- *session_description* can be up to 240 characters and cannot contain special characters; with Release 12.2(33)SXH and later releases, the description can contain spaces.

  Note You can enter 240 characters after the **description** command.

- *ERSPAN_source_span_session_number* can range from 1 to 80.
- **cpu rp** is the route processor (RP).
- **cpu sp** is the switch processor (SP).
- *single_interface* is as follows:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

    Note Port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 19-8.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- *ERSPAN_flow_id* can range from 1 to 1023.
- All ERSPAN source sessions on a switch must use the same source IP address. Enter the **origin ip address** *ip_address* **force** command to change the origin IP address configured in all ERSPAN source sessions on the switch.
- *ttl_value* can range from 1 to 255.
- *ipp_value* can range from 0 to 7.
- *dscp_value* can range from 0 to 63.

This example shows how to configure session 3 to monitor bidirectional traffic from Gigabit Ethernet port 4/1:

```
Router(config)# monitor session 3 type erspan-source
Router(config-mon-erspan-src)# source interface gigabitethernet 4/1
Router(config-mon-erspan-src)# destination
Router(config-mon-erspan-src-dst)# ip address 10.1.1.1
Router(config-mon-erspan-src-dst)# origin ip address 20.1.1.1
Router(config-mon-erspan-src-dst)# erspan-id 101
```

For additional examples, see the "Configuration Examples" section on page 68-35.

Configuring ERSPAN Destination Sessions

Note You cannot monitor ERSPAN traffic locally.

To configure an ERSPAN destination session, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session ERSPAN_destination_session_number type erspan-destination`	Configures an ERSPAN destination session number and enters ERSPAN destination session configuration mode for the session.
Step 3	`Router(config-mon-erspan-dst)# description session_description`	(Optional) Describes the ERSPAN destination session.
Step 4	`Router(config-mon-erspan-dst)# destination {single_interface | interface_list | interface_range | mixed_interface_list} [ingress [learning]]`	Associates the ERSPAN destination session number with the destinations.
Step 5	`Router(config-mon-erspan-dst)# source`	Enters ERSPAN destination session source configuration mode.
Step 6	`Router(config-mon-erspan-dst-src)# ip address ip_address [force]`	Configures the ERSPAN flow destination IP address. This must be an address on a local interface and match the address that you entered in the "Configuring ERSPAN Source Sessions" section on page 68-30, Step 7.
Step 7	`Router(config-mon-erspan-dst-src)# erspan-id ERSPAN_flow_id`	Configures the ID number used by the destination and destination sessions to identify the ERSPAN traffic. This must match the ID that you entered in the "Configuring ERSPAN Source Sessions" section on page 68-30, Step 8.
Step 8	`Router(config-mon-erspan-dst-src)# vrf vrf_name`	(Optional) Configures the VRF name used instead of the global routing table.
Step 9	`Router(config-mon-erspan-dst)# no shutdown`	Activates the ERSPAN destination session.
Step 10	`Router(config-mon-erspan-dst-src)# end`	Exits configuration mode.

When configuring monitor sessions, note the following information:

- *ERSPAN_destination_span_session_number* can range from 1 to 80.
- *single_interface* is as follows:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

    Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 19-8.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- All ERSPAN destination sessions on a switch must use the same IP address on the same destination interface. Enter the **ip address** *ip_address* **force** command to change the IP address configured in all ERSPAN destination sessions on the switch.

  Note You must also change all ERSPAN source session destination IP addresses (see the "Configuring ERSPAN Source Sessions" section on page 68-30, Step 7).

- *ERSPAN_flow_id* can range from 1 to 1023.
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the switch to transmit traffic that is addressed to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:
  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 17-6.
  - If the destination is a trunk and the attached device transmits tagged traffic back to the switch, you can use either ISL or 802.1Q trunking.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the switch, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.
  - Do not configure the destinations with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.
  - Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

This example shows how to configure an ERSPAN destination session to send ERSPAN ID 101 traffic arriving at IP address 10.1.1.1 to Gigabit Ethernet port 2/1:

```
Router(config)# monitor session 3 type erspan-destination
Router(config-erspan-dst)# destination interface gigabitethernet 2/1
Router(config-erspan-dst)# source
Router(config-erspan-dst-src)# ip address 10.1.1.1
Router(config-erspan-dst-src)# erspan-id 101
```

For additional examples, see the "Configuration Examples" section on page 68-35.

Configuring Source VLAN Filtering in Global Configuration Mode

Note

- To configure source VLAN filtering in SPAN configuration mode, see the following sections:
  - Configuring Local SPAN (SPAN Configuration Mode), page 68-20
  - Configuring RSPAN Source Sessions in SPAN Configuration Mode, page 68-24
  - Configuring ERSPAN, page 68-30
- Source VLAN filtering reduces the amount of traffic that is sent from SPAN sources to SPAN destinations.

Source VLAN filtering monitors specific VLANs when the source is a trunk port.

To configure source VLAN filtering when the local SPAN or RSPAN source is a trunk port, perform this task:

	Command	Purpose
Step 1	Router# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **monitor session** *session_number* **filter** *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*	Configures source VLAN filtering when the local SPAN or RSPAN source is a trunk port.

When configuring source VLAN filtering, note the following information:

- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID* - *last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...

This example shows how to monitor VLANs 1 through 5 and VLAN 9 when the source is a trunk port:

```
Router(config)# monitor session 2 filter vlan 1 - 5 , 9
```

Verifying the Configuration

To verify the configuration, enter the **show monitor session** command.

This example shows how to verify the configuration of session 2:

```
Router# show monitor session 2
Session 2
------------
Type : Remote Source Session

Source Ports:
    RX Only:       Fa3/1
Dest RSPAN VLAN:   901
Router#
```

This example shows how to display the full details of session 2:

```
Router# show monitor session 2 detail
Session 2
------------
Type : Remote Source Session

Source Ports:
    RX Only:       Fa1/1-3
    TX Only:       None
    Both:          None
Source VLANs:
    RX Only:       None
    TX Only:       None
    Both:          None
Source RSPAN VLAN: None
Destination Ports: None
Filter VLANs:      None
Dest RSPAN VLAN:   901
```

Configuration Examples

This example shows the configuration of RSPAN source session 2:

```
Router(config)# monitor session 2 source interface fastethernet1/1 - 3 rx
Router(config)# monitor session 2 destination remote vlan 901
```

This example shows how to clear the configuration for sessions 1 and 2:

```
Router(config)# no monitor session range 1-2
```

This example shows the configuration of an RSPAN source session with multiple sources:

```
Router(config)# monitor session 2 source interface fastethernet 5/15 , 7/3 rx
Router(config)# monitor session 2 source interface gigabitethernet 1/2 tx
Router(config)# monitor session 2 source interface port-channel 102
Router(config)# monitor session 2 source filter vlan 2 - 3
Router(config)# monitor session 2 destination remote vlan 901
```

This example shows how to remove sources for a session:

```
Router(config)# no monitor session 2 source interface fastethernet 5/15 , 7/3
```

This example shows how to remove options for sources for a session:

```
Router(config)# no monitor session 2 source interface gigabitethernet 1/2
Router(config)# no monitor session 2 source interface port-channel 102 tx
```

This example shows how to remove source VLAN filtering for a session:

```
Router(config)# no monitor session 2 filter vlan 3
```

This example shows the configuration of RSPAN destination session 8:

```
Router(config)# monitor session 8 source remote vlan 901
Router(config)# monitor session 8 destination interface fastethernet 1/2 , 2/3
```

This example shows the configuration of ERSPAN source session 12:

```
monitor session 12 type erspan-source
 description SOURCE_SESSION_FOR_VRF_GRAY
 source interface Gi8/48 rx
 destination
  erspan-id 120
  ip address 10.8.1.2
  origin ip address 32.1.1.1
  vrf gray
```

This example shows the configuration of ERSPAN destination session 12:

```
monitor session 12 type erspan-destination
 description DEST_SESSION_FOR_VRF_GRAY
 destination interface Gi4/48
 source
  erspan-id 120
  ip address 10.8.1.2
  vrf gray
```

This example shows the configuration of ERSPAN source session 13:

```
monitor session 13 type erspan-source
 source interface Gi6/1 tx
 destination
  erspan-id 130
  ip address 10.11.1.1
  origin ip address 32.1.1.1
```

This example shows the configuration of ERSPAN destination session 13:

```
monitor session 13 type erspan-destination
 destination interface Gi6/1
 source
  erspan-id 130
  ip address 10.11.1.1
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Configuring SNMP IfIndex Persistence

This chapter describes how to configure the SNMP ifIndex persistence feature in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding SNMP IfIndex Persistence

The SNMP ifIndex persistence feature provides an interface index (ifIndex) value that is retained and used when the switch reboots. The ifIndex value is a unique identifying number associated with a physical or logical interface.

There is no requirement in the relevant RFCs that the correspondence between particular ifIndex values and their interfaces be maintained when the switch reboots, but many applications (for example, device inventory, billing, and fault detection) require maintenance of this correspondence.

You can poll the switch at regular intervals to correlate the interfaces to the ifIndexes, but it is not practical to poll constantly. The SNMP ifIndex persistence feature provides permanent ifIndex values, which eliminates the need to poll interfaces.

The following definitions are based on RFC 2233, "The Interfaces Group MIB using SMIv2." The following terms are values in the Interfaces MIB (IF-MIB):

- **ifIndex** —A unique number (greater than zero) that identifies each interface for SNMP identification of that interface.
- **ifName**—The text-based name of the interface, for example, "ethernet 3/1."
- **ifDescr—**A description of the interface. Recommended information for this description includes the name of the manufacturer, the product name, and the version of the interface hardware and software.

Configuring SNMP IfIndex Persistence

These sections describe how to configure SNMP ifIndex persistence:

- Enabling SNMP IfIndex Persistence Globally, page 69-2 (Optional)
- Enabling and Disabling SNMP IfIndex Persistence on Specific Interfaces, page 69-3 (Optional)

Note To verify that ifIndex commands have been configured, use the **more system:running-config** command.

Enabling SNMP IfIndex Persistence Globally

SNMP ifIndex persistence is disabled by default. To globally enableSNMP ifIndex persistence, perform this task:

Command	Purpose
`Router(config)# snmp-server ifindex persist`	Globally enables SNMP ifIndex persistence.

In the following example, SNMP ifIndex persistence is enabled for all interfaces:

```
router(config)# snmp-server ifindex persist
```

Disabling SNMP IfIndex Persistence Globally

To globally disable SNMP ifIndex persistence after enabling it, perform this task:

Command	Purpose
`Router(config)# no snmp-server ifindex persist`	Globally disables SNMP ifIndex persistence.

In the following example, SNMP ifIndex persistence is disabled for all interfaces:

```
router(config)# no snmp-server ifindex persist
```

Enabling and Disabling SNMP IfIndex Persistence on Specific Interfaces

To enable SNMP ifIndex persistence only on a specific interface, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {vlan vlan_ID}	{type[1] slot/port}	{port-channel port_channel_number}`	Selects an interface to configure.
Step 2	`Router(config-if)# snmp ifindex persist`	Enables SNMP ifIndex persistence on the specified interface.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.

1. *type* = any supported interface type.

Note The **[no] snmp ifindex persistence** interface command cannot be used on subinterfaces. A command applied to an interface is automatically applied to all the subinterfaces associated with that interface.

In the following example, SNMP ifIndex persistence is enabled for Ethernet interface 3/1 only:

```
router(config)# interface ethernet 3/1
router(config-if)# snmp ifindex persist
router(config-if)# exit
```

In the following example, SNMP ifIndex persistence is disabled for Ethernet interface 3/1 only:

```
router(config)# interface ethernet 3/1
router(config-if)# no snmp ifindex persist
router(config-if)# exit
```

Clearing SNMP IfIndex Persistence Configuration from a Specific Interface

To clear the interface-specific SNMP ifIndex persistence setting and configure the interface to use the global configuration setting, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Enters interface configuration mode for the specified interface. Note that the syntax of the interface command will vary depending on the platform you are using.
Step 2	`Router(config-if)# snmp ifindex clear`	Clears any interface-specific SNMP ifIndex persistence configuration for the specified interface and returns to the global configuration setting.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.

In the following example, any previous setting for SNMP ifIndex persistence on Ethernet interface 3/1 is removed from the configuration. If SNMP ifIndex persistence is globally enabled, SNMP ifIndex persistence will be enabled for Ethernet interface 3/1. If SNMP ifIndex persistence is globally disabled, SNMP ifIndex persistence will be disabled for Ethernet interface 3/1.

```
router(config)# interface ethernet 3/1
router(config-if)# snmp ifindex clear
router(config-if)# exit
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Using Top-N Reports

This chapter describes how to use Top-N reports in Cisco IOS Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding Top-N Reports

These sections describe Top-N reports:

Top-N Reports Overview

Top-N reports allows you to collect and analyze data for each physical port on a switch. When Top-N reports start, they obtain statistics from the appropriate hardware counters and then go into sleep mode for a user-specified interval. When the interval ends, the reports obtain the current statistics from the same hardware counters, compare the current statistics from the earlier statistics, and store the difference. The statistics for each port are sorted by one of the statistic types that are listed in Table 70-1.

Table 70-1 Valid Top-N Statistic Types

Statistic Type	Definition
broadcast	Number of input/output broadcast packets
bytes	Number of input/output bytes
errors	Number of input errors
multicast	Number of input/output multicast packets
overflow	Number of buffer overflows
packets	Number of input/output packets
utilization	Utilization

Note When calculating the port utilization, Top-N reports bundles the Tx and Rx lines into the same counter and also looks at the full-duplex bandwidth when calculating the percentage of utilization. For example, a Gigabit Ethernet port would be 2000-Mbps full duplex.

Understanding Top-N Reports Operation

When you enter the **collect top** command, processing begins and the system prompt reappears immediately. When processing completes, the reports are not displayed immediately on the screen; the reports are saved for later viewing. The Top-N reports notify you when the reports are complete by sending a syslog message to the screen.

To view the completed reports, enter the **show top counters interface report** command. Only completed reports are displayed. For reports that are not completed, there is a short description of the process information.

To terminate a Top-N reports process, enter the **clear top counters interface report** command. Pressing **Ctrl-C** does not terminate Top-N reports processes. The completed reports remain available for viewing until you remove them by entering the **clear top counters interface report** {**all** | *report_num*} command.

Using Top-N Reports

These sections describe how to use Top-N reports:

- Enabling Top-N Reports Creation, page 70-3
- Displaying Top-N Reports, page 70-3
- Clearing Top-N Reports, page 70-4

Enabling Top-N Reports Creation

To enable Top-N reports creation, perform this task:

Command	Purpose
Router# **collect top** [*number_of_ports*] **counters interface** {*type*[1] | **all** | **layer-2** | **layer-3**} [**sort-by** *statistic_type*[2]] [**interval** *seconds*]	Enables Top-N reports creation.

1. *type* = **fastethernet**, **gigabitethernet**, **tengigabitethernet**, **port-channel**
2. *statistic_type* = **broadcast**, **bytes**, **errors**, **multicast**, **overflow**, **packets**, **utilization**

When enabling Top-N reports creation, note the following information:

- You can specify the number of busiest ports for which to create reports (the default is 20).
- You can specify the statistic type by which ports are determined to be the busiest (the default is utilization).
- You can specify the interval over which statistics are collected (range: 0 through 999; the default is 30 seconds).
- Except for a utilization report (configured with the **sort-by utilization** keywords), you can specify an interval of zero to create a report that displays the current counter values instead of a report that displays the difference between the start-of-interval counter values and the end-of-interval counter values.

This example shows how to enable Top-N reports creation for an interval of 76 seconds for the four ports with the highest utilization:

```
Router# collect top 4 counters interface all sort-by utilization interval 76
TopN collection started.
```

Displaying Top-N Reports

To display Top-N reports, perform this task:

Command	Purpose
Router# **show top counters interface report** [*report_num*]	Displays Top-N reports. **Note** To display information about all the reports, do not enter a *report_num* value.

Top-N reports statistics are not displayed in these situations:

- If a port is not present during the first poll.
- If a port is not present during the second poll.
- If a port's speed or duplex changes during the polling interval.
- If a port's type changes from Layer 2 to Layer 3 during the polling interval.
- If a port's type changes from Layer 3 to Layer 2 during the polling interval.

This example shows how to display information about all the Top-N reports:

```
Router# show top counters interface report
Id Start Time                Int N   Sort-By   Status  Owner
```

```
-- ---------------------------- --- --- --------- ------- ----------------------
1  08:18:25 UTC Tue Nov 23 2004 76  20  util      done    console
2  08:19:54 UTC Tue Nov 23 2004 76  20  util      done    console
3  08:21:34 UTC Tue Nov 23 2004 76  20  util      done    console
4  08:26:50 UTC Tue Nov 23 2004 90  20  util      done    console
```

Note Reports for which statistics are still being obtained are shown with a status of pending.

This example shows how to display a specific Top-N report:

```
Router# show top counters interface report 1
Started By          : console
Start Time          : 08:18:25 UTC Tue Nov 23 2004
End Time            : 08:19:42 UTC Tue Nov 23 2004
Port Type           : All
Sort By             : util
Interval            : 76 seconds
Port     Band  Util Bytes       Packets     Broadcast  Multicast  In-  Buf-
         width      (Tx + Rx)   (Tx + Rx)   (Tx + Rx)  (Tx + Rx)  err  ovflw
------- ----- ---- ----------- ----------- ---------- ---------- ---- -----
Fa2/5    100   50   726047564   11344488    11344487   1          0    0
Fa2/48   100   35   508018905   7937789     0          43         0    0
Fa2/46   100   25   362860697   5669693     0          43         0    0
Fa2/47   100   22   323852889   4762539     4762495    43         0    0
```

Clearing Top-N Reports

To clear Top-N reports, perform one of these tasks:

Command	Purpose
Router# **clear top counters interface report**	Clears all the Top-N reports that have a status of done.
Router# **clear top counters interface report** [*report_num*]	Clears Top-N report number *report_num* regardless of status.

This example shows how to remove all reports that have a status of done:

```
Router# clear top counters interface report
04:00:06: %TOPN_COUNTERS-5-DELETED: TopN report 1 deleted by the console
04:00:06: %TOPN_COUNTERS-5-DELETED: TopN report 2 deleted by the console
04:00:06: %TOPN_COUNTERS-5-DELETED: TopN report 3 deleted by the console
04:00:06: %TOPN_COUNTERS-5-DELETED: TopN report 4 deleted by the console
```

This example shows how to remove a report number 4:

```
Router# clear top counters interface report 4
04:52:12: %TOPN_COUNTERS-5-KILLED: TopN report 4 killed by the console
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Using the Layer 2 Traceroute Utility

This chapter describes how to use the Layer 2 traceroute utility.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

---

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter contains these sections:

Understanding the Layer 2 Traceroute Utility

The Layer 2 traceroute utility identifies the Layer 2 path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses. The utility determines the path by using the MAC address tables of the switches in the path. When the Layer 2 traceroute utility detects a device in the path that does not support Layer 2 traceroute, it continues to send Layer 2 trace queries and allows them to time out.

The Layer 2 traceroute utility can only identify the path from the source device to the destination device. The utility cannot identify the path that a packet takes from the source host to the source device or from the destination device to the destination host.

Usage Guidelines

When using the Layer 2 traceroute utility, follow these guidelines:

- Cisco Discovery Protocol (CDP) must be enabled on all the devices in the network. For the Layer 2 traceroute utility to function properly, do not disable CDP. If any devices in the Layer 2 path are transparent to CDP, the Layer 2 traceroute utility cannot identify these devices on the path.
- A switch is defined as reachable from another switch when you can test connectivity by using the ping privileged EXEC command. All devices in the Layer 2 path must be mutually reachable. To verify the ping connectivity you need to use the IP address that the CDP advertises on its Layer 2 interfaces.
- The maximum number of hops identified in the path is ten.
- You can enter the **traceroute mac** or the **traceroute mac ip** privileged EXEC command on a switch that is not in the Layer 2 path from the source device to the destination device. All devices in the path must be reachable from this switch.
- The **traceroute mac** command output shows the Layer 2 path only when the specified source and destination MAC addresses belong to the same VLAN. If you specify source and destination MAC addresses that belong to different VLANs, the Layer 2 path is not identified, and an error message appears.
- If you specify a multicast source or destination MAC address, the path is not identified, and an error message appears.
- If the source or destination MAC address belongs to multiple VLANs, you must specify the VLAN to which both the source and destination MAC addresses belong. If the VLAN is not specified, the path is not identified, and an error message appears.
- The **traceroute mac ip** command output shows the Layer 2 path when the specified source and destination IP addresses belong to the same subnet. When you specify the IP addresses, the Layer 2 traceroute utility uses the Address Resolution Protocol (ARP) to associate the IP addresses with the corresponding MAC addresses and the VLAN IDs.
  - If an ARP entry exists for the specified IP address, the Layer 2 traceroute utility uses the associated MAC address and identifies the Layer 2 path.
  - If an ARP entry does not exist, the Layer 2 traceroute utility sends an ARP query and tries to resolve the IP address. If the IP address is not resolved, the path is not identified, and an error message appears.
- When multiple devices are attached to one port through hubs (for example, multiple CDP neighbors are detected on a port), the Layer 2 traceroute utility terminates at that hop and displays an error message.
- The Layer 2 traceroute utility is not supported in Token Ring VLANs.

Using the Layer 2 Traceroute Utility

To display the Layer 2 path that a packet takes from a source device to a destination device, perform one of these tasks in privileged EXEC mode:

Command	Purpose
Router# **traceroute mac** [**interface** *type interface_number*] *source_mac_address* [**interface** *type interface_number*] *destination_mac_address* [**vlan** *vlan_id*] [**detail**]	Uses MAC addresses to trace the path that packets take through the network.
Router# **traceroute mac ip** {*source_ip_address* | *source_hostname*} {*destination_ip_address* | *destination_hostname*} [**detail**]	Uses IP addresses to trace the path that packets take through the network.

These examples show how to use the **traceroute mac** and **traceroute mac ip** commands to display the physical path a packet takes through the network to reach its destination:

```
Router# traceroute mac 0000.0201.0601 0000.0201.0201

Source 0000.0201.0601 found on con6[WS-C2950G-24-EI] (2.2.6.6)
con6 (2.2.6.6) :Fa0/1 => Fa0/3
con5                 (2.2.5.5        )  :    Fa0/3 => Gi0/1
con1                 (2.2.1.1        )  :    Gi0/1 => Gi0/2
con2                 (2.2.2.2        )  :    Gi0/2 => Fa0/1
Destination 0000.0201.0201 found on con2[WS-C3550-24] (2.2.2.2)
Layer 2 trace completed
Router#
Router# traceroute mac 0001.0000.0204 0001.0000.0304 detail
Source 0001.0000.0204 found on VAYU[WS-C6509] (2.1.1.10)
1 VAYU / WS-C6509 / 2.1.1.10 :
                Gi6/1 [full, 1000M] => Po100 [auto, auto]
2 PANI / WS-C6509 / 2.1.1.12 :
                Po100 [auto, auto] => Po110 [auto, auto]
3 BUMI / WS-C6509 / 2.1.1.13 :
                Po110 [auto, auto] => Po120 [auto, auto]
4 AGNI / WS-C6509 / 2.1.1.11 :
                Po120 [auto, auto] => Gi8/12 [full, 1000M] Destination 0001.0000.0304
found on AGNI[WS-C6509] (2.1.1.11) Layer 2 trace completed.
Router#
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Using the Mini Protocol Analyzer

This chapter describes how to use the Mini Protocol Analyzer on the Catalyst 6500 series switches. Release 12.2(33)SXI and later releases support the Mini Protocol Analyzer feature.

Note For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

This chapter consists of these sections:

Understanding How the Mini Protocol Analyzer Works

The Mini Protocol Analyzer captures network traffic from a SPAN session and stores the captured packets in a local memory buffer. Using the provided filtering options, you can limit the captured packets to:

- Packets from selected VLANs, ACLs, or MAC addresses.
- Packets of a specific EtherType.
- Packets of a specified packet size.

You can start and stop the capture using immediate commands, or you can schedule the capture to begin at a specified date and time.

The captured data can be displayed on the console, stored to a local file system, or exported to an external server using normal file transfer protocols. The format of the captured file is libpcap, which is supported by many packet analysis and sniffer programs. Details of this format can be found at the following URL:

http://www.tcpdump.org/

By default, only the first 68 bytes of each packet are captured.

Configuring the Mini Protocol Analyzer

To configure a capture session using the Mini Protocol Analyzer, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)# [`**`no`**`]` **`monitor session`** *`number`* **`type capture`**	Configures a SPAN session number with packets directed to the processor for capture. Enters capture session configuration mode. The session number range is 1 to 80. The **no** prefix removes the session.
Step 3	`Router(config-mon-capture)#` **`buffer-size`** *`buf_size`*	(Optional) Sets the size in KB of the capture buffer. The range is 32-65535 KB; the default is 2048 KB.
Step 4	`Router(config-mon-capture)#` **`description`** *`session_description`*	(Optional) Describes the capture session. The description can be up to 240 characters and cannot contain special characters. If the description contains spaces, it must be enclosed in quotation marks("").
Step 5	`Router(config-mon-capture)#` **`rate-limit`** *`pps`*	(Optional) Sets a limit on the number of packets per second (*pps*) that can be captured. The range is 10-100000 packets per seconds; the default is 10000 packets per second.
Step 6	`Router(config-mon-capture)#` **`source`** `{{`**`interface`** `{`*`single_interface`* | *`interface_list`* | *`interface_range`* | *`mixed_interface_list`*`}` | **`port-channel`** *`channel_id`*`}}` | `{`**`vlan`** `{`*`vlan_ID`* | *`vlan_list`* | *`vlan_range`* | *`mixed_vlan_list`*`}}[`**`rx`** | **`tx`** | **`both`**`]`	Associates the capture session with source ports or VLANs, and selects the traffic direction to be monitored. The default traffic direction is both.
Step 7	`Router(config-mon-capture)#` **`exit`**	Exits the capture session configuration mode.

When configuring a capture session, note the following information:

- Only one capture session is supported; multiple simultaneous capture sessions cannot be configured.

- The **source interface** command argument is either a single interface, or a range of interfaces described by two interface numbers (the lesser one first, separated by a dash), or a comma-separated list of interfaces and ranges.

Note When configuring a source interface list, you must enter a space before and after the comma. When configuring a source interface range, you must enter a space before and after the dash.

- The **source vlan** command argument is either a single VLAN number from 1 through 4094 (except reserved VLANs), or a range of VLANs described by two VLAN numbers (the lesser one first, separated by a dash), or a list of VLANs and ranges.

Note When configuring a source VLAN list, do not enter a space before or after the comma. When configuring a source VLAN range, do not enter a space before or after the dash. Note that this requirement differs from the requirement for source interface lists and ranges.

- Data capture does not begin when the capture session is configured. The capture is started by the **monitor capture start** or **monitor capture schedule** command described in the "Starting and Stopping a Capture" section on page 72-5.
- Although the capture buffer is linear by default, it can be made circular as a run-time option in the **monitor capture start** or **monitor capture schedule** command.
- When no hardware rate limit registers are available, the capture session is disabled.
- When the fabric switching mode is **truncated**, with Release 12.2(33)SXI4 and later releases, you can enter an MPA configuration, but the default rate limiter and, if configured, the MPA rate limiter are not active while the fabric switching mode is **truncated**.

  Note Ignore the **Rate-limiter is not configurable** system message.

  With releases earlier than Release 12.2(33)SXI4, because the **truncated** fabric switching mode does not support the MPA rate limiter, you cannot enter an MPA configuration when the fabric switching mode is **truncated**.
- The source VLAN cannot be changed if a VLAN filter is configured. Remove any VLAN filters before changing the source VLAN.

Filtering the Packets to be Captured

Several options are provided for filtering the packets to be captured. Filtering by ACL and VLAN is performed in hardware before any rate-limiting is applied; all other filters are executed in software. Software filtering can decrease the capture rate.

To filter the packets to be captured by the Mini Protocol Analyzer, perform this task in capture session configuration mode:

	Command	Purpose
Step 1	`Router(config-mon-capture)#` **`filter access-group`** {*acl_number* | *acl_name*}	(Optional) Captures only packets from the specified ACL.
Step 2	`Router(config-mon-capture)#` **`filter vlan`** {*vlan_ID* | *vlan_list* | *vlan_range* | *mixed_vlan_list*}	(Optional) Captures only packets from the specified source VLAN or VLANs.
Step 3	`Router(config-mon-capture)#` **`filter ethertype`** *type*	(Optional) Captures only packets of the specified EtherType. The *type* can be specified in decimal, hex, or octal.
Step 4	`Router(config-mon-capture)#` **`filter length`** *min_len* [*max_len*]	(Optional) Captures only packets whose size is between *min_len* and *max_len,* inclusive. If *max_len* is not specified, only packets of exactly size *min_len* will be captured. The range for *min_len* is 0 to 9216 bytes and the range for *max_len* is 1 to 9216 bytes.
Step 5	`Router(config-mon-capture)#` **`filter mac-address`** *mac_addr*	(Optional) Captures only packets from the specified MAC address.
Step 6	`Router(config-mon-capture)#` **`end`**	Exits the configuration mode.

When configuring capture filtering, note the following information:

- The **filter vlan** argument is either a single VLAN number from 1 through 4094 (except reserved VLANs), or a range of VLANs described by two VLAN numbers (the lesser one first, separated by a dash), or a list of VLANs and ranges.

  Note When configuring a filter VLAN list, you must enter a space before and after the comma. When configuring a filter VLAN range, you must enter a space before and after the dash. Note that this requirement differs from the requirement for source VLAN lists and ranges described in the preceding section.

- To enter an EtherType as a decimal number, enter the number (1 to 65535) with no leading zero. To enter a hexadecimal number, precede four hexadecimal characters with the prefix 0x. To enter an octal number, enter numeric digits (0 to 7) with a leading zero. For example, the 802.1Q EtherType can be entered in decimal notation as 33024, in hexadecimal as 0x8100, or in octal as 0100400.

- Enter a MAC address as three 2-byte values in dotted hexadecimal format. An example is 0123.4567.89ab.
- The **no** keyword removes the filter.

> **Note** After removing a VLAN filter using the **no** keyword, you must exit configuration mode, reenter the capture configuration mode, and issue the **source vlan** command before making other capture configuration changes.

- When you configure a VLAN filter, the capture source or destination must be a VLAN. When you configure a port filter, the capture source or destination must be a port.

Starting and Stopping a Capture

The commands to start and stop a capture are not stored as configuration settings. These commands are executed from the console in EXEC mode. You can start a capture immediately or you can set a future date and time for the capture to start. The capture ends when one of the following conditions occurs:

- A stop or clear command is entered from the console.
- The capture buffer becomes full, unless it is configured as a circular buffer.
- The optionally specified number of seconds has elapsed.
- The optionally specified number of packets has been captured.

When the capture stops, the SPAN session is ended and no further capture session packets are forwarded to the processor.

When starting a packet capture, you have the option to override some configured settings.

To start, stop, or cancel a capture, perform this task:

	Command	Purpose
Step 1	Router# **monitor capture** [**buffer size** *buf_size*][**length** *cap_len*][**linear** | **circular**][**filter** *acl_number* | *acl_name*] {**start** [**for** *count* (**packets** | **seconds**}] | **schedule at** *time date*}	Starts a capture with optional run-time configuration changes. The capture can start immediately or it can start at a specified time and date. • The **buffer size** option overrides the configured or default capture buffer size. • The **length** option determines the number of bytes that will be captured from each packet. The range for *cap_len* is 0 to 9216 bytes; the default is 68 bytes. A value of 0 causes the entire packet to be captured. • The **circular** option specifies that the capture buffer will overwrite earlier entries once it fills. The **linear** option specifies that the capture will stop when the buffer fills. The default is **linear**. • The **filter** option applies the specified ACL. • The **for** option specifies that the capture will end after the specified number of seconds has elapsed or the specified number of packets has been captured.
Step 2	Router# **monitor capture stop**	Stops the capture.
Step 3	Router# **monitor capture clear** [**filter**]	Clears any run-time configuration settings, clears any pending scheduled capture, and clears the capture buffer. The **filter** option clears only the run-time filter settings.

When using these commands, note the following information:

- The format for *time* and *date* is hh:mm:ss dd mmm yyyy. The hour is specified in 24-hour notation, and the month is specified by a three-letter abbreviation. For example, to set a capture starting time of 7:30 pm on October 31, 2006, use the notation 19:30:00 31 oct 2006. The time zone is GMT.
- When you specify a capture filter ACL in the start command, the new ACL will not override any configured ACLs. The new ACL will execute in software.

Displaying and Exporting the Capture Buffer

To display the captured packets or information about the capture session, or to export the captured packets for analysis, perform this task:

	Command	Purpose
Step 1	Router# **show monitor capture**	Displays the capture session configuration.
Step 2	Router# **show monitor capture status**	Displays the capture session state, mode, and packet statistics.
Step 3	Router# **show monitor capture buffer** [*start* [*end*]] [**detail**][**dump** [*nowrap* [*dump_length*]] [**acl** *acl_number* | *acl_name*]]	Displays the capture buffer contents. • The *start* and *end* parameters specify packet number indices in the capture buffer. When a *start* index is specified with no *end* index, only the single packet at the *start* index is displayed. When both the *start* and *end* indices are specified, all packets between these indices are displayed. The range is 1 to 4294967295. • The **detail** option adds expanded and formatted protocol and envelope information for each packet, including the packet arrival time. • The **dump** option displays the hexadecimal contents of the packet. If *nowrap* is specified with *dump_length*, one line of hexadecimal packet content of *dump_length* characters will be displayed for each packet. If *dump_length* is not specified, a line of 72 characters will be displayed. The range of *dump_length* is 14 to 256. • The **acl** option causes the display of only those packets that match the specified ACL.
Step 4	Router# **show monitor capture buffer** [*start* [*end*]] **brief** [**acl** *acl_number* | *acl_name*]	Displays only packet header information.
Step 5	Router# **monitor capture export buffer** *url*	Copies the contents of the capture buffer to the specified file system or file transfer mechanism.

Mini Protocol Analyzer Configuration, Operation, and Display Examples

This section provides examples for configuring the Mini Protocol Analyzer, for starting and stopping a capture session, and for displaying the results of a capture session.

General Configuration Examples

This example shows how to minimally configure the Mini Protocol Analyzer:

```
Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# monitor session 1 type capture
Router(config-mon-capture)# end

Router# show mon cap
Capture instance [1] :
======================
Capture Session ID : 1
Session status     : up
rate-limit value   : 10000
redirect index     : 0x807
buffer-size        : 2097152
capture state      : OFF
capture mode       : Linear
capture length     : 68

Router#
```

This example shows how to configure the buffer size, session description, and rate limit:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# monitor session 1 type capture
Router(config-mon-capture)# buffer-size 4096
Router(config-mon-capture)# description "Capture from ports, no filtering."
Router(config-mon-capture)# rate-limit 20000
Router(config-mon-capture)# end
Router#
Router# show monitor capture
Capture instance [1] :
======================
Capture Session ID : 1
Session status     : up
rate-limit value   : 20000
redirect index     : 0x807
buffer-size        : 4194304
capture state      : OFF
capture mode       : Linear
capture length     : 68

Router#
```

This example shows how to configure the source as a mixed list of ports:

```
Router(config-mon-capture)# source interface gig 3/1 - 3 , gig 3/5
```

This example shows how to configure the source as a mixed list of VLANs:

```
Router(config-mon-capture)# source vlan 123,234-245
```

Filtering Configuration Examples

This example shows how to configure for capturing packets with the following attributes:

- The packets belong to VLANs 123 or 234 through 245
- The packets are of 802.1Q EtherType (hexadecimal 0x8100, decimal 33024)
- The packet size is exactly 8192 bytes
- The source MAC address is 01:23:45:67:89:ab
- The packets conform to ACL number 99

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# monitor session 1 type capture
Router(config-mon-capture)# source vlan 123,234-245
Router(config-mon-capture)# filter ethertype 0x8100
Router(config-mon-capture)# filter length 8192
Router(config-mon-capture)# filter mac-address 0123.4567.89ab
Router(config-mon-capture)# filter access-group 99
Router(config-mon-capture)# end

Router# show monitor capture
Capture instance [1] :
======================
Capture Session ID : 1
Session status     : up
rate-limit value   : 20000
redirect index     : 0x7E07
Capture vlan       : 1019
buffer-size        : 4194304
capture state      : OFF
capture mode       : Linear
capture length     : 68
Sw Filters          :
     ethertype   : 33024
     src mac     : 0123.4567.89ab
  Hw  acl          : 99

Router# show monitor session 1
Session 1
---------
Type                   : Capture Session
Description            : capture from ports
Source VLANs           :
    Both               : 123,234-245
Capture buffer size    : 4096 KB
Capture rate-limit
             value     : 20000
Capture filters        :
     ethertype         : 33024
     src mac           : 0123.4567.89ab
     acl               : 99

Egress SPAN Replication State:
Operational mode       : Centralized
Configured mode        : Distributed (default)
```

```
Router#
```

This example shows how to capture packets whose size is less than 128 bytes:

```
Router(config-mon-capture)# filter length 0 128
```

This example shows how to capture packets whose size is more than 256 bytes:

```
Router(config-mon-capture)# filter length 256 9216
```

Operation Examples

This example shows how to start and stop a capture:

```
Router# monitor capture start
Router# monitor capture stop
Router#
```

This example shows how to start a capture to end after 60 seconds:

```
Router# monitor capture start for 60 seconds
Router#
```

This example shows how to start a capture at a future date and time:

```
Router# monitor capture schedule at 11:22:33 30 jun 2008
capture will start at : <11:22:33 UTC Mon Jun 30 2008>  after 32465825 secs
Router#
```

This example shows how to start a capture with options to override the buffer size and to change to a circular buffer:

```
Router# monitor capture buffer size 65535 circular start
Router#
```

This example shows how to export the capture buffer to an external server and a local disk:

```
Router# monitor capture export buffer tftp://server/user/capture_file.cap
Router# monitor capture export buffer disk1:capture_file.cap
```

Display Examples

These examples show how to display configuration information, session status, and capture buffer contents.

Displaying the Configuration

To display the capture session configuration, enter the **show monitor capture** command.

```
Router# show monitor capture
Capture instance [1] :
======================
Capture Session ID : 1
Session status     : up
rate-limit value   : 10000
redirect index     : 0x807
buffer-size        : 2097152
```

```
capture state      : OFF
capture mode       : Linear
capture length     : 68
```

This example shows how to display more details using the **show monitor session** *n* command:

```
Router# show monitor session 1
Session 1
---------
Type                    : Capture Session
Source Ports            :
    Both                : Gi3/1-3,Gi3/5
Capture buffer size     : 32 KB
Capture filters         : None

Egress SPAN Replication State:
Operational mode        : Centralized
Configured mode         : Distributed (default)
```

This example shows how to display the full details using the **show monitor session** *n* **detail** command:

```
Router# show monitor session 1 detail
Session 1
---------
Type                    : Capture Session
Description             : -
Source Ports            :
    RX Only             : None
    TX Only             : None
    Both                : Gi3/1-3,Gi3/5
Source VLANs            :
    RX Only             : None
    TX Only             : None
    Both                : None
Source RSPAN VLAN       : None
Destination Ports       : None
Filter VLANs            : None
Dest RSPAN VLAN         : None
Source IP Address       : None
Source IP VRF           : None
Source ERSPAN ID        : None
Destination IP Address  : None
Destination IP VRF      : None
Destination ERSPAN ID   : None
Origin IP Address       : None
IP QOS PREC             : 0
IP TTL                  : 255
Capture dst_cpu_id      : 1
Capture vlan            : 0
Capture buffer size     : 32 KB
Capture rate-limit
             value      : 10000
Capture filters         : None

Egress SPAN Replication State:
Operational mode        : Centralized
Configured mode         : Distributed (default)
```

Displaying the Capture Session Status

To display the capture session status, enter the **show monitor capture status** command.

```
Router# show monitor capture status
capture state      : ON
capture mode       : Linear
Number  of packets
          received : 253
           dropped : 0
          captured : 90
```

Displaying the Capture Buffer Contents

To display the capture session contents, enter the **show monitor capture buffer** command. These examples show the resulting display using several options of this command:

```
Router# show monitor capture buffer
 1      IP: s=10.12.0.5 , d=224.0.0.10, len 60
 2      346   0180.c200.000e  0012.44d8.5000  88CC 020707526F7
 3      60    0180.c200.0000  0004.c099.06c5  0026 42420300000
 4      60    ffff.ffff.ffff  0012.44d8.5000  0806 00010800060
 5      IP: s=7.0.84.23 , d=224.0.0.5, len 116
 6      IP: s=10.12.0.1 , d=224.0.0.10, len 60


Router# show monitor capture buffer detail
 1       Arrival time : 09:44:30 UTC Fri Nov 17 2006
         Packet Length : 74 , Capture Length : 68
         Ethernet II :  0100.5e00.000a  0008.a4c8.c038  0800
         IP: s=10.12.0.5 , d=224.0.0.10, len 60, proto=88
 2       Arrival time : 09:44:31 UTC Fri Nov 17 2006
         Packet Length : 346 , Capture Length : 68
346   0180.c200.000e  0012.44d8.5000  88CC 020707526F757463031


Router# show monitor capture buffer dump
 1      IP: s=10.12.0.5 , d=224.0.0.10, len 60
08063810:                       0100 5E00000A            ..^...
08063820: 0008A4C8 C0380800 45C0003C 00000000  ..$H@8..E@.<....
08063830: 0258CD8F 0A0C0005 E000000A 0205EE6A  .XM.....`.....nj
08063840: 00000000 00000000 00000000 00000064  ...............d
08063850: 0001000C 01000100 0000000F 0004      ..............
 2      346   0180.c200.000e  0012.44d8.5000  88CC 020707526F757465720415
 3      60    0180.c200.0000  0004.c099.06c5  0026 4242030000000000800000
 4      60    ffff.ffff.ffff  0012.44d8.5000  0806 0001080006040001001244
 5      IP: s=7.0.84.23 , d=224.0.0.5, len 116
0806FCB0:                       0100 5E000005            ..^...
0806FCC0: 0015C7D7 AC000800 45C00074 00000000  ..GW,...E@.t....
0806FCD0: 01597D55 07005417 E0000005 0201002C  .Y}U..T.`......,
0806FCE0: 04040404 00000000 00000002 00000010  ................
0806FCF0: 455D8A10 FFFF0000 000A1201 0000      E]............


Router# show monitor capture buffer dump nowrap
 1      74    0100.5e00.000a  0008.a4c8.c038  0800 45C0003C000000
 2      346   0180.c200.000e  0012.44d8.5000  88CC 020707526F7574
 3      60    0180.c200.0000  0004.c099.06c5  0026 42420300000000
 4      60    ffff.ffff.ffff  0012.44d8.5000  0806 00010800060400
```

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

PART 12

Appendixes

Configuring Ethernet Services Plus Line Cards

This chapter describes how to configure the features that are supported on Ethernet Services Plus (ES+) line cards. Release 12.2(33)SXJ1 and later releases support ES+ line cards.

Note

- For complete syntax and usage information for the commands used in this chapter, see the Cisco IOS Master Command List, at this URL:

  http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html

- You can configure ES+ line card ports as MPLS core P router ports that carry L2VPN Advanced VPLS (A-VPLS) feature traffic. See Chapter 33, “Configuring A-VPLS.”

This chapter consists of these sections:

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Release 12.2SX ES+ Line Card Support and Restrictions

- Ethernet Services Plus Extended Transport (ES+XT) line cards:

Transceivers:	LAN/WAN PHY, OTN/G.709		
Ports	**Product ID**	**DFC**	**Network Processors (NP)**
4 10GE ports: (Dual fabric connections)	76-ES+XT-4TG3CXL	DFC3CXL	Port 1: NP0 Port 2: NP1 Port 3: NP2 Port 4: NP3
	76-ES+XT-4TG3C	DFC3C	
2 10GE ports: (Single fabric connection)	76-ES+XT-2TG3CXL	DFC3CXL	Port 1: NP0 Port 2: NP1
	76-ES+XT-2TG3C	DFC3C	

- Ethernet Services Plus (ES+) line cards:

Transceivers:	XFP		
Ports	**Product ID**	**DFC**	**Network Processors (NP)**
4 10GE ports: (Dual fabric connections)	7600-ES+4TG3CXL	DFC3CXL	Port 1: NP0 Port 2: NP1 Port 3: NP2 Port 4: NP3
	7600-ES+4TG3C	DFC3C	
2 10GE ports: (Single fabric connection)	7600-ES+2TG3CXL	DFC3CXL	Port 1: NP0 Port 2: NP1
	7600-ES+2TG3C	DFC3C	

- ES+ line cards are supported with Supervisor Engine 720-10GE and Supervisor Engine 720.
- ES+ line cards are not supported with Supervisor Engine 32.
- ES+ line cards are not supported in PFC3A mode.
- ES+ line cards do not support these interface types:
  - Layer 2 access or trunk ports (ports configured with the **switchport** command).
  - Port-channel interfaces (ES+ line card ports cannot be members of EtherChannels).
  - Service instances
- CDP is disabled by default on ES+ line card ports (CSCtk12860).

Line Card Configuration

These sections provide information about configuring ES+ line cards:

- Displaying the ES+ Line Card Type, page A-3
- Resetting an ES+ Line Card, page A-3

Displaying the ES+ Line Card Type

To verify the ES+ line card type, you can use the **show module** command. There are other commands that also provide ES+ line card hardware information, such as the **show idprom** command, and the **show running-config interface** command.

The following example shows output from the **show module** command with an ES+ line card installed in slot 8:

```
Router# show module es_plus_slot
```

The following example shows output from the **show idprom** command for an ES+ line card installed in slot 8:

```
Router# show idprom module es_plus_slot
```

The following example shows sample output from the **show running-config interface** command to verify that the newly created interface appears in the running configuration:

```
Router# show running-config interface es_plus_slot
```

Resetting an ES+ Line Card

To reset an ES+ line card, use the following command in privileged EXEC configuration mode:

Command	Purpose
Router# **hw-module module** *slot* **reset**	Turns power off and on to the ES+ line card in the specified slot.

Configuring QoS

This section provides information about configuring Quality of Service (QoS) on ES+ line cards.

Note QoS on the ES+ line cards uses Layer 2 frame size.

This chapter includes the following sections:

Supported Interfaces

The ES+ line cards support QoS on these interface types:

- Layer 3 interfaces (routed ports)
- Layer 3 subinterfaces
- SVI interfaces

QoS Functions

Ingress QoS Functions

Ingress Trust

Trust is a port assignment instructing the port to trust (leave) existing priorities as they are on incoming frames or to rewrite the priorities back to zero.

A packet can arrive at an interface with a priority value already present in the packets header. The router needs to determine if the priority setting was set by a valid application or network device according to pre defined rules or if it was set by a user hoping to get better service.

The router has to decide whether to honor the priority value or change it to another value. How the router makes this determination is by using the port "trust" setting.

Layer 3 interfaces (routed ports) and the Layer 3 subinterfaces always trust Differentiated Services Code Point (DSCP) by default.

To change the ingress type of service (ToS), use marking. For information on marking, see the "Configuring Marking" section on page A-16.

Note The ES+ line card marks a packet as *trust cos* when ingress marking for CoS is configured for a routed interface. Hence, the CoS value configured using the **set cos** *value* command is retained on the outgoing packet. This cos value is not overwritten by earl or derived from dscp.

Ingress Classification

Classification entails using a traffic descriptor to categorize a packet within a specific group to define that packet and make it accessible for QoS handling on the network. Using packet classification, you can partition network traffic into multiple priority levels or classes of service.

Traffic is classified to determine whether it should be:

- Marked for further processing
- Policed to rate limit specific traffic types

For information on configuring classification, see the "Configuring Classification" section on page A-6.

Ingress Policing

Policing provides a means to limit the amount of bandwidth that traffic traveling through a given port, or a collection of ports in a VLAN, can use. Policing works by defining an amount of data that the router is willing to send or receive in kilobytes per second.

When policing is configured, it limits the flow of data through the router by dropping or marking down the QoS value. Policing allows the router to limit the rate of specific types to a level lower than what they might get otherwise based only the interface bandwidth.

For information on configuring policing, see the "Configuring Policing" section on page A-9.

Ingress Marking

After it has been classified, traffic can be marked. Marking is a way to selectively modify the classification bits in a packet to identify traffic within the network. Other interfaces can then match traffic based on the markings. For information on configuring marking, see the "Configuring Marking" section on page A-16.

Egress QoS Functions

Egress Classification

Classification uses a traffic descriptor to categorize a packet within a specific group to define that packet and make it accessible for QoS handling on the network. Using packet classification, you can partition network traffic into multiple priority levels or classes of service.

Traffic is classified to determine whether it should be:

- Marked for further processing
- Queued to rate limit specific traffic types

For information on configuring classification, see the "Configuring Classification" section on page A-6.

Egress Policing

ES+ line cards support egress port policing.

Egress Marking

After traffic has been classified, the router can mark it. You use marking to selectively modify the classification bits in the packet to differentiate packets based on the designated markings. For information on configuring marking, see the "Configuring Marking" section on page A-16.

Egress Shaping

Traffic shaping allows you to control the traffic going out an interface in order to match its flow to the speed of the remote target interface and to ensure that the traffic conforms to policies contracted for it. You can use shaping to meet downstream requirements, thereby eliminating bottlenecks in topologies with data-rate mismatches. For information on configuring shaping, see the "Configuring Shaping" section on page A-19.

Egress Queue Scheduling

The egress line card uses Class-based Weighted Fair Queuing (CBWFQ) and Weighted Random Early Detection (WRED) congestion avoidance to help prevent congestion and keep its buffers from overflowing. For information on configuring egress scheduling, see the "Configuring Bandwidth and CBWFQ" section on page A-24.

Configuring Classification

Classification Overview

Use the QoS classification features to select your network traffic and categorize it into classes for further QoS processing based on matching certain criteria. The default class, named "class-default," is the class to which any traffic that does not match any of the selection criteria in the configured class maps is directed.

Classification Restrictions and Usage Guidelines

Follow these restrictions and usage guidelines when configuring the QoS classification:

- Classification on ES+ line cards on SVIs is supported only for EoMPLS and VPLS.
- The **match not** command is not supported.

Table A-1 provides information about supported QoS classification features.

Table A-1 QoS Classification Feature Support

Feature (match command)	Supported Interfaces
Match on access list (ACL) number (**match access-group** command)	Input and output for the following interface types: • Layer 3 interfaces (routed ports) • Layer 3 subinterfaces **Note** Deny ACL is not supported on ES+ line cards.
Match on Class of Service (CoS) (**match cos** command)	Input and output for the following interface types: • Layer 3 interfaces (routed ports), to match subinterface traffic and policy-map applied on the Layer 3 interface. • Layer 3 subinterfaces • SVI interfaces only for EoMPLS and VPLS
Match on input VLAN (**match input vlan** command)	Output for Layer 3 interfaces (routed ports), used with non-intelligent line card on the input side and an ES+ line card on the output side. The service policy is applied on the output side to match the VLAN from the input side.
Match on IP DSCP (**match ip dscp** command)	Input and output for the following interface types: • Layer 3 interfaces (routed ports) • Layer 3 subinterfaces
Match on IP precedence (**match ip precedence** command)	Input and output for the following interface types: • Layer 3 interfaces (routed ports) • Layer 3 subinterfaces
Match on MPLS experimental (EXP) bit (**match mpls experimental** command)	Input and output for the following interface types: • Layer 3 interfaces (routed ports) • Layer 3 subinterfaces
Match on VLAN (**match vlan** command) **Note** Matches the outer VLAN of a Layer 2 IEEE 802.1Q frame.	Input and output for the following interface types: • Layer 3 interfaces (routed ports) for classification on SVIs only for EoMPLS and VPLS • Layer 3 subinterfaces

	Command	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# class-map [match-all | match-any] class-map-name`	Creates a traffic class, where: • **match-all**—(Optional) Specifies that all match criteria in the class map must be matched, using a logical AND of all matching statements defined under the class. This is the default. • **match-any**—(Optional) Specifies that one or more match criteria must match, using a logical OR of all matching statements defined under the class. • *class-map-name*—Specifies the user-defined name of the class. **Note** You can define up to 1000 unique class maps.
Step 4	`Router(config-cmap)# match type`	Specifies the matching criterion to be applied to the traffic, where *type* represents one of the supported **match** commands shown in Table A-1. **Note** A single class map can contain up to 8 different **match** command statements.

This example shows how to configure a class map named ipp5, and enter a match statement for IP precedence 5:

```
Router> enable
Router# configure terminal
Router(config)# class-map ipp5
Router(config-cmap)# match ip precedence 5
```

This is an example of configuring class matching on multiple match statements.

```
Router> enable
Router# configure terminal
Router(config)# class-map match-any many (id 1047)
Router(config-cmap)# match ip precedence 3
Router(config-cmap)# match access-group 100
Router(config-cmap)# match mpls experimental 5
```

This is an example of configuring class matching on named ACLS.

```
Router> enable
Router# configure terminal
Router(config)# class-map match-all acl9 (id 1049)
Router(config-cmap)# match access-group name rock
```

This example shows how to display class-map information for a specific class map using the **show class-map** command:

```
Router# show class-map ipp5
    class map match-all ipp5 (id 1)
        match ip precedence 5
```

This example shows how to display class map information matching on extended ACLs using the **show class-map** command.

```
Router# show class-map acl5
    class map match-all acl5 (id 1042)
        match access-group 102
```

This example shows how to verify classification on a VLAN in the parent class of a H-QoS policy.

```
Router# show policy-map match
    policy map match
        class vlan11
           shape average 2000000 8000 8000
           service-policy match4
        class vlan12
           shape average 2000000 8000 8000
           service-policy match4
        class vlans
           shape average 500000000 2000000 2000000
           service-policy match2
```

Configuring Policing

Policing Overview

The ES+ line cards support the following QoS features:

- Individual actions
- Multiple actions
- Single rate, 2-color and 3-color policers
  - Granularity
  - Accuracy (rate and bucket depths)
  - Statistics
  - Percent based policer
- Dual Rate, 3 color, percent based policer
- Color blind mode (color aware policer not supported)
- Hierarchical policies (up to two levels)
- 255 profiles at different rates

Policing is supported at the input and output for the following interface typess:

- Layer 3 interfaces (routed ports)
- Layer 3 subinterfaces

Policing Restrictions and Usage Guidelines

When configuring policing, follow these restrictions and usage guidelines:

- The ES+ line cards supports a maximum of 1,024 unique global policy-maps per line card.
- Maximum class maps per policy-map are 255.
- Policer CIR and PIR can be any value between 64,000 bps to 10 Gb/s.
- If a service policy configures both class-based marking and marking as part of a policing action, then the marking using policing takes precedence over any class-based marking.
- When configuring policing paired with priority actions:
  - If there are some other bandwidth classes configured in the policy-map, then either **exceed** or **violate** action must be dropped. The **conform** action can be any action.
  - If no other bandwidth class is configured, then **conform**, **exceed**, and **violate** can be any action.
- Up to 48,000 policers per NP are supported for one rate 2 color or two rate 3 color policers.

Table A-2 provides information about which policing features are supported for the ES+ line cards.

Table A-2 QoS Policing Feature Support

Policing Command	Policing Action (set command)
police *bps value* **conform-action** *action* **exceed-action** *action*	• Transmit the packet (**transmit** action) • Drop the packet (**drop** command) • Set the IP precedence value (**set ip precedence** command) • Set the IP DSCP value (**set ip dscp** command) • Set the MPLS EXP bit (0–7) on imposition (**set-mpls-experimental-imposition** command) • Set the MPLS EXP bit in the topmost label (**set-mpls-experimental-topmost** command) • Set the COS value (**set cos** command)
police cir percent *%* **conform-action** *action* **exceed-action** *action*	• Transmit the packet (**transmit** action) • Drop the packet (**drop** command) • Set the IP precedence value (**set ip precedence** command) • Set the IP DSCP value (**set ip dscp** command) • Set the MPLS EXP bit (0–7) on imposition (**set-mpls-experimental-imposition** command) • Set the MPLS EXP bit in the topmost label (**set-mpls-experimental-topmost** command) • Set the COS value (**set cos** command) • Set the COS-inner value (**set cos-inner** command)

Table A-2 QoS Policing Feature Support (continued)

Policing Command	Policing Action (set command)
`police cir` *bps value* `pir` *bps value* `conform-action` *action* `exceed-action` *action* `violate-action` *action*	• Transmit the packet (**transmit** action) • Drop the packet (**drop** command) • Set the IP precedence value (**set ip precedence** command) • Set the IP DSCP value (**set ip dscp** command) • Set the MPLS EXP bit (0–7) on imposition (**set-mpls-experimental-imposition** command) • Set the MPLS EXP bit in the topmost label (**set-mpls-experimental-topmost** command) • Set the CoS value (**set cos** command) • Set the CoS-inner value (**set cos-inner** command)
`police cir percent` *%* `pir percent` *%* `conform-action` *action* `exceed-action` *action* `violate-action` *action*	• Transmit the packet (**transmit** action) • Drop the packet (**drop** command) • Set the IP precedence value (**set ip precedence** command) • Set the IP DSCP value (**set ip dscp** command) • Set the MPLS EXP bit (0–7) on imposition (**set-mpls-experimental-imposition** command) • Set the MPLS EXP bit in the topmost label (**set-mpls-experimental-topmost** command) • Set the COS value (**set cos** command) • Set the COS-inner value (**set cos-inner** command)

Configuring Policy Maps, Class Maps, and Policing

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`policy-map`** *`policy-map-name`*	Creates or modifies a traffic policy and enters policy-map configuration mode, where *policy-map-name* specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.

	Command	Purpose
Step 4	`Router (config-pmap)# class {class-name | class-default}`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode, where: • *class-name*—Specifies that the policy applies to a user-defined class name previously configured. • **class-default**—Specifies that the policy applies to the default traffic class.
Step 5	`Router(config-pmap-c)# police bps-value conform-action action exceed-action action`	Specifies a maximum bandwidth usage by a traffic class through the use of a token bucket algorithm, where: • *bps value*—Specifies the average rate in bits per second. Valid values are 16000 to 10Gb/s. • *action*—Specifies the actions that are taken on a packet when it conforms or exceeds. The possible actions are shown in Table A-2.
Or		
	`Router(config-pmap-c)# police cir percent percentage conform-action action exceed-action action`	Configures traffic policing on the basis of a percentage of bandwidth available on an interface, where: • **cir**—Specifies the committed information rate. Indicates that the committed information rate (CIR) will be used for policing traffic. • **percent**—Specifies that a percentage of bandwidth will be used for calculating the CIR. • *percentage*—Specifies the CIR bandwidth percentage. Valid values are 1 to 100. • *action*—Specifies the he actions that are taken on a packet when it conforms or exceeds. The possible actions are shown in Table A-2.
Or		
	`Router(config-pmap-c)# police cir bps-value pir bps-value conform-action action exceed-action action violate-action action`	Configures traffic policing using two rates, the CIR and the peak information rate (PIR), where: • **cir**—Specifies the committed information rate. Indicates that the CIR will be used for policing traffic. • **pir**—Specifies the peak information rate. Indicates that the PIR will be used for policing traffic. • *bps-value*—Specifies the average rate in bits per second. Valid values are 64000 to 200000000. • *action*—Specifies the he actions that are taken on a packet when it conforms or exceeds. The possible actions are shown in Table A-2.

Command	Purpose
Or	
Router(config-pmap-c)# **police cir percent** *percentage* **pir percent** *percentage* **conform-action** *action* **exceed-action** *action* **violate-action** *action*	Configures traffic policing using two rates, the CIR and the PIR, where: • **cir**—Specifies the committed information rate. Indicates that the CIR will be used for policing traffic. • **percent**—Specifies that a percentage of bandwidth will be used for calculating the CIR. • *percentage*—Specifies the CIR or PIR bandwidth percentage. Valid values are 1 to 100. • **pir**—Specifies the peak information rate. Indicates that the PIR will be used for policing traffic. • *action*—Specifies the he actions that are taken on a packet when it conforms or exceeds. The possible actions are shown in Table A-2.

In the following example, all actions are configured in separate lines.

```
Router# (config)# policy-map ABC
Router(config-pmap)# class class-default
Router(config-pmap-c)# police 10000000 8000 8000
Router(config-pmap-c-police)# conform-action set-cos-transmit 2
Router(config-pmap-c-police)# exceed-action set-cos-transmit 1
Router(config-pmap-c-police)# end
Router# show policy-map ABC
    Policy Map ABC
        Class class-default
            police cir 10000000 bc 8000 be 8000
                conform-action set-cos-transmit 2
                exceed-action set-cos-transmit 1
```

This example configures a 1 rate 2-color policer:

```
Router(config)# policy-map 1r2c
Router(config-pmap)# class class-default
Router(config-pmap-c)# police 2000000
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
Router(config-pmap-c-police)# end
Router# show policy-map 1r2c
    Policy Map 1r2c
        Class class-default
            police cir 2000000 bc 62500
                conform-action transmit
                exceed-action drop
```

This example configures a 1 rate 2-color policer with percent:

```
Router(config)# policy-map 1r2c_percent
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir percent 20
Router(config-pmap-c-police)# conform-action set-cos-transmit 0
Router(config-pmap-c-police)# exceed-action drop
Router(config-pmap-c-police)# end
```

```
Router# show policy-map 1r2c_percent
    Policy Map 1r2c_percent
        Class class-default
            police cir percent 20
                conform-action set-cos-transmit 0
                exceed-action drop
```

This example configures a 2 rate 3-color policer:

```
Router(config)# policy-map 2r3c
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir 2000000 pir 3000000
Router(config-pmap-c-police)# conform-action set-prec-transmit 3
Router(config-pmap-c-police)# exceed-action set-prec-transmit 2
Router(config-pmap-c-police)# violate-action set-prec-transmit 1
Router(config-pmap-c-police)# end
Router# show policy-map 2r3c
    Policy Map 2r3c
        Class class-default
            police cir 2000000 bc 62500 pir 3000000 be 93750
                conform-action set-prec-transmit 3
                exceed-action set-prec-transmit 2
                violate-action set-prec-transmit 1
```

This example configures a 2 rate 3-color policer with percent:

```
Router(config)# policy-map 2r3c_percent
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir percent 10 pir percent 20
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action set-cos-transmit 0
Router(config-pmap-c-police)# violate-action drop
Router(config-pmap-c-police)# end
Router# show policy-map 2r3c_percent
    Policy Map 2r3c_percent
        Class class-default
            police cir percent 10 pir percent 20
                conform-action transmit
                exceed-action set-cos-transmit 0
                violate-action drop
```

This example configures a single rate two color policer in class-default with a CIR of 64 Kbps, a conform action of transmit and an exceed action of drop with as small a Bc as possible:

```
Router> enable
Router# configure terminal
Router(config)# policy-map police
Router(config-pmap)# class test8
Router(config-pmap-c)# police 64000 2000
```

This example configures a single rate two color policer in class-default and a child policy with policing:

```
Router> enable
Router# configure terminal
Router(config)# policy-map police5
Router(config-pmap)# class test18
Router(config-pmap-c)# service policy child-level
Router(config-pmap-c)# police cir 64000 50
```

The following example shows a 2R3C configuration in a class and policy-map:

```
Router> enable
Router# configure terminal
Router(config)# policy-map test
Router(config-pmap)# class cos2
Router(config-pmap-c)# police 1000000 pir 2000000 conform-action set-cos-transmit 3
exceed-action set-cos-transmit 1 violate-action drop
```

The following example configures a dual rate three color policer in class-default with a CIR of 64 Kbps, and PIR doubled the CIR rate, a conform action of transmit, and an exceed action mark dscp af11 and violate mark dscp cs1 with default setting on Bc.

```
Router> enable
Router# configure terminal
Router(config)# policy-map qos_test
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir 64000 bc 2000 pir 128000 be 2000 conform-action transmit
exceed-action set-dscp-transmit af11 violate-action set-dscp-transmit cs1
```

The following example configures a dual rate three color policer in class-default.

```
Router> enable
Router# configure terminal
Router(config)# policy-map test
Router(config-pmap)# class class-default
Router(config-pmap-c)# police cir percent 20 pir percent 40 conform-action transmit
exceed-action set-prec-transmit 1 violate-action drop
```

Use the following commands to verify policing:

Command	Purpose
Router# **show policy-map**	Displays all configured policy-maps.
Router# **show policy-map** *policy-map-name*	Displays the user-specified policy-map.
Router# **show policy-map interface**	Displays statistics and configurations of all input and output policies that are attached to an interface.

This example shows how to display policing statistics using the **show policy-map interface** command in the EXEC mode.

```
Router# show policy-map interface
TenGigabitEthernet3/1
service-policy output: x
class-map: a (match-all)
0 packets, 0 bytes
5 minute rate 0 bps
match: ip precedence 0
police:
1000000 bps, 10000 limit, 10000 extended limit
conformed 0 packets, 0 bytes; action: transmit
exceeded 0 packets, 0 bytes; action: drop
    conformed 0 bps, exceed 0 bps, violate 0 bps
```

Attaching a QoS Traffic Policy to an Interface

Before a traffic policy can be enabled for a class of traffic, it must be configured on an interface. A traffic policy also can be attached to Ethernet subinterfaces and main interfaces. Traffic policies can be applied for traffic coming into an interface (input), and for traffic leaving that interface (output).

- Attaching a QoS Traffic Policy for an Input Interface, page A-16
- Attaching a QoS Traffic Policy to an Output Interface, page A-16

Attaching a QoS Traffic Policy for an Input Interface

When you attach a traffic policy to an input interface, the policy is applied to traffic coming into that interface. To attach a traffic policy for an input interface, use the following command beginning in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`service-policy input`** *`policy-map-name`*	Attaches a traffic policy to the input direction of an interface, where *policy-map-name* specifies the name of the traffic policy to configure.

Attaching a QoS Traffic Policy to an Output Interface

When you attach a traffic policy to an output interface, the policy is applied to traffic leaving that interface. To attach a traffic policy to an output interface, use the following command beginning in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`service-policy output`** *`policy-map-name`*	Attaches a traffic policy to the output direction of an interface, where *policy-map-name* specifies the name of the traffic policy to configure.

Configuring Marking

- Marking Overview, page A-16
- Marking Restrictions and Usage Guidelines, page A-17
- Configuring Policy Maps, Class Maps, and Marking, page A-18

Marking Overview

After you have created your traffic classes, you can configure traffic policies to configure marking features to apply certain actions to the selected traffic in those classes.

In most cases, the purpose of a packet mark is identification. After a packet is marked, downstream devices identify traffic based on the marking and categorize the traffic according to network needs. This categorization occurs when the **match** commands in the traffic class are configured to identify the packets by the mark (for example, **match ip precedence**, **match ip dscp**, **match cos**, and so on). The traffic policy using this traffic class can then set the appropriate QoS features for the marked traffic.

In some cases, the markings can be used for purposes besides identification. Distributed WRED, for instance, can use the IP precedence, IP DSCP, or MPLS EXP values to detect and drop packets.

Marking Restrictions and Usage Guidelines

When configuring class-based marking on an ES+ line card, follow these restrictions and usage guidelines:

- There is no limit on the number of marking statements per class map.
- Marking can be configured at the parent.
- EARL marking is not used.
- Marking can be combined with queueing policies.
- Marking statistics are not provided in **show policy-map interface** command output. You can refer to classification statistics in place of marking statistics.

Table A-3 provides information about supported QoS class-based marking features.

Table A-3 QoS Class-Based Marking Feature Support

Marking Feature (set command)	Supported Interfaces
Set IP DSCP (**set ip dscp** command—Marks the IP differentiated services code point (DSCP) in the type of service (ToS) byte with a value from 0 to 63.)	Input and output for these interface types: • Layer 3 interfaces (routed port) • Layer 3 subinterfaces
Set IP precedence (**set ip precedence** command—Marks the precedence value in the IP header with a value from 0 to 7.)	Input and output for these interface types: • Layer 3 interfaces (routed port) • Layer 3 subinterfaces
Set Layer 2 IEEE 802.1Q CoS (**set cos** command—Marks the CoS value from 0 to 7 in an 802.1Q tagged frame.)	Input and output for these interface types: • Layer 3 interfaces (routed ports) to match subinterface traffic and policy-map applied on the Layer 3 interface. • Layer 3 subinterface
Set MPLS experimental (EXP) bit on label imposition (**set mpls experimental imposition** command)	Input for these interface types: • Layer 3 interfaces (routed port) • Layer 3 subinterfaces
Set MPLS EXP topmost (**set mpls experimental topmost** command)	Input and output for these interface types: • Layer 3 interfaces (routed port) • Layer 3 subinterfaces

Configuring Policy Maps, Class Maps, and Marking

	Command	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# policy-map policy-map-name`	Creates or modifies a traffic policy and enters policy-map configuration mode, where *policy-map-name* specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
Step 4	`Router(config-pmap)# class {class-name | class-default}`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode, where: • *class-name*—Specifies that the policy applies to a user-defined class name previously configured. • **class-default**—Specifies that the policy applies to the default traffic class.
Step 5	`Router(config-pmap-c)# set type`	Specifies the marking action to be applied to the traffic, where *type* represents one of the forms of the **set** supported commands as shown in Table A-3.

This example shows the creation of a service policy called policy1. This service policy is associated to a previously defined classification policy through the use of the **class** command. This example assumes that a classification policy called class1 was previously configured.

```
Router> enable
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# set ip precedence 1
```

This example configures marking to set the imposed MPLS EXP bits to 1:

```
Router> enable
Router# configure terminal
Router(config)# policy-map test
Router(config-pmap)# class test
Router(config-pmap-c)# set mpls experimental imposition 1
```

This example configures marking to set the inner cos value:

```
Router> enable
Router# configure terminal
Router(config)# policy-map test
Router(config-pmap)# class test
Router(config-pmap-c)# set cos inner 1
```

This example configures marking to set the imposed MPLS EXP bits to 1:

```
Router> enable
Router# configure terminal
Router(config)# policy-map test
Router(config-pmap)# class test
Router(config-pmap-c)# set mpls experimental topmost 1
```

Use the following commands to verify marking:

Command	Purpose
`Router# show policy-map`	Displays all configured policy-maps.
`Router# show policy-map policy-map-name`	Displays the user-specified policy-map.
`Router# show policy-map interface`	Displays statistics and configurations of all input and output policies that are attached to an interface.

Configuring Shaping

- Shaping Restrictions and Usage Guidelines, page A-19
- Configuring Class Maps, Policy Maps and Shaping, page A-20

Shaping Restrictions and Usage Guidelines

When configuring shaping, follow these restrictions and usage guidelines:

- Up to 256 shaping profiles are supported at the parent level and 64 at the child level and flat policy.
- Shaping can be performed at all levels of the hierarchy.
- Shaping rates range from 64 Kbps to link rate.
- Dual shapers are not supported.
- Main interface supports two-level policy-maps:
  - Parent user defined classes
  - Child user defined classes
- Shaper CIR granularity for child level shaper:
  - 64,000 bps to 32,768,000 bps: granularity of 16,000 bps
  - 32,768,000 bps to 131,008,000 bps: granularity of 64,000 bps
- Shaper CIR granularity for parent level shaper:
  - Can be any value between 64,000 bps to 10 Gb/s
- Maximum shaper rate in the leaf policy-map is 130 Mb/s.
- The **shape average percent** command is not supported.

Table A-4 lists the supported QoS traffic shaping features.

Table A-4 QoS Traffic Shaping Feature Support

Traffic Shaping Feature (command)	Purpose
Class-based shaping (**shape average** commands)	Input and output for these interface types: • Layer 3 interfaces (routed ports) • Layer 3 subinterfaces

Configuring Class Maps, Policy Maps and Shaping

	Command	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# class-map [match-all	match-any] class-map-name`	Creates a class map to be used for matching packets to a class.
Step 4	`Router(config-cmap)# match [ip dscp ip-dscp-value	ip precedence ip-precedence-value	mpls experimental mpls-exp-value]`	Specifies a specific IP DSCP, IP precedence, or MPLS EXP value as a match criterion.
Step 5	`Router(config)# policy-map policy-name`	Specifies the name of the policy-map to configure.
Step 6	`Router(config-pmap)# class class-name`	Specifies the name of a predefined class included in the service policy.
Step 7	`Router(config-pmap-c)# shape average cir [bc] [be]`	Specifies the average rate traffic shaping.

This example shows traffic shaping on a main interface; traffic leaving interface tengi1/1 is shaped at the rate of 10 Mb/s:

```
Router> enable
Router# configure terminal
Router(config)# class-map class-interface-all
Router(config-cmap)# match ip precedence 2
Router(config-cmap)# exit
Router(config)# policy-map dts-interface-all-action
Router(config-pmap)# class class-interface-all
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# exit
Router(config)# interface tengi1/1
Router(config-if)# service-policy output dts-interface-all-action
```

In this example, shape is applied at the parent level of an HQoS policy-map.

```
Router> enable
Router# configure terminal
Router(config)# policy-map child2
Router(config-pmap)# class prec5
Router(config-pmap-c)# shape average 100000000
Router(config)# policy-map pcd
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 300000000
Router(config-if)# service-policy child2
```

This example configures a shaping policy in default-class with WRED:

```
Router> enable
Router# configure terminal
Router(config)# policy map qos_test
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape ave 100mbps
Router(config-pmap-c)# random-detect dscp-based aggregate
```

Use the following commands to verify traffic shaping:

Command	Purpose
Router# **show interface** [*interface-name*]	Displays detail status of the traffic shaping.
Router# **show policy** *policy-name*	Displays the configuration of all classes composing the specified traffic policy.
Router# **show policy** *policy-name* **class** *class-name*	Displays the configuration of the specified class of the specified traffic policy.

Configuring QoS Queue Scheduling

- QoS Queue Scheduling Restrictions and Usage Guidelines, page A-21
- Configuring WRED, page A-22
- Configuring Bandwidth and CBWFQ, page A-24

QoS Queue Scheduling Restrictions and Usage Guidelines

When configuring queueing features, follow these restrictions and usage guidelines:

- The number of data queues configurable per policy-map at child level depends on the priority queue configuration:
  - If there are no priority queue configured, each subscriber can have up to 8 normal queues.
  - If there is any priority queue of any priority level configured, each subscriber can have 2 priority queues and up to 6 normal queues.
  - If there is only 1 priority queue configured, the other priority queue is reserved and cannot be used as a normal queue.
- 4k parent queues for ingress and 8k parent queues for egress per NP (nonconfigurable).
- 32K child queues on ingress and 64k child queues for egress per NP (nonconfigurable).
- Parent class-default on sub-interface scales more.
- Parent user-defined classmap is supported on Layer 3 interface (routed port; output only).
- QoS queue scheduling supports the following commands:
  - **bandwidth** *x kbps*
  - **bandwidth percent** *x%*
  - **bandwidth remaining percent** *x %*
  - **queue-limit** *queue-size*
  - **queue-limit** *queue-size* **packets**

- **random-detect**
- **random-detect** *min-threshold max-threshold mark-prob*
- **random-detect dscp-based aggregate**
- **random-detect dsc**p *0-63 min-threshold max-threshold mark-prob*
- **random-detect prec-based**
- **random-detect precedence** *0-7 min-threshold max-threshold mark-prob*

Configuring WRED

WRED Overview

Weighted RED (WRED) generally drops packets selectively based on IP precedence. Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence. WRED is supported on the output of these interfaces:

- Layer 3 interfaces (routed ports)
- Layer 3 subinterfaces

WRED Aggregate and Non-Aggregate Mode

WRED Aggregate mode and Non-Aggregate modes define how the hardware resources are internally used to provide the WRED behavior. There are 8 WRED curves. In a WRED Non-Aggregate mode, a single or Prec value maps to one WRED curve and in a WRED Aggregate mode, multiple dscp values are mapped to one WRED curve.

The set of subclass (DSCP precedence) values defined on a random-detect dscp (aggregate) CLI is aggregated into a single hardware WRED resource. The statistics for these subclasses are also aggregated.

WRED Restrictions and Usage Guidelines

When configuring WRED, follow these restrictions and usage guidelines:

- WRED support is precedence-based, DSCP-based, and CoS-based. The default with the **random-detect** command is precedence-based WRED.
  - DSCP-based is supported only in aggregate mode, as dscp takes 64 possible values, and maps multiple DSCP values to each of the 8 WRED curves. Example: DSCP 30, 50, 60 takes WRED Curve1, DSCP 10, 40 takes WRED Curve2.
  - CoS is supported only in non-aggregate mode, as CoS takes eight possible values, and maps single value to each of the 8 WRED curves.
  - IP-prec is supported in both aggregate and non-aggregate mode.

- The support per interface is as follows:
  - For subinterfaces, WRED supports dscp and prec based only.
  - Queue limit is not supported with WRED command.
- WRED is not supported in parent classes.
- WRED is not supported for priority queues of all priority levels.
- Random Detect in class queue needs a queueing feature.
- Random Detect in default class does not need a queueing feature.
- ES+ line cards do not support discard-class-based, ECN, and WRED.
- ES+ line cards support aggregate WRED.
- Supports 8 curves per queue
- The **show policymap interface** command for WRED does not display transmitted packet and tail drop counts. Only random drops are displayed.
- The maximum threshold value must be between 16 and 1000000.
- EXP-based WRED for MPLS packets is supported.

Configuring Policy Maps, Class Maps, and WRED

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`policy-map`** *`policy-name`*	Specifies the name of the policy-map to configure.
Step 4	`Router(config-pmap)#` **`class`** *`class-name`*	Specifies the name of a predefined class included in the service policy.
Step 5	`Router(config-pmap-c)#` **`shape average cir`** `[`*`bc`*`] [`*`be`*`]`	Shapes traffic to the indicated bit rate for the specified class.
Step 6	`Router(config-pmap-c)#` **`random-detect`**	Enables WRED.

This is an example of a WRED configuration.

```
Router> enable
Router# configure terminal
Router(config)# policy-map wredtest
Router(config-pmap)# class cos5
Router(config-pmap-c)# shape average 200000000
Router(config-pmap-c)# random-detect dscp-based aggregate
Router(config-pmap-c)# random-detect dscp values 0 min 100 max 200 mark-prob 1
Router(config-pmap-c)# random-detect dscp values 1 min 300 max 500 mark-prob 1
Router(config-pmap-c)# random-detect dscp values 2 min 600 max 900 mark-prob 1
```

The following example configures a class-map which matches IPP=1, 3, 5 and 7, and configures a WRED policy that is applied to the egress interface:

```
Router> enable
Router# configure terminal
Router(config)# policy-map wred
Router(config-pmap)# class IPP1
Router(config-pmap-c)# shape average 100000000
```

```
Router(config-pmap-c)# random-detect precedence-based
Router(config-pmap)# class IPP3
Router(config-pmap-c)# shape average 100000000
Router(config-pmap-c)# random-detect precedence-based
Router(config-pmap)# class IPP5
Router(config-pmap-c)# shape average 100000000
Router(config-pmap-c)# random-detect precedence-based
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 100000000
Router(config-pmap-c)# random-detect precedence-based
```

The following example show the output of the **show policy-map** interface command (transmit packets are not displayed).

```
Router> enable
Router# configure terminal
Router# show policy-map interface tengig 11/1
TenGigabitEthernet11/1:

Service-policy output: temp_parent

Class-map: class-default (match-any)
139358 packets, 71351296 bytes
5 minute offered rate 1745000 bps, drop rate 283000 bps
Match: any
        0 packets, 0 bytes
        5 minute rate 0 bps
Queueing
queue limit 2048 packets
(queue depth/total drops/no-buffer drops) 0/104062/0
(pkts output/bytes output) 35296/18071552
shape (average) cir 10000000, bc 40000, be 40000
target shape rate 10000000

Service-policy : temp

Counters last updated 00:00:00 ago

Class-map: class-default (match-any)
139358 packets, 71351296 bytes
5 minute offered rate 1745000 bps, drop rate 1304000 bps
Match: any
        0 packets, 0 bytes
        5 minute rate 0 bps
queue limit 2048 packets
(queue depth/total drops/no-buffer drops) 0/104062/0
(pkts output/bytes output) 35296/18071552
Exp-weight-constant: 9 (1/512)
Mean queue depth: 0 packets
class Random drop Tail drop Minimum Maximum Mark
pkts/bytes pkts/bytes thresh thresh prob
```

Configuring Bandwidth and CBWFQ

- Bandwidth and CBWFQ Overview, page A-25
- Bandwidth and CBWFQ Restrictions and Usage Guidelines, page A-25
- Configuring Policy Maps, Class Maps, and Bandwidth, page A-25

Bandwidth and CBWFQ Overview

Class-based weighted fair queueing (CBWFQ) extends the standard WFQ functionality to provide support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria and access control lists (ACLs).

Bandwidth is supported on the output of these interface types:

- Layer 3 interfaces (routed ports)
- Layer 3 subinterfaces

WFQ is a method to determine bandwidth or allocating remaining bandwidth to queueing entities at a specific level in the hierarchical QoS. You can distribute bandwidth or remaining bandwidth to each entity based on the commit and excess weights set on the WFQ configuration attached to the entity. The commit and excess WFQ weights are initially programmed into WFQ profile registers, where later the WFQ profiles are attached to one or more queuing entities based on whether or not they share the same or similar bandwidth configuration.

The layer 3 and layer 4 level WFQ profiles belong to one hardware pool and can be commonly used among the layers.

Bandwidth and CBWFQ Restrictions and Usage Guidelines

When configuring Bandwidth and CBWFQ, follow these restrictions and usage guidelines:

- The **bandwidth** *kbps* and **bandwidth percent** *x%* commands are supported.
- The **bandwidth remaining percent** command is supported at the child level. The **bandwidth remaining ratio** command is supported at the parent and child level.
- The **bandwidth** command used within a QoS policy-map must be consistent across classes.

Configuring Policy Maps, Class Maps, and Bandwidth

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`policy-map`** *`policy-map-name`*	Creates or modifies a traffic policy and enters policy-map configuration mode, where *policy-map-name* specifies the name of the traffic policy to configure. Names can be a maximum of 40 alphanumeric characters.
Step 4	`Router(config)#` **`class`** `{`*`class-name`* `|` **`class-default`**`}`	Specifies the name of the traffic class to which this policy applies and enters policy-map class configuration mode, where: • *class-name*—Specifies that the policy applies to a user-defined class name previously configured. • **class-default**—Specifies that the policy applies to the default traffic class.
Step 5	`Router(config-pmap-c)#` **`bandwidth`** `{`**`bandwidth-kbps`** `|` **`percent percent`** `|` **`percent`** *`percent`*`}}`	Specifies the amount of bandwidth, in kbps, or percentage of available bandwidth, to be assigned to the class. The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.

This example shows a service policy called policy1 that specifies the amount of bandwidth to allocate for traffic classes 1 and 2:

```
Router> enable
Router# configure terminal
Router(config)# class-map class1
Router(config-cmap)# match ip dscp 30
Router(config-cmap)# exit
Router(config)# class-map class2
Router(config-cmap)# match ip dscp 10
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 30000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 20000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface tengigabitethernet 2/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

The following example configures a QoS policy with multiple user class with rate guarantee setting using the **bandwidth** command.

```
Router(config)# policy-map policy1
Router(config)# class c1
Router(config-pmap-c)# bandwidth percent 1%
Router(config-pmap)# class c2
Router(config-pmap-c)# bandwidth percent 10%
Router(config-pmap)# class c3
Router(config-pmap-c)# bandwidth percent 88%
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth 1%
```

The following example configures a QoS policy with multiple user class with rate guarantee setting:

```
Router> enable
Router# configure terminal
Router(config)# policy-map child_policy
Router(config-pmap)# class video
Router(config-pmap-c)# police 10000000
Router(config-pmap)# class critical
Router(config-pmap-c)# bandwidth remaining percent 80
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining percent 20
```

Use the following commands to verify CBWFQ:

Command	Purpose
`Router# show policy-map policy-map`	Displays the configuration of all classes that make up the specified policy-map.
`Router# show policy-map policy-map class class-name`	Displays the configuration of the specified class of the specified policy-map.

Command	Purpose
Router# **show policy-map interface** *interface-name*	Displays the configuration of all classes configured for all policy-maps on the specified interface.
Router# **show queue** *interface-type interface-number*	Displays queueing configuration and statistics for a particular interface.

Configuring LLQ

LLQ Overview

Low-Latency Queuing (LLQ) uses the **priority** command to allocate bandwidth to the class maps in the policy-map.

LLQ is supported on the output of the following interfaces:

- Main Layer 3 interface
- Layer 3 subinterface

LLQ Restrictions and Usage Guidelines

When configuring LLQ, follow these restrictions and usage guidelines:

- Ingress LLQ is not supported
- Egress LLQ
  - LLQ/PQ is supported only on egress for ES+ Layer 3 and Layer 3 subinterfaces.
  - Dual priority queues are supported (priority level 1 and priority level 2).
  - LLQ configuration is allowed at the child policy.
  - The **priority** and **priority level** commands are supported but you cannot use both in the same policy-map.
  - Basic priority/low latency queue with bit rates is not supported.
  - Basic low latency queue with percent is not supported.
  - Priority queue with bit rates is not supported.

Configuring LLQ

	Command	Purpose
Step 1	Router> **enable**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	Router# **configure terminal**	Enters global configuration mode.
Step 3	Router(config)# **policy-map** *policy-name*	Specifies the name of the policy-map to configure.

	Command	Purpose
Step 4	`Router(config-pmap)#` **`class`** `{`*`class-name`*`	`**`class-default`**`}`	Specifies the name of a predefined class included in the service policy.
Step 5	`Router(config-pmap-c)#` **`police`** *`bps-value`* **`conform-action`** *`action`* **`exceed-action`** *`action`*	Specifies a maximum bandwidth usage by a traffic class through the use of a token bucket algorithm, where: • *bps-value*—Specifies the average rate in bits per second. Valid values are 64000 to 200000000. • *action*—Specifies the he actions that are taken on a packet when it conforms or exceeds. The possible actions are shown in Table A-2.
	Or	
	`Router(config-pmap-c)#` **`police cir percent`** `%` **`conform-action`** *`action`* **`exceed-action`** *`action`*	Configures traffic policing on the basis of a percentage of bandwidth available on an interface, where: • **cir**—Specifies the committed information rate. Indicates that the CIR will be used for policing traffic. • **percent**—Specifies that a percentage of bandwidth will be used for calculating the CIR. • *%*—Specifies the CIR bandwidth percentage. Valid values are 1 to 100. • *action*—Specifies the he actions that are taken on a packet when it conforms or exceeds. The possible actions are shown in Table A-2.
	Or	
	`Router(config-pmap-c)#` **`police cir`** *`bps-value`* **`pir`** *`bps-value`* **`conform-action`** *`action`* **`exceed-action`** *`action`* **`violate-action`** *`action`*	Configures traffic policing using two rates, the CIR and the PIR, where: • **cir**—Specifies the committed information rate. Indicates that the CIR will be used for policing traffic. • **pir**—Specifies the peak information rate. Indicates that the PIR will be used for policing traffic. • *bps-value*—Specifies the average rate in bits per second. Valid values are 64000 to 200000000. • *action*—Specifies the he actions that are taken on a packet when it conforms or exceeds. The possible actions are shown in Table A-2.
	Or	

	Command	Purpose
	Router(config-pmap-c)# **police cir percent** *percentage* **pir percent** % **conform-action** *action* **exceed-action** *action* **violate-action** *action*	Configures traffic policing using two rates, the CIR and the PIR, where: • **cir**—Specifies the committed information rate. Indicates that the CIR will be used for policing traffic. • **percent**—Specifies that a percentage of bandwidth will be used for calculating the CIR. • *%*—Specifies the CIR or PIR bandwidth percentage. Valid values are 1 to 100. • **pir**—Specifies the peak information rate. Indicates that the PIR will be used for policing traffic. • *action*—Specifies the he actions that are taken on a packet when it conforms or exceeds. The possible actions are shown in Table A-2.
Step 6	Router(config-pmap-c)# **priority** [*level*] {1-2}	Gives strict priority to a class of traffic belonging to the policy-map.

The following example configures a simple LLQ QoS policy on a class c1 with strict priority setting.

```
Router> enable
Router# configure terminal
Router(config)# policy map qos_llq
Router(config-pmap)# class c1
Router(config-pmap-c)# police 500000000
Router(config-pmap-c)# priority
```

The following example configures an LLQ policy with multiple priority classes with a smallest percent value and default burst value for testing:

```
Router> enable
Router# configure terminal
Router(config-pmap)# class-map voice
Router(config-pmap-c)# police cir percent 10
Router(config-pmap-c)# priority
Router(config-pmap)# class-map video
Router(config-pmap-c)# police cir percent 20
Router(config-pmap-c)# priority
Router(config-pmap)# class-default
```

Configuring Hierarchical QoS

- Hierarchical QoS Overview, page A-30
- Hierarchical QoS Examples, page A-31

Hierarchical QoS Overview

The ES+ line cards support hierarchical QoS (H-QoS) that you configure using Cisco Modular QoS CLI (MQC). The following H-QoS capabilities are supported:

- Two-level H-QoS (A policy-map with two levels has three levels of hierarchy when attached on the main interface.)
- Granular QoS—Policing and shaping, down to 64 Kbps data rate
- Color blind policing—2-rate, 3-color policers and 1-rate, 2-color policers

Note Color aware policing is not supported

- Egress classification
- QoS on TenGigabitEthernet 802.1Q subinterface(s)
- Egress Class-based Weighted Fair Queuing (CBWFQ)
- Low Latency Queuing (LLQ) (Ingress and Egress)
- Egress H-QoS on IP/MPLS and Layer 2 CoS classification
- AToM QoS features on Ethernet L2VPNs
- Hierarchical policing
- Scaling for ES+ line cards:
  - 128,000 queues
  - 16,000 traffic shapers
  - 48,000 policers per NP
  - 8,000 H-QoS policy-maps per NP in egress.
  - 24000 policers per ES+ line card.

Follow these restrictions while configuring Hierarchical QoS:

- Support up to128,000 queues.
- Support up to 16,000 traffic shapers.
- Support up to 48,000 policers per NP.
- Support up to 8,000 H-QoS policy-maps per NP in egress and 3904 policy-maps per NP in ingress.

Follow these restrictions and usage guidelines while configuring Hierarchical QoS:

- Support up to 16 queues for each port.
- Support up to 16 queues per port channel.
- Single fabric connection ES+ line cards support up to 24000 policers per line card.
- Dual fabric connection ES+ line cards support up to 48000 policers per line card.
- Supports up to 8,000 H-QoS policy maps per NP in egress and 3904 H-QoS policy-maps per NP in ingress.
- If a child policy is applied with a QoS queuing feature, only the child classes with queuing feature is considered for the queue restriction per port. The parent class is not considered.
- If a child policy is not applied with a QoS queuing feature, then parent class is considered for queue restriction per port.

ES+ line cards support parent and child class hierarchical levels on the physical (main) interface and subinterface (logical layer).

A policy-map with two levels has three levels of hierarchy when attached on the main interface, and four levels of hierarchy when attached on a subinterface.

Table A-5 provides information about supported H-QoS features.

Table A-5 Hierarchical QoS Feature Support

Interface Type	Marking	Policing	Shaping	Bandwidth	Priority and Priority Percent	Priority and Policing	WRED
Layer 3 interface (routed port)	CoS, precedence/DSCP, EXP	Yes	Yes	Yes	**No**	Yes	Yes
Layer 3 subinterface	CoS, precedence/DSCP, EXP	Yes	Yes	Yes	**No**	Yes	Yes

Hierarchical QoS Examples

This example configures the child policy to allocate different percentages of bandwidth by class:

```
Router> enable
Router# configure terminal
Router(config)# policy-map child
Router(config-pmap)# class user-a
Router(config-pmap-c)# bandwidth percent 40
Router(config-pmap-c)# exit
Router(config-pmap)# class user-b
Router(config-pmap-c)# bandwidth percent 60
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

This example applies the parent service policy to an output subinterface:

```
Router> enable
Router# configure terminal
Router(config)# interface tengigabitethernet 2/1.1
Router(config-if-srv)# encapsulation dot1q 11
Router(config-if)# service-policy output parent
```

This example shows how to configure a 2 level H-QoS policy on a main interface:

```
Router(config)# policy-map child_1
Router(config-pmap)# class prec1
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# exit
Router(config-pmap)# class prec2
Router(config-pmap-c)# priority level 2
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# police 100000
Router(config-pmap-c)# exit
Router(config)# policy-map HQoS_parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 100000000
Router(config-pmap-c)# service-policy child_1
```

This example shows how to configure a 2 level H-QoS policy on a subinterface:

```
Router(config)# policy-map child_1
Router(config-pmap)# class cos1
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# exit
Router(config-pmap)# class cos 2
Router(config-pmap-c)# priority level 2
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# police 100000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# policy-map HQoS_parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 100000000
Router(config-pmap-c)# service-policy child_1
```

This example configures an ingress 2-level H-QOS policy on a main-interface:

```
Router(config)# policy-map child_1
Router(config-pmap)# class prec123
Router(config-pmap-c)# random-detect precedence based
Router(config-pmap-c)# exit
Router(config-pmap)# class prec456
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap)# exit
Router(config)# policy-map HQoS_parent
Router(config-pmap)# class ACL_c1
Router(config-pmap-c)# police 100000
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# service policy child_1
Router(config-pmap-c)# exit
Router(config-pmap)# class ACL_c2
Router(config-pmap-c)# police 100000
Router(config-pmap-c)# priority level 2
Router(config-pmap-c)# service policy child_2
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# police 100000
Router(config-pmap-c)# service policy child_3
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# policy-map HQos_grandparent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape 100000000
Router(config-pmap-c)# service-policy HQoS_parent
```

Configuring MPLS Traffic Engineering Class-Based Tunnel Selection

- MPLS Traffic Engineering Overview, page A-33
- MPLS Traffic Engineering Class-Based Tunnel Selection Restrictions and Usage Guidelines, page A-33

- Creating Multiple MPLS Member TE or DS-TE Tunnels with the Same Headend and the Same Tailend, page A-34
- Creating a Master Tunnel, Attaching Member Tunnels, and Making the Master Tunnel Visible, page A-35
- Verifying the MPLS Configuration, page A-38

MPLS Traffic Engineering Overview

Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) enables you to dynamically route and forward traffic with different class of service (CoS) values onto different TE tunnels between the same tunnel headend and the same tailend. The TE tunnels can be regular TE tunnels or DiffServ-aware TE (DS-TE) tunnels.

The set of TE/DS-TE tunnels from the same headend to the same tailend that you configure to carry different CoS values is referred to as a "tunnel bundle." Tunnels are "bundled" by creating a master tunnel and then attaching member tunnels to the master tunnel. After configuration, CBTS dynamically routes and forwards each packet into the tunnel that meets the following requirements:

- Is configured to carry the CoS of the packet
- Has the right tailend for the destination of the packet

Because CBTS offers dynamic routing over DS-TE tunnels and requires minimum configuration, it greatly eases deployment of DS-TE in large-scale networks.

CBTS can distribute all CoS values on eight different tunnels or multiple COS value to multiple tunnels .

CBTS also allows the TE tunnels of a tunnel bundle to exit headend routers through different interfaces.

CBTS configuration involves performing the following tasks:

- Creating multiple (DS-) TE tunnels with the same headend and tailend and indicating on each of these tunnels which CoSs are to be transported on the tunnel.
- Creating a master tunnel, attaching the member tunnels to it, and making the master tunnel visible for routing.

MPLS Traffic Engineering Class-Based Tunnel Selection Restrictions and Usage Guidelines

When configuring MPLS Traffic Engineering Class-Based Tunnel Selection (CBTS), follow these restrictions and usage guidelines:

- CBTS has the following prerequisites:
  - MPLS enabled on all tunnel interfaces
  - Cisco Express Forwarding (CEF) or distributed CEF (dCEF) enabled in general configuration mode
- CBTS has the following restrictions:
  - For a given destination, all CoS values are carried in tunnels terminating at the same tailend. Either all CoS values are carried in tunnels or no values are carried in tunnels. In other words, for a given destination, you cannot map some CoS values in a DS-TE tunnel and other CoS values in a Shortest Path First (SPF) Label Distribution Protocol (LDP) or SPF IP path.

- No LSP is established for the master tunnel and regular traffic engineering attributes (bandwidth, path option, fast reroute) are irrelevant on a master tunnel. TE attributes (bandwidth, bandwidth pool, preemption, priorities, path options, and so on) are configured completely independently for each tunnel.
- CBTS does not allow load-balancing of a given EXP value in multiple tunnels. If two or more tunnels are configured to carry a given experimental (EXP) value, CBTS picks one of these tunnels to carry this EXP value (which is calculated through pre-defined rules).
- CBTS supports aggregate control of bumping (that is, it is possible to define default tunnels to be used if other tunnels go down). However, CBTS does not allow control of bumping if the default tunnel goes down. CBTS does not support finer-grain control of bumping. For example, if the voice tunnel goes down, redirect voice to T2, but if video goes down, redirect to T3.
- The operation of CBTS is not supported with Any Transport over MPLS (AToM), MPLS TE Automesh, or label-controlled (LC) ATM.

Creating Multiple MPLS Member TE or DS-TE Tunnels with the Same Headend and the Same Tailend

Perform the following task to create multiple MPLS member TE or DS-TE tunnels with the same headend and same tailend and to configure EXP values to be carried by each of these tunnels. The procedure begins in global configuration mode.

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`interface tunnel`** *`number`*	Configures a tunnel interface type and enters interface configuration mode. • *number*—Number of the tunnel interface that you want to create or configure.
Step 4	`Router(config-if)#` **`ip unnumbered`** *`type number`*	Enables IP processing on an interface without assigning an explicit IP address to the interface. • *type*—Type of another interface on which the router has an assigned IP address. • *number*—Number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface.
Step 5	`Router(config-if)#` **`tunnel destination`** `{`*`hostname`* `|` *`ip-address`*`}`	Specifies the destination of the tunnel for this path option. • *hostname*—Name of the host destination. • *ip-address*—IP address of the host destination expressed in four-part, dotted decimal notation.
Step 6	`Router(config-if)#` **`tunnel mode mpls traffic-eng`**	Sets the mode of a tunnel to MPLS for TE.

	Command	Purpose
Step 7	`Router(config-if)#` **`tunnel mpls traffic-eng bandwidth`** `[`**`sub-pool`** `|` **`global`**`]` *`bandwidth`*	Configures the bandwidth for the MPLS TE tunnel. If automatic bandwidth is configured for the tunnel, use the **tunnel mpls traffic-eng bandwidth** command to configure the initial tunnel bandwidth, which is adjusted by the auto-bandwidth mechanism. • **sub-pool**—(Optional) Indicates a subpool tunnel. • **global**—(Optional) Indicates a global pool tunnel. Entering this keyword is not necessary, because all tunnels are global pool in the absence of the **sub-pool** keyword. But if users of pre-DiffServ-aware Traffic Engineering (DS-TE) images enter this keyword, it is accepted. • *bandwidth*—Bandwidth, in kilobits per second, set aside for the MPLS traffic engineering tunnel. Range is between 1 and 4294967295. **Note** You can configure any existing **mpls traffic-eng** command on these TE or DS-TE tunnels.
Step 8	`Router(config-if)#` **`tunnel mpls traffic-eng exp`** `[`*`list-of-exp-values`*`] [`**`default`**`]`	Specifies an EXP value or values for an MPLS TE tunnel. • *list-of-exp-values*—EXP value or values that are are to be carried by the specified tunnel. Values range from 0 to 7. • **default**—The specified tunnel is to carry all EXP values that are: – Not explicitly allocated to another tunnel – Allocated to a tunnel that is currently down
Step 9	`Router(config-if)#` **`exit`**	Exits to global configuration mode.

Repeat on the same headend router to create additional tunnels from this headend to the same tailend.

Creating a Master Tunnel, Attaching Member Tunnels, and Making the Master Tunnel Visible

Perform the followings task to create a master tunnel, attach member tunnels to it, and make the master tunnel visible for routing. The procedure begins in global configuration mode.

	Command	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.

	Command	Purpose
Step 3	`Router(config)#` **`interface tunnel`** *`number`*	Configures a tunnel interface type and enters interface configuration mode, where *number* is the number of the tunnel interface that you want to create or configure.
Step 4	`Router(config-if)#` **`ip unnumbered`** *`type number`*	Enables IP processing on an interface without assigning an explicit IP address to the interface. • *type*—Type of another interface on which the router has an assigned IP address. • *number*—Number of another interface on which the router has an assigned IP address. It cannot be another unnumbered interface.
Step 5	`Router(config-if)#` **`tunnel destination`** `{`*`hostname`* `|` *`ip-address`*`}`	Specifies the destination of the tunnel for this path option. • *hostname*—Name of the host destination. • *ip-address*—IP address of the host destination expressed in four-part, dotted decimal notation.
Step 6	`Router(config-if)#` **`tunnel mode mpls traffic-eng exp-bundle master`**	Specifies this is the master tunnel for the CBTS configuration.
Step 7	`Router(config-if)#` **`tunnel mode mpls traffic-eng exp-bundle member`** *`tunnel-id`*	Attaches a member tunnel to the master tunnel. • *tunnel-id*—Number of the tunnel interface to be attached to the master tunnel. Repeat this command for each member tunnel.
Step 8	`Router(config-if)#` **`tunnel mpls traffic-eng autoroute announce`**	Specifies that the Interior Gateway Protocol (IGP) should use the tunnel (if the tunnel is up) in its enhanced shortest path first (SPF) calculation.
Step 9	`Router(config-if)#` **`tunnel mpls traffic-eng autoroute metric`** `{`**`absolute`** `|` **`relative`**`}` *`value`*	(Optional) Specifies the MPLS TE tunnel metric that the IGP-enhanced SPF calculation uses. • **absolute**—Indicates the absolute metric mode; you can enter a positive metric value. • **relative**—Indicates the relative metric mode; you can enter a positive, negative, or zero value. • *value*—Metric that the IGP enhanced SPF calculation uses. The relative value can be from -10 to 10.

Note Even though the value for a relative metric can be from -10 to +10, configuring a tunnel metric with a negative value is considered a misconfiguration. If the metric to the tunnel tailend appears to be 4 from the routing table, then the cost to the tunnel tailend router is actually 3 because 1 is added to the cost for getting to the loopback address. In this instance, the lowest value that you can configure for the relative metric is -3.

Note Alternatively, static routing could be used instead of autoroute to make the TE or DS-TE tunnels visible for routing.

The following example shows how to configure Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Class-Based Tunnel Selection (CBTS). Tunnel1, Tunnel2, and Tunnel3 are member tunnels, and Tunnel4 is the master tunnel.

```
Router> enable
Router# configure terminal
Router(config)# interface tunnel1
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel destination 24.1.1.1
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000
Router(config-if)# tunnel mpls traffic-eng exp 5
Router(config-if)# exit

Router(config)# interface tunnel2
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel destination 24.1.1.1
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel mpls traffic-eng bandwidth 50000
Router(config-if)# tunnel mpls traffic-eng exp 3 4
Router(config-if)# exit

Router(config)# interface tunnel3
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel destination 24.1.1.1
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel mpls traffic-eng bandwidth 10000
Router(config-if)# tunnel mpls traffic-eng exp default
Router(config-if)# exit

Router(config)# interface tunnel4
Router(config-if)# ip unnumbered loopback0
Router(config-if)# tunnel destination 24.1.1.1
Router(config-if)# tunnel mode mpls traffic-eng
Router(config-if)# tunnel mpls traffic-eng exp-bundle master
Router(config-if)# tunnel mpls traffic-eng exp-bundle member tunnel1
Router(config-if)# tunnel mpls traffic-eng exp-bundle member tunnel2
Router(config-if)# tunnel mpls traffic-eng exp-bundle member tunnel3
Router(config-if)# tunnel mpls traffic-eng autoroute announce
```

Verifying the MPLS Configuration

The following **show** commands can be used to verify that the MPLS TE or DS-TE tunnels are operating and announced to the IGP. The commands are all entered in privileged EXEC configuration mode.

Command	Purpose
show mpls traffic-eng topology {*A.B.C.D* | **igp-id** {**isis** *nsap-address* | **ospf** *A.B.C.D*} [**brief**]}	Shows the MPLS traffic engineering global topology as currently known at this node. • *A.B.C.D*—Specifies the node by the IP address (router identifier to interface address). • **igp-id**—Specifies the node by IGP router identifier. • **isis** *nsap-address*—Specifies the node by router identification (*nsap-address*) if you are using Integrated Intermediate System-to-Intermediate System (IS-IS). • **ospf** *A.B.C.D*—Specifies the node by router identifier if you are using Open Shortest Path First (OSPF). • **brief**—Provides a less-detailed version of the topology.
show mpls traffic-eng exp	Displays EXP mapping.
show ip cef [*type number*] [**detail**]	Displays entries in the forwarding information base (FIB) or displays a summary of the FIB. • *type number*—Identifies the interface type and number for which to display FIB entries. • **detail**—Displays detailed FIB entry information.
show mpls forwarding-table [*network* {*mask* | *length*} [**detail**]]	Displays the contents of the MPLS label forwarding information base (LFIB). • *network*—Identifies the destination network number. • *mask*—Identifies the network mask to be used with the specified network. • *length*—Identifies the number of bits in the destination mask. • **detail**—Displays information in long form (includes length of encapsulation, length of MAC string, maximum transmission unit [MTU], and all labels).
show mpls traffic-eng autoroute	Displays tunnels that are announced to the Interior Gateway Protocol (IGP).

The **show mpls traffic-eng topology** command output displays the MPLS TE global topology:

```
Router# show mpls traffic-eng topology 10.0.0.1

IGP Id: 10.0.0.1, MPLS TE Id:10.0.0.1 Router Node  (ospf 10  area 0) id 1
link[0]: Broadcast, DR: 180.0.1.2, nbr_node_id:6, gen:18
      frag_id 0, Intf Address:180.0.1.1
      TE metric:1, IGP metric:1, attribute_flags:0x0
      SRLGs: None
      physical_bw: 100000 (kbps), max_reservable_bw_global: 1000 (kbps)
      max_reservable_bw_sub: 0 (kbps)
```

```
                      Global Pool       Sub Pool
    Total Allocated   Reservable        Reservable
    BW (kbps)         BW (kbps)         BW (kbps)
       ---------------   -----------       ----------
 bw[0]:           0            1000               0
 bw[1]:           0            1000               0
 bw[2]:           0            1000               0
 bw[3]:           0            1000               0
 bw[4]:           0            1000               0
 bw[5]:           0            1000               0
 bw[6]:           0            1000               0
 bw[7]:         100             900               0

link[1]: Broadcast, DR: 180.0.2.2, nbr_node_id:7, gen:19
      frag_id 1, Intf Address:180.0.2.1
      TE metric:1, IGP metric:1, attribute_flags:0x0
      SRLGs: None
      physical_bw: 100000 (kbps), max_reservable_bw_global: 1000 (kbps)
          max_reservable_bw_sub: 0 (kbps)
                          Global Pool       Sub Pool
        Total Allocated   Reservable        Reservable
        BW (kbps)         BW (kbps)         BW (kbps)
    ---------------   -----------       ----------
 bw[0]:           0            1000               0
 bw[1]:           0            1000               0
 bw[2]:           0            1000               0
 bw[3]:           0            1000               0
 bw[4]:           0            1000               0
 bw[5]:           0            1000               0
 bw[6]:           0            1000               0
 bw[7]:           0            1000               0
```

The **show mpls traffic-eng exp** command output displays EXP mapping information about a tunnel:

```
Router# show mpls traffic-eng exp

Destination: 10.0.0.9
    Master:Tunnel10Status: IP

    Members: StatusConf EXPActual EXP
    Tunnel1UP/ACTIVE55
    Tunnel2UP/ACTIVEdefault0 1 2 3 4 6 7
    Tunnel3UP/INACTIVE(T)2
    Tunnel4DOWN3
    Tunnel5UP/ACTIVE(NE)

(T)=Tailend is different to master
(NE)=There is no exp value configured on this tunnel.
```

The **show ip cef detail** command output displays detailed FIB entry information for a tunnel:

```
Router# show ip cef tunnel1 detail

IP CEF with switching (Table Version 46), flags=0x0
31 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 2
2 instant recursive resolutions, 0 used background process
8 load sharing elements, 8 references
6 in-place/0 aborted modifications
34696 bytes allocated to the FIB table data structures
universal per-destination load sharing algorithm, id 9EDD49E1
1(0) CEF resets
    Resolution Timer: Exponential (currently 1s, peak 1s)
    Tree summary:
    8-8-8-8 stride pattern
```

```
    short mask protection disabled
31 leaves, 23 nodes using 26428 bytes
Table epoch: 0 (31 entries at this epoch)
Adjacency Table has 13 adjacencies
10.0.0.9/32, version 45, epoch 0, per-destination sharing
0 packets, 0 bytes
tag information set, all rewrites inherited
local tag: tunnel head
via 0.0.0.0, Tunnel1, 0 dependencies
traffic share 1
next hop 0.0.0.0, Tunnel1
valid adjacency
tag rewrite with Tu1, point2point, tags imposed {12304}
0 packets, 0 bytes switched through the prefix
tmstats: external 0 packets, 0 bytes
internal 0 packets, 0 bytes
```

The **show mpls forwarding-table detail** command output displays detailed information from the MPLS LFIB:

```
Router# show mpls forwarding 10.0.0.9 detail

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
Tun hd Untagged    10.0.0.9/32       0          Tu1        point2point
    MAC/Encaps=14/18, MRU=1500, Tag Stack{12304}, via Fa6/0
    00027D884000000ED70178A88847 03010000
    No output feature configured
Per-exp selection: 1
Untagged    10.0.0.9/32       0          Tu2        point2point
    MAC/Encaps=14/18, MRU=1500, Tag Stack{12305}, via Fa6/1
    00027D884001000ED70178A98847 03011000
    No output feature configured
Per-exp selection: 2  3
Untagged    10.0.0.9/32       0          Tu3        point2point
    MAC/Encaps=14/18, MRU=1500, Tag Stack{12306}, via Fa6/1
    00027D884001000ED70178A98847 03012000
    No output feature configured
Per-exp selection: 4  5
Untagged    10.0.0.9/32       0          Tu4        point2point
    MAC/Encaps=14/18, MRU=1500, Tag Stack{12307}, via Fa6/1
    00027D884001000ED70178A98847 03013000
    No output feature configured
Per-exp selection: 0  6  7
```

The **show mpls traffic-eng autoroute** command output displays tunnels that are announced to the Interior Gateway Protocol (IGP).

```
Router# show mpls traffic-eng autoroute

MPLS TE autorouting enabled
destination 10.0.0.9, area ospf 10  area 0, has 4 tunnels
Tunnel1     (load balancing metric 20000000, nexthop 10.0.0.9)
(flags: Announce)
Tunnel2     (load balancing metric 20000000, nexthop 10.0.0.9)
(flags: Announce)
Tunnel3     (load balancing metric 20000000, nexthop 10.0.0.9)
(flags: Announce)
Tunnel4     (load balancing metric 20000000, nexthop 10.0.0.9)
(flags: Announce)
```

Configuring IPoDWDM

IPoDWDM-Capable ES+ Line Cards

Dense wavelength-division multiplexing (IPoDWDM) is supported on these ES+ line cards:

- 76-ES+XT-2TG3CXL
- 76-ES+XT-4TG3CXL

WAN PHY and OTN Support on ES+ Line Cards

WAN PHY and OTN Overview

ES+ line card ports support the Optical Transport Network (OTN) and Wide Area Network (WAN) PHY.

This feature provides the software functionality to support OTN and WAN PHY on ES+ line cards.

WAN PHY leverages 10 Gig SONET infrastructure and accesses WAN facilities using:

- Dark Fiber
- Dark Wavelengths
- SONET TDM Networks

This feature provides low cost optic solutions required for short distances networks that implement store and forward network design requiring no optical amplifiers.

The OTN is based on the Optical Transport Hierarchy (OTH) developed by ITU. The OTN is based on the network architecture defined in ITU G.872 "Architecture for the Optical Transport Network (OTN)". The G.872 standard defines an architecture composed of the Optical Channel (OCh), Optical Multiplex Section (OMS), and Optical Transmission Section (OTS). The use of digitally framed signal with digital overhead for optical channel enables you to implement the management requirements of OCh. It also allows the use of Forward Error Correction (FEC) system to improve the system performance. The two new digital layer networks introduced to implement this feature are ODU and OTU.

OTN architecture (ITU-T G.872 standard) defines two interface classes:

- Inter-domain interface (IrDI): The OTN IrDI interface class defines the interface (with the 3Rs [Reamplification, Reshaping and Retiming] processing) at each end of the operator interface. the operator interface can also be the interface between different vendors within the same operator
- Intra-domain interface (IaDI): The IaDI interface class defines the interface within an operator or a vendor domain.

OTN has the following advantages:

- Stronger forward error correction
- More levels of Tandem Connection Monitoring (TCM)
- Transparent transport of client signals
- Switching scalability

Restrictions and Usage Guidelines

When configuring the WAN PHY / OTN support on ES+ line cards, follow these restrictions and usage guidelines:

- The distances between the two switching equipments using the WAN PHY and the DWDM facility depends on the XFP used. Refer the data sheets of relevant XFP.
- The MAC address is common for WAN PHY and LAN PHY. The WAN PHY operates at a rate compatible with the payload rate of OC-192c/VC-464c.

Configuring ITU-T G.709 Transport Modes

Use the **transport-mode** command in interface configuration mode to configure LAN, WAN, and OTN transport modes. The **transport-mode** command **otn** option has the **bit-transparent** sub-option, using which bit transparent mapping into OPU1e or OPU2e can be configured.

Note The hardware combination of Cisco-INTEL OC192 + 10GBASE-L XFP is not supported because of bit rate incompatibility between INTEL XFP and OTN for the following transport mode configurations:

- opu1e - 10GBASE-R over OPU1e without fixed stuffing (11.0491Gb/s)
- opu2e - 10GBASE-R over OPU2e with fixed stuffing (11.0957Gb/s)

	Command or Action	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`interface tengigabitethernet`** *`slot/port`*	Specifies the Ten Gigabit Ethernet interface to configure.
Step 4	`Router(config-if)#` **`transport-mode`** `{`**`lan`** `|` **`wan`** `|` **`otn bit-transparent`** `{`**`opu1e`** `|` **`opu2e`**`}}`	Configures the transport mode.

DWDM Provisioning

All DWDM provisioning configurations take place on the controller. To configure a DWDM controller, use the **controller dwdm** command in global configuration mode.

The g709 configuration commands can be used only when the controller is in the shutdown state. Use the **no shutdown** command after configuring the parameters, to remove the controller from shutdown state and to enable the controller to move to up state.

	Command or Action	Purpose
Step 1	`Router>` **`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)#` **`controller dwdm`** *`slot/port`*	Configures the DWDM controller.

The following are examples of IP over DWDM commands:

```
Router# show run int tengigabitethernet 2/3
Building configuration...

Current configuration : 96 bytes
!
interface TenGigabitEthernet2/3
 ip address 11.11.11.2 255.255.255.0
 transport-mode otn bit-transparent opu2e
end

Router# show controller dwdm 2/3
G709 Information:

Controller dwdm 3/1, is down (shutdown)

Transport mode LAN (10GBASE-R, 10.3125Gb/s)

TAS state is : OOS
Description: connected to a ginsu LC
G709 status : Disabled

OTU
LOS = 18          LOF = 0           LOM = 0
AIS = 0           BDI = 1           BIP = 14504
TIM = 0           IAE = 0           BEI = 2289

ODU
AIS = 0           BDI = 0           TIM = 0
OCI = 0           LCK = 0           PTIM = 0
BIP = 14500       BEI = 2266

FEC Mode: FEC
EC(current second) = 0
EC = 31361        UC = 56318597


        pre-FEC BER < 9.00E-11
        Q > 6.45        Q Margin > 7.52  DBQ
Detected Alarms: NONE
Asserted Alarms: NONE
Detected Alerts: NONE
Asserted Alerts: NONE
Alarm reporting enabled for: LOS LOF LOM OTU-AIS OTU-IAE OTU-BDI OTU-TIM ODU-AIS ODU-OCI
ODU-LCK ODU-BDI ODU-PTIM ODU-TIM ODU-BIP Alert reporting enabled for: OTU-SM-TCA
ODU-SD-BER ODU-SF-BER ODU-PM-TCA BER thresholds: ODU-SF = 10e-3  ODU-SD = 10e-6 TCA
thresholds: SM = 10e-3  PM = 10e-3

OTU TTI Sent     String ASCII: This_is_a_static_string
OTU TTI Received String ASCII:
OTU TTI Received String HEX  : 0000000000000000000000000000000000000000000000000
                               0000000000000000000000000000000000000000000000000
```

```
                               000000000000000000000000000000 OTU TTI Expected String
ASCII: This_is_a_static_string

ODU TTI Sent     String ASCII: This_is_a_static_string
ODU TTI Received String ASCII:
ODU TTI Received String HEX  : 0000000000000000000000000000000000000000000000000
                               0000000000000000000000000000000000000000000000000
                               000000000000000000000000000000 ODU TTI Expected String
ASCII: This_is_a_static_string

Optics Information:

optics type: DWDM XFP Tunable
Wavelength: C-band, channel 10, 1558.17 nm, 192.40 THz Transceiver Rx optical power  =
-40.0 dBm
Transceiver Tx power        =  1.5 dBm
TX Laser current bias       = 20988 uAmps

Virtual Link Info:

 Adjacency info: This_is_a_static_string


C7600 Node ID :

        0 :26:B :28:68:80


Connectivity Info:

         Network Connection ID : This_is_a_static_string


Network SRLG values:


         Set 1:  6142   19113   14477   26689   4989   31230
         Set 2:  14967   7234   29164   19852   15452   17460
         Set 3:  14852   28561   6364   12832   21486   14312
         Set 4:  30337   19184   28532   15403   21048   27105
         Set 5:  18102   24607   16426   14253   21500   21952
         Set 6:  13523   17545   7863   538   5251   18205
         Set 7:  22331   27781   17862   26935   10028   16539
         Set 8:  865   29015   7144   20299   27504   2190
         Set 9:  13470   7222   8500   6988   18852   20882
         Set 10:  21512   702   14117   1870   19304   13075
         Set 11:  11919   26281   1898   18454   9948   15302
         Set 12:  24263   24747   5275   29138   17325   19226
         Set 13:  10917   18739   16263   20739   13147   18471
         Set 14:  1126   24967   26662   16266   32124   32739
         Set 15:  20342   29828   7591   18968   2421   24934
         Set 16:  3366   27109   22805   3591   7227   9339

Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# int tengigabitethernet 2/3
Router(config-if)# transport-mode ?
  lan  10GBASE-R LAN pass-through (10.3125Gb/s)
  otn  10GE over Optical Transport Network (G.709)
  wan  10GBASE-W WAN SONET/SDH (9.95328Gb/s)

Router(config-if)# transport-mode otn ?
  bit-transparent  10GBASE-R transparently mapped into OTU-2
```

```
Router(config-if)# transport-mode otn bit-transparent  ?
  opu1e  10GBASE-R over OPU1e without fixed stuffing (11.0491Gb/s)
  opu2e  10GBASE-R over OPU2e with fixed stuffing (11.0957Gb/s)


Router(config-if)# transport-mode otn bit-transparent opu2e
Router(config-if)# end
Router# show int tengigabitethernet2/3
TenGigabitEthernet2/3 is up, line protocol is up (connected)
  Hardware is X40G 10Gb 802.3, address is 00d0.03e2.1c00 (bia 00d0.03e2.1c00)
  Internet address is 11.11.11.1/24
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 10Gb/s, clock source internal
  Transport mode OTN (10GBASE-R over OPU2e with fixed stuffing, 11.0957Gb/s)
  input flow-control is off, output flow-control is off
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  L2 Switched: ucast: 0 pkt, 0 bytes - mcast: 2360 pkt, 221372 bytes
  L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes mcast
  L3 out Switched: ucast: 0 pkt, 0 bytes mcast: 0 pkt, 0 bytes
     2392 packets input, 223718 bytes, 0 no buffer
     Received 2477 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     2477 packets output, 229905 bytes, 0 underruns
     0 output errors, 0 collisions, 13 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out

Router#
Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# controller dwdm 2/3
Router(config-controller)# ?
Controller configuration commands:
  default      Set a command to its defaults
  description  Controller specific description
  exit         Exit from controller configuration mode
  g709         Configure G709 parameters
  help         Description of the interactive help system
  no           Negate a command or set its defaults
  shutdown     Configure dwdm controller processing

Router(config-controller)# g709 ?
fec  Configure FEC mode
odu  Configure odu parameters
otu  Configure otu parameters
tti-processing Configure Trail Trace Identifier  processing

Router(config-controller)# g709 fec ?
disable   Disable FEC
enhanced  Enhanced FEC mode
```

```
standard  Standard FEC mode

Router(config-controller)# g709 odu ?
overhead   Configure ODU overhead
report     Configure odu alarm reporting
threshold  Configure odu threshold

Router(config-controller)#g709 odu overhead ?
tti  Configure ODU Trail Trace Identifier buffer

Router(config-controller)#g709 odu overhead tti ?
expected   Set expected TTI
sent       Set transmit TTI

Router(config-controller)#g709 odu overhead tti expected ?
ascii  Enter ASCII string
hex    Enter hex string- Length should be even number

Router(config-controller)#g709 odu overhead tti expected ascii ?
    WORD  LINE  ASCII text (Max 64 characters)

Router(config-controller)#g709 odu overhead tti expected hex ?
    Hex-data  LINE  Hex nibbles (Max 128- The string length should
             be an even number)
Router(config-controller)#g709 odu overhead tti sent ?
ascii  Enter ASCII string
hex    Enter hex string- Length should be even number

Router(config-controller)#g709 odu overhead tti sent ascii ?
WORD  LINE ASCII text (Max 64 characters)

Router(config-controller)#g709 odu overhead tti sent hex ?
Hex-data  LINE  Hex nibbles (Max 128- The string length should
             be an even number)
Router(config-controller)# g709 odu report ?
ais      Set Alarm Indication Signal reporting status
bdi      Set Backward Defect Indication reporting status
lck      Set Upstream Connection Locked reporting status
oci      Set Open Connection Indication reporting status
pm-tca   Set Path Monitoring BER TCA reporting status
ptim     Set Payload Type Identifier Mismatch reporting status
sd-ber   Set SM BER in excess of SD threshold reporting status
sf-ber   Set SM BER in excess of SF threshold reporting status
tim      Set Trace Identifier Mismatch reporting status

Router(config-controller)# g709 odu threshold ?
pm-tca   Set Path Monitoring Threshold Crossing Alert threshold
sd-ber   Set Signal Degrade BER threshold
sf-ber   Set Signal Failure BER threshold

Router(config-controller)# g709 odu threshold pm-tca ?
<3-9>  Bit Error Rate (10 to the minus n) (default 3)
<cr>

Router(config-controller)# g709 odu threshold sd-ber ?
<3-9>  Bit Error Rate (10 to the minus n) (default 6)
<cr>

Router(config-controller)# g709 odu threshold sf-ber ?
<3-9>  Bit Error Rate (10 to the minus n) (default 3)
<cr>
Router(config-controller)# g709 otu ?
    overhead   Configure OTU overhead
    report     Configure otu alarm reporting
```

```
    threshold  Configure otu threshold

Router(config-controller)#g709 otu overhead ?
tti  Configure OTU Trail Trace Identifier buffer

Router(config-controller)#g709 otu overhead tti ?
expected   Set expected TTI
sent       Set transmit TTI

Router(config-controller)#g709 otu overhead tti expected ?
ascii  Enter ASCII string
hex    Enter hex string- Length should be even number

Router(config-controller)#g709 otu overhead tti expected ascii ?
    WORD  LINE  ASCII text (Max 64 characters)

Router(config-controller)#g709 otu overhead tti expected hex ?
    Hex-data  LINE  Hex nibbles (Max 128- The string length should be an
            even number)

Router(config-controller)#g709 otu overhead tti sent ?
ascii  Enter ASCII string
hex    Enter hex string- Length should be even number

Router(config-controller)#g709 otu overhead tti sent ascii ?
WORD  LINE ASCII text (Max 64 characters)

Router(config-controller)#g709 otu overhead tti sent hex ?
Hex-data  LINE  Hex nibbles (Max 128- The string length should be an
            even number)

Router(config-controller)# g709 otu report ?
  ais     Set Alarm Indication Signal reporting status
  bdi     Set Backward Defect Indication reporting status
  iae     Set Incoming Alignment Error reporting status
  lof     Set OTU Loss of Frame reporting status
  lom     Set Loss of Multiple Frame reporting status
  los     Set Loss of Signal reporting status
  sm-tca  Set Section Monitoring BER TCA reporting status
  tim     Set Trace Identifier Mismatch reporting status

Router(config-controller)# g709 otu threshold ?
  sm-tca  Set Section Monitoring Threshold Crossing Alert threshold

Router(config-controller)# g709 otu threshold sm-tca ?
  <3-9>  Bit Error Rate (10 to the minus n) (default 3)
  <cr>
```

Enabling OTN Mode Alarms Assertion

By default, all the OTN mode alarms are enabled. To control OTN alarms, disable all the alarms and enable the specific alarms by performing the following steps. Standard FEC is the default FEC mode. Use the **show controller** command to verify the alarm status and FEC mode. Perform the steps detailed in the section to enable OTN mode alarm assertion. Configure same transport mode or FEC mode on both the routers. The FEC modes, standard and disable, are compatible with each other.

	Command or Action	Purpose
Step 1	`Router> `**`enable`**	Enables privileged EXEC mode (enter your password if prompted).
Step 2	`Router# `**`configure terminal`**	Enters global configuration mode.
Step 3	`Router(config)# `**`controller dwdm`** *`slot/port`*	Configures the DWDM controller.
Step 4	`Router(config-controller)# `**`shutdown`**	Shuts down the DWDM controller.
Step 5	`Router(config-controller)# {`**`g709`**` | `**`no g709`**`} `**`fec`**` {`**`disable`**` | `**`standard`**` | `**`enhanced`**`}`	Configures the FEC modes
Step 6	`Router(config-controller)# {`**`g709`**` | `**`no g709`**`} `**`otu report`**` {`**`ais`**` | `**`bdi`**` | `**`iae`**` | `**`lof`**` | `**`los`**` | `**`sm-tca`**` | `**`tim`**`}`	Specifies the supported otu alarms and configures the otu threshold. By default, all alarms are reported.
Step 7	`Router(config-controller)# {`**`g709`**` | `**`no g709`**`} `**`odu report`**` { `**`ais`**` | `**`bdi`**` | `**`lck`**` | `**`oci`**` | `**`pm-tca`**` | `**`ptim`**` | `**`sd-ber`**` | `**`sf-ber`**` | `**`tim`**` }`	Specifies the supported odu alarms and configures the odu threshold. By default, all the alarms are reported.
Step 8	`Router(config-controller)# {`**`g709`**` | `**`no g709`**`} `**`otu threshold sm-tca val`**	Set the threshold value to detect section monitoring signal degrade or signal failure alerts.
Step 9	`Router(config-controller)# {`**`g709`**` | `**`no g709`**`} `**`odu threshold`**` {`**`pm-tca`**` | `**`sd-ber`**` | `**`sf-ber`**`}` *`t_value`*	Sets the ber threshold limit to t_value power of ten.
Step 10	`Router(config-controller)# {`**`g709`**` | `**`no g709`**`} `**`otu overhead tti`**` {`**`expected`**` | `**`sent`**`} {`**`ascii`**` | `**`hex`**`}` *`tti-string`*	Specifies the trail trace identifier for otu level.
Step 11	`Router(config-controller)# {`**`g709`**` | `**`no g709`**`} `**`odu overhead tti`**` {`**`expected`**` | `**`sent`**`} {`**`ascii`**` | `**`hex`**`}` *`tti-string`*	Specifies the trail trace identifier for odu level.
Step 12	`Router(config-controller)# `**`no shutdown`**	Sets the controller to no shutdown mode.
Step 13	`Router(config-controller)# `**`end`**	Ends the session.

Note You need to shutdown the interface using **shutdown** command before changing the FEC mode to EFEC.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller dwdm 4/21
Router(config-controller)# shutdown
Router(config-controller)# g709 fec enhanced
Router(config-controller)# g709 otu report los
Router(config-controller)# no g709 otu report lof
Router(config-controller)# no g709 otu threshold sm-tca
Router(config-controller)# g709 odu threshold sd-ber 3
Router(config-controller)# no shutdown
Router(config-controller)# end
```

Use the **show controllers** command to verify the configuration for alarm assertion.

```
Router# show controllers dwdm 4/21
Controller dwdm 4/2, is up (no shutdown)

TAS state is : IS
G709 status : Enabled

OTU
        LOS = 1           LOF = 0           LOM = 0
```

```
        AIS = 0            BDI = 1            BIP = 0
        TIM = 0            IAE = 0            BEI = 0

ODU
        AIS = 0            BDI = 0            TIM = 0
        OCI = 0            LCK = 0            PTIM = 0
        BIP = 0            BEI = 0

FEC Mode: FEC
        EC(current second) = 0
        EC = 0             UC = 0
    pre-FEC BER < 9.00E-11
        Q > 6.45        Q Margin > 7.52  DBQ
Detected Alarms: NONE
Asserted Alarms: NONE
Detected Alerts: NONE
Asserted Alerts: NONE
Alarm reporting enabled for: LOS LOF LOM OTU-AIS OTU-IAE OTU-BDI ODU-AIS ODU-OCI ODU-LCK
ODU-BDI ODU-PTIM ODU-BIP Alert reporting enabled for: OTU-SM-TCA ODU-SD-BER ODU-SF-BER
ODU-PM-TCA BER thresholds: ODU-SF = 10e-3  ODU-SD = 10e-6 TCA thresholds: SM = 10e-3  PM =
10e-3

OTU TTI Sent     String ASCII: Tx TTI Not Configured
OTU TTI Received String ASCII:
OTU TTI Received String HEX  : 0000000000000000000000000000000000000000000000000
                               0000000000000000000000000000000000000000000000000
                               0000000000000000000000000000000 OTU TTI Expected String
ASCII: Exp TTI Not Configured

ODU TTI Sent     String ASCII: Tx TTI Not Configured
ODU TTI Received String ASCII:
ODU TTI Received String HEX  : 0000000000000000000000000000000000000000000000000
                               0000000000000000000000000000000000000000000000000
                               000000000000000000000000000000 ODU TTI Expected String
ASCII: Exp TTI Not Configured
```

Upgrading Field-Programmable Devices

Field-programmable devices (FPDs) support separate upgrades. This chapter includes the following sections:

Overview of FPD Images and Packages

An FPD image package is used to upgrade FPD images. Whenever a Cisco IOS image is released that supports the ES+ line cards, a companion FPD image package is also released for that Cisco IOS software release. The FPD image package is available from Cisco.com and is accessible from the Cisco Software Center page where you also go to download your Cisco IOS software image.

With ES+ line cards, when you upgrade the Cisco IOS image, you should download the FPD image package file before booting the router using the new Cisco IOS release. If the ES+ line cards requires an FPD upgrade and the Cisco IOS image is unable to locate an FPD image package, the system messages will indicate that the FPD image is incompatible and you will need to go to the Cisco Software Center on Cisco.com to download the FPD image package for your Cisco IOS software release. An FPD incompatibility on an ES+ line cards disables all interfaces on that ES+ line cards until the incompatibility is addressed.

ES+ line cards have these FPD-upgradeable components:

- Base board
  - ROMMON
  - I/O field-programmable Gate Array (FPGA)
  - Selene FPGA
- Link daughter card with a link FPGA
- DFC
  - Packet Engine FPGA
  - K' (k prime) FPGA

Note The FPD automatic upgrade feature only searches for the FPD image package file that is the same version number as the Cisco IOS release being used by the system. Ensure that the FPD image package file on your system is compatible with your Cisco IOS release and do not change the name of the FPD image package file.

FPD Quick Upgrade

This section provides information if you want to upgrade ES+ line card FPDs. These instructions are not always feasible for operating network environments and are not the only methods available for upgrading FPDs. This section addresses the following topics:

- FPD Quick Upgrade Before Upgrading your Cisco IOS Release (Recommended), page A-50
- FPD Quick Upgrade After Upgrading your Cisco IOS Release, page A-51

FPD Quick Upgrade Before Upgrading your Cisco IOS Release (Recommended)

Step 1 When getting your Cisco IOS image, download the FPD image package for the Cisco IOS release that you are upgrading to any Flash disk on your router before booting the new version of Cisco IOS. The FPD image package can be retrieved from the same site where you went to get your Cisco IOS image. Do not change the name of the FPD image package.

Step 2 Boot using the new version of Cisco IOS. When the new Cisco IOS boots, it by default searches for the FPD image package in the router flash file systems and the FPD images will be updated automatically as part of the IOS boot process.

FPD Quick Upgrade After Upgrading your Cisco IOS Release

Step 1 An FPD upgrade is not always necessary after Cisco IOS is reloaded. If you have already reloaded your Cisco IOS, enter the **show hw-module all fpd** command to see if all system FPDs are compatible. If the FPDs are compatible, no further action is necessary. If at least one FPD needs an upgrade, proceed to Step 2.

Step 2 Go to the cisco.com site where you downloaded your specific Cisco IOS software and locate the FPD image package.

Step 3 Download this FPD image package to a Flash disk on your router. Do not change the name of the FPD image package.

Do not change any FPD-related settings on your system (if **upgrade fpd auto** or **upgrade fpd path** has been changed, change the settings back to the default settings using the **no** form of the command). Reboot your Cisco IOS release software. When the new Cisco IOS boots, it by default searches for the FPD image package in the Flash file systems and the FPD images will be updated automatically as part of the IOS boot process.

Upgrading FPD Images

This section documents some of the common scenarios where FPD image updates are necessary. It discusses the following scenarios:

- Migrating to a Newer Cisco IOS Release, page A-51
- Upgrading FPD Images in a Non-Production System, page A-52

Migrating to a Newer Cisco IOS Release

This section discusses the following topics:

- Upgrading FPD Images Before Upgrading Cisco IOS Release (Recommended), page A-51
- Upgrade FPD Images after Upgrading the New Cisco IOS Release, page A-52

Upgrading FPD Images Before Upgrading Cisco IOS Release (Recommended)

If you are still running your old Cisco IOS Release but are preparing to load a newer version of Cisco IOS, upgrade FPD for the new Cisco IOS Release. Placing the FPD image package for the IOS release that you are upgrading to before upgrading IOS is the recommended method for upgrading FPD because it is simple in addition to being fast. To perform this type of FPD upgrade, follow these steps:

Step 1 While still running the Cisco IOS release that will be upgraded, place the FPD image package for the new version of Cisco IOS onto one of your router's Flash file systems. You can locate the FPD image package for a specific IOS release on cisco.com from the same area where you download that Cisco IOS software image. The switch and ES+ line cards should continue to operate normally since this action will have no impact on the current FPDs.

Caution Do not change the filename of the FPD image package file. The Cisco IOS searches for the FPD image package file by filename, so the FPD image package file cannot be found if it has been renamed.

Step 2 Reboot your router using the new upgraded Cisco IOS image. As part of the bootup process, the router will search for the FPD image package. Since the default settings for the FPD image package search are to check for the FPD image package for the specific Cisco IOS Release in a Flash file system, the FPD image package will be located during the bootup procedure and all FPDs that required upgrades will be upgraded.

Step 3 When the router has booted, verify the upgrade was successful by entering the **show hw-module all fpd** command.

Upgrade FPD Images after Upgrading the New Cisco IOS Release

The following steps explain how to upgrade FPD images if you have already upgraded your Cisco IOS release but still need to upgrade your FPD images.

To perform an FPD upgrade after the new Cisco release has been booted, follow these steps:

Step 1 If you are unsure if your FPD images for your ES+ line cards are compatible, enter the **show hw-module all fpd** command to verify compatibility of all ES+ line cards. If all of your ES+ line cards are compatible, there is no reason to perform this upgrade.

Step 2 If an FPD upgrade is necessary, place the FPD image package for the new version of Cisco IOS onto the router's Flash Disk or on an accessible FTP or TFTP server. You can locate the FPD image package on cisco.com from the same area where you downloaded your Cisco IOS software image.

Step 3 Enter the **upgrade hw-module** [**slot** *slot-number*] *file-url* command. The *file-url* command should direct users to the location of the FPD image package.

If multiple ES+ line cards require upgrades, the different pieces of hardware will have to be updated individually.

Note With the new Cisco IOS release running, if the ES+ line cards are disabled or powered down due to any FPD upgrade errors, the only way to do an FPD upgrade is by reloading the line card using **hw-module reset** command (assuming that you have already copied the necessary FPD bundle file in to the file system).The **upgrade hw-module** command works only when the line card is in the UP state.

Step 4 Verify the upgrade was successful by entering the **show hw-module all fpd** command.

Upgrading FPD Images in a Non-Production System

Adding an ES+ line cards to a production system presents the possibility that the ES+ line cards may contain versions of FPD images that are incompatible with the Cisco IOS release currently running the router. In addition, the FPD upgrade operation can be a very CPU-intensive operation and therefore the upgrade operation may take more time when it is performed on a production system. The performance impact will vary depending on various factors, including network traffic load, the type of processing engine used, type of ES+ line cards, and the type of service configured. Use a non-production system to upgrade the ES+ line card FPD image.

Before beginning the upgrade, ensure:

- The spare system is running the same version of the Cisco IOS software release that the target production system is running.
- The automatic upgrade feature is enabled on the spare system (the automatic upgrade feature is enabled by default. It can also be enabled using the **upgrade fpd auto** command).

Use the following procedure to perform an upgrade on a spare system:

Step 1 Download the FPD image package file to the router's flash file system or TFTP or FTP server accessible by the spare system. In most cases, it is preferable to place the file in a Flash file system since the router, by default, searches for the FPD image package in the Flash file systems. If the Flash file systems are full, use the **upgrade fpd path** command to direct the router to search for the FPD image package in the proper location.

Step 2 Insert the ES+ line card into the spare system.

If an upgrade is required, the system will perform the necessary FPD image updates so that when this ES+ line card is inserted to the target production system it will not trigger an FPD upgrade operation there.

Step 3 Verify the upgrade was successful by entering the **show hw-module all fpd** command.

Step 4 Remove the ES+ line card from the spare system after the upgrade.

Step 5 Insert the ES+ line card into the target production system.

Verifying System Compatibility

If a spare system is not available to perform an upgrade, you can check for system compatibility by disabling the automatic upgrade feature before inserting the ES+ line card (the automatic upgrade feature is enabled by default. It can be disabled using the **no upgrade fpd auto** command).

- If the FPD images on the ES+ line card are compatible with the system, you will only need to re-enable the automatic upgrade feature (the automatic upgrade feature can be re-enabled using the **upgrade fpd auto** command).
- If the FPD images on the ES+ line card are not compatible with the system, the ES+ line card is disabled but will not impact system performance by attempting to perform an automatic upgrade.

Use the following procedure to check the FPD images on the ES+ line card for system compatibility:

Step 1 Disable the automatic upgrade feature using the **no upgrade fpd auto** global configuration command.

Step 2 Insert the ES+ line card into the system.

If the FPD images are compatible, the ES+ line card will operate successfully after bootup.

If the FPD images are not compatible, the ES+ line card is disabled. At this point we recommend that you wait for a scheduled maintenance when the system is offline to manually perform the FPD upgrade using one of the procedures outlined in the "Upgrading FPD Images" section on page A-51.

Step 3 Re-enable the automatic upgrade feature using the **upgrade fpd auto** global configuration command.

Optional FPD Procedures

This section provides information for optional FPD-related functions. None of the topics discussed in this section are necessary for completing FPD upgrades, but may be useful in some FPD-related scenarios. It covers the following topics:

- Manually Upgrading ES+ Line Card FPD Images, page A-54
- Upgrading FPD from an FTP or TFTP Server, page A-54
- Modifying the Default Path for the FPD Image Package File Location, page A-55
- Displaying Current and Minimum Required FPD Image Versions, page A-56
- Displaying Information About the Default FPD Image Package, page A-56

Manually Upgrading ES+ Line Card FPD Images

To manually upgrade the current FPD version on an ES+ line card, use the following command:

```
Router# upgrade hw-module [slot slot-number] file file-url
```

In this example, *slot-number* is the slot where the ES+ line card is installed, *file-url* is the location and name of the FPD image package file.

Caution An image upgrade can require a long period of time to complete depending on the ES+ line card.

Upgrading FPD from an FTP or TFTP Server

The generally recommended method to perform an FPD image upgrade is to download the FPD image package to a Flash file system and use the FPD automatic upgrade. By default, the system searches the Flash file system for the FPD image package file when an FPD incompatibility is detected.

This default behavior of loading an FPD image from Flash can be changed using the **upgrade fpd path** global configuration command, which sets the path to search for the FPD image package file to a location other than the router's Flash file systems.

For large deployments where all the systems are being upgraded to a specific Cisco IOS software release, we recommend that the FPD image package file be placed on an FTP or TFTP server that is accessible to all the affected systems, and then use the **upgrade fpd path** global configuration command to configure the routers to look for the FPD image package file from the FTP or TFTP server prior to the reloading of the system with the new Cisco IOS release.

Note This approach can also be used if there is not enough disk space on the system Flash card to hold the FPD image package file.

To download an FPD image package file to an FTP or TFTP server, use the following procedure:

Step 1 Copy the FPD image package file to the FTP or TFTP server.

Step 2 From global configuration mode, use the **upgrade fpd path** command to instruct the router to locate the FPD image package file from the FTP or TFTP server location.

For example, enter one of the following global configuration commands from the target system's console:

```
Router(config)# upgrade fpd path tftp://my_tftpserver/fpd_pkg_dir/
```

or

```
Router(config)# upgrade fpd path ftp://login:password@my_ftpserver/fpd_pkg_dir/
```

Note The final "/" at the end of each of the above examples is required. If the path is specified without the trailing "/" character, the command will not work properly.

In these examples, *my_tftpserver* or *my_ftpserver* is the path to server name, *fpd_pkg_dir* is the directory on the TFTP server where the FPD image package is located, and *login:password* is your FTP login name and password.

Step 3 Make sure that the FPD automatic upgrade feature is enabled by examining the output of the **show running-config** command. (Look for the *upgrade fpd auto* configuration line in the output. If there are no upgrade commands in the output, then **upgrade fpd auto** is enabled because it is the default setting.) If automatic upgrades are disabled, use the **upgrade fpd auto** global configuration command to enable automatic FPD upgrades.

Step 4 Enter the **show upgrade fpd file** command to ensure your router is connecting properly to the default FPD image package. If you are able to generate output related to the FPD image package using this command, the upgrade should work properly.

Step 5 Save the configuration and reload the system with the new Cisco IOS release.

During the system startup after the reload, the necessary FPD image version check for all the ES+ line cards will be performed and any upgrade operation will occur automatically if an upgrade is required. In each upgrade operation, the system extracts the necessary FPD images to the ES+ line card from the FPD image package file located on the FTP or TFTP server.

Modifying the Default Path for the FPD Image Package File Location

By default, the Cisco IOS software looks for the FPD image package file on a Flash file system when performing an automatic FPD image upgrade.

Note Be sure there is enough space on one of your Flash file systems to accommodate the FPD image package file.

Alternatively, you can store an FPD image package file elsewhere. However, because the system looks on the Flash file systems by default, you need to change the FPD image package file location so that the system is directed to search an alternate location (such an FTP or TFTP server) that is accessible by the Cisco IOS software. Enter the **upgrade fpd path** *fpd-pkg-dir-url* global configuration command, where *fpd-pkg-dir-url* is the alternate location, to instruct the router to search for the FPD image package elsewhere.

When specifying the *fpd-pkg-dir-url*, be aware of the following:

- The *fpd-pkg-dir-url* is the path to the FPD image package, but do not include the FPD image package as part of *fpd-pkg-dir-url*. Enter:

  ```
  upgrade fpd path tftp://mytftpserver/myname/myfpdpkgpath/
  ```

  The filename is not specified.
- The final "/" character in the *fpd-pkg-dir-url* is required.

If the **upgrade fpd path** global configuration command has not been entered to direct the router to locate an FPD image package file in an alternate location, the system searches the Flash file systems on the switch for the FPD image package file.

Failure to locate an FPD image package file when an upgrade is required will disable the ES+ line card. Because ES+ line cards will not come online until FPD is compatible, the ES+ line card will also be disabled if it requires an FPD upgrade and the automatic upgrade feature is disabled.

Displaying Current and Minimum Required FPD Image Versions

To display the current version of FPD images on the ES+ line cards installed on your router, use the **show hw-module** [*slot-number* | **all**] **fpd** command, where *slot-number* is the slot number where the ES+ line card is installed. Entering the **all** keyword shows information for hardware in all router slots.

The following examples show the output when using this **show** command.

The output display in this example shows that FPD versions on the ES+ line cards in the system meet the minimum requirements:

```
Router# show hw-module all fpd
```

This example shows the output when verifying the FPD for the ES+ line card in a specific slot:

```
Router# show hw-module slot 9 fpd
```

Displaying Information About the Default FPD Image Package

You can use the **show upgrade fpd package default** command to find out which ES+ line cards are supported with your current Cisco IOS release and which FPD image package you need for an upgrade.

```
Router# show upgrade fpd package default
```

FPD Image Upgrade Examples

This section provides examples of automatic and manual FPD image upgrades. It includes the following examples:

- Automatic FPD Image Upgrade Example, page A-56
- Manual FPD Image Upgrade Example, page A-57

Automatic FPD Image Upgrade Example

The following example uses the **upgrade fpd auto to do an automatic upgrade**.

```
Router# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# upgrade fpd ?
  auto  Auto upgrade all FPD images
  path  Set path to locate the FPD image package file for auto upgrade

Router(config)# upgrade fpd auto ?
  <cr>

Router(config)# upgrade fpd auto
Router(config)# exit
Router# show version
*Jun 18 10:27:00.078 sum08: %SYS-5-CONFIG_I: Configured from console by consoh ver
```

```
Cisco IOS Software, rsp72043_rp Software (rsp72043_rp-ADVENTERPRISEK9_DBG-M), Version
12.2(nightly.SR080616) NIGHTLY BUILD, synced to rainier
RAINIER_BASE_FOR_V122_33_SRA_THROTTLE
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Tue 17-Jun-08 00:10 by cuotran

ROM: System Bootstrap, Version 12.2(33r)SRB3, RELEASE SOFTRouterRE (fc1)

 Router uptime is 22 hours, 29 minutes
Uptime for this control processor is 22 hours, 29 minutes
System returned to ROM by reload (SP by reload)
System image file is "disk0:rsp72043-adventerprisek9_dbg-mz.autobahn76_061608"
Last reload type: Normal Reload
```

Manual FPD Image Upgrade Example

In the following example, FPD for the ES+ line card in slot 8 is upgraded manually:

```
Router# upgrade hw-module slot 8 ?
  fpd  Field programmable device upgrade option
Router# upgrade hw-module slot 8 fpd ?
  file  Upgrade with field programmable device package/bundle file

Router# upgrade hw-module slot 8 fpd file c
Router# upgrade hw-module slot 8 fpd file d
*Jun 17 13:24:12.531 sum08: %FPD_MGMT-3-INCOMP_IMG_VER: Incompatible I/O FPGA (FPD ID=2)
image version detected for 7600-ES+40G3CXL card in slot 8. Detected version = 0.16,
minimum required version = 0.17. Current HW version = 0.118.
*Jun 17 13:24:12.531 sum08: %FPD_MGMT-3-INCOMP_IMG_VER: Incompatible 40x1G LinkFPGA (FPD
ID=6) image version detected for 7600-ES+40G card in slot-dc 8-2. Detected version = 0.14,
minimum required version = 0.15. Current HW version = 0.106.
*Jun 17 13:24:12.531 sum08: %FPD_MGMT-5-UPGRADE_ATTEMPT: Attempting to automatically
upgrade the FPD image(s) for 7600-ES+40G3CXL card in slot 8. Use 'show upgrade fpd
progress' command to view the upgrade progress ...
*Jun 17 13:24:12.547 sum08: %FPD_MGMT-6-BUNDLE_DOWNLOAD: Downloading FPD image bundle for
7600-ES+40G3CXL card in slot 8 ...i
Router#upgrade hw-module slot 8 fpd file disk
*Jun 17 16:24:12.551: %FABRIC_INTF_ASIC-DFC8-5-FABRICSYNC_DONE: Fabric ASIC 0 Channel 1:
Fabric sync done.
*Jun 17 16:24:12.575: %FABRIC_INTF_ASIC-DFC8-5-FABRICSYNC_DONE: Fabric ASIC 1 Channel 1:
Fabric sync done.
```

Troubleshooting

These sections describe techniques that you can use to troubleshoot the operation of ES+ line cards.

- Using the Cisco IOS Event Tracer to Troubleshoot Problems, page A-58
- Troubleshooting XFP Issues, page A-58
- Preparing for Online Insertion and Removal of ES+ Line Cards, page A-59

The first section provides information about basic interface troubleshooting. If you are having a problem with your XFP transceivers, use the steps in the "Using the Cisco IOS Event Tracer to Troubleshoot Problems" section to begin your investigation of a possible interface configuration problem.

Using the Cisco IOS Event Tracer to Troubleshoot Problems

Note The Event Tracer feature is intended for use as a software diagnostic tool and should be configured only under the direction of a Cisco Technical Assistance Center (TAC) representative.

The Event Tracer feature provides a binary trace facility for troubleshooting Cisco IOS software. This feature gives Cisco service representatives additional insight into the operation of the Cisco IOS software and can be useful in helping to diagnose problems in the unlikely event of an operating system malfunction or, in the case of redundant systems, Route Processor switch over.

Event tracing works by reading informational messages from specific Cisco IOS software subsystem components that have been pre-programmed to work with event tracing, and by logging messages from those components into system memory. Trace messages stored in memory can be displayed on the screen or saved to a file for later analysis.

For more information about using the Event Tracer feature, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/netmgmt/configuration/guide/nm_event_tracer.html

Troubleshooting XFP Issues

Use the following commands when troubleshooting XFP transceiver issues:

Command	Purpose
Router# **show interfaces** [*interface interface-number*] **capabilities** [**module** *number*]	Displays the interface capabilities for a module, an interface, or all interfaces.
Router# **show interfaces** [*interface interface-number*] **status** [**err-disabled** | **module** *number*]	Displays the interface status.
Router# **show interfaces** [*interface interface-number*] **transceiver** [**threshold violations**] [**detail** | {**module** *number*}]	Displays information about the optical transceivers that have digital optical monitoring (DOM) enabled
Router# **show idprom interface**	Displays IDPROMs for the transceiver plugged into the port.

Make sure the optics power rating are suitable for the link. For both ends of a link, check for "+", "++", "-", "--" in the output the following command to confirm that the power ratings are correct:

```
Router# show interface tengigabitethernet 2/1 transceiver detail
Transceiver monitoring is disabled for all interfaces.

ITU Channel not available (Wavelength not available),
Transceiver is internally calibrated.
mA: milliamperes, dBm: decibels (milliwatts), NA or N/A: not applicable.
++ : high alarm, +  : high warning, -  : low warning, -- : low alarm.
A2D readouts (if they differ), are reported in parentheses.
The threshold values are calibrated.

                            High Alarm  High Warn  Low Warn   Low Alarm
           Temperature       Threshold   Threshold  Threshold  Threshold
Port       (Celsius)         (Celsius)   (Celsius)  (Celsius)  (Celsius)
---------- ------------------ ----------  ---------  ---------  ---------
Te2/1        29.6               74.0        70.0         0.0       -4.0

                            High Alarm  High Warn  Low Warn   Low Alarm
```

```
           Voltage              Threshold  Threshold  Threshold  Threshold
Port       (Volts)              (Volts)    (Volts)    (Volts)    (Volts)
---------- ---------------      ---------- ---------  ---------  ---------
Te2/1       N/A                    N/A        N/A        N/A        N/A

                             High Alarm  High Warn  Low Warn   Low Alarm
           Current              Threshold  Threshold  Threshold  Threshold
Port       (milliamperes)       (mA)       (mA)       (mA)       (mA)
---------- -----------------    ---------- ---------  ---------  ---------
Te2/1       56.8          --      N/A        N/A        N/A       N/A

           Optical              High Alarm High Warn  Low Warn   Low Alarm
           Transmit Power       Threshold  Threshold  Threshold  Threshold
Port       (dBm)                (dBm)      (dBm)      (dBm)      (dBm)
---------- -----------------    ---------- ---------  ---------  ---------
Te2/1        1.6                   7.9        3.9        0.0       -4.0

           Optical              High Alarm High Warn  Low Warn   Low Alarm
           Receive Power        Threshold  Threshold  Threshold  Threshold
Port       (dBm)                (dBm)      (dBm)      (dBm)      (dBm)
---------- -----------------    ---------- ---------  ---------  ---------
Te2/1       -1.2          ++      -3.0       -7.0       -24.0      -28.2
```

If there is an alarm or warning, verify the following:

- Both ends of a link must use the same tranceiver type.
- The tranceivers (LR, IR, ZR) must be appropriate for the length of the link.
- Use an attenuator to control the power ratings.

Preparing for Online Insertion and Removal of ES+ Line Cards

Online insertion and removal (OIR) supports ES+ line cards, in addition to each of the XFP transceivers.

You can remove an ES+ line card with its transceivers still intact, or you can remove an transceiver independently from the ES+ line card, leaving the ES+ line card installed.

This section includes the following topics on OIR support:

- Preparing for Online Removal of an ES+ Line Card, page A-59
- Verifying Deactivation and Activation of an ES+ Line Card, page A-60
- Deactivation and Activation Configuration Examples, page A-61

Preparing for Online Removal of an ES+ Line Card

To do an OIR, you can power down an ES+ line card (which automatically deactivates any installed optical transceivers) and remove the ES+ line card still intact.

Although graceful deactivation of an ES+ line card is preferred using the **no power enable module** command, the Cisco 7600 series router does support removal of the ES+ line card without deactivating it first. If you plan to remove an ES+ line card, you can deactivate the ES+ line card first, using the **no power enable module** global configuration command. When you deactivate an ES+ line card using this command, it automatically deactivates each of the optical transceivers that are installed in that ES+ line card. Therefore, it is not necessary to deactivate each of the optical transceivers prior to deactivating the ES+ line card.

Either a blank filler plate or a functional optical transceiver should reside in every subslot of an ES+ line card during normal operation.

Deactivating an ES+ Line Card

To deactivate an ES+ line card and its installed optical transceivers prior to removal of the line card, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`no power enable module`** *`slot`*	Shuts down any installed interfaces, and deactivates the ES+ line card in the specified slot.

Reactivating an ES+ Line Card

When you deactivate an ES+ line card, whether or not you have performed an OIR, you must use the **power enable module** global configuration command to reactivate the ES+ line card.

If you did not issue a command to deactivate the optical transceivers installed in an ES+ line card, but you did deactivate the ES+ line card using the **no power enable module** command, then you do not need to reactivate the optical transceivers after an OIR of the ES+ line card. The installed optical transceivers automatically reactivate upon reactivation of the ES+ line card in the router.

For example, consider the case in which you remove an ES+ line card from the router to replace it with another ES+ line card. You reinstall the same optical transceivers into the new ES+ line card. When you enter the **power enable module** command on the router, the optical transceivers will automatically reactivate with the new ES+ line card.

To activate an ES+ line card and its installed optical transceivers after the ES+ line card has been deactivated, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`power enable module`** *`slot`*	Activates the ES+ line card in the specified slot and its installed optical transceivers, where: • *slot*—Specifies the chassis slot number where the ES+ line card is installed.

Verifying Deactivation and Activation of an ES+ Line Card

To verify the deactivation of an ES+ line card, enter the **show module** command in privileged EXEC configuration mode. Observe the Status field associated with the ES+ line card that you want to verify.

The following example shows that the ES+ line card located in slot 10 is deactivated. This is indicated by its “PwrDown” status.

```
Router# show module 10
```

To verify activation and proper operation of an ES+ line card, enter the **show module** command and observe “Ok” in the Status field as shown in the following example:

```
Router# show module 10
```

Deactivation and Activation Configuration Examples

This section provides the following examples of deactivating and activating an ES+ line card and optical transceivers:

- Deactivation of an ES+ Line Card Configuration Example, page A-61
- Activation of an ES+ Line Card Configuration Example, page A-61

Deactivation of an ES+ Line Card Configuration Example

Deactivate an ES+ line card when you want to perform OIR of the ES+ line card. The following example deactivates the ES+ line card that is installed in slot 5 of the router, its optical transceivers, and all of the interfaces. The corresponding console messages are shown:

```
Router# configure terminal
Router(config)# no power enable module 5
1w4d: %OIR-6-REMCARD: Card removed from slot 5, interfaces disabled
1w4d: %C6KPWR-SP-4-DISABLED: power to module in slot 5 set off (admin request)
```

Activation of an ES+ Line Card Configuration Example

Activate an ES+ line card if you have previously deactivated it. If you did not deactivate the optical transceivers, the optical transceivers automatically reactivate with reactivation of the ES+ line card.

The following example activates the ES+ line card that is installed in slot 5 of the router, its optical transceivers, and all of the interfaces (as long as the **hw-module subslot shutdown** command was not issued to also deactivate the optical transceivers):

```
Router# configure terminal
Router(config)# power enable module 5

Notice that there are no corresponding console messages shown with activation. If you
re-enter the power enable module command, a message is displayed indicating that the
module is already enabled:

Router(config)# power enable module 5
% module is already enabled
```

APPENDIX B

Online Diagnostic Tests

Note

- For information about configuring online diagnostic tests see Chapter 12, "Configuring Online Diagnostics."
- Before you enable any online diagnostics tests, enable console logging to see all warning messages.
- We recommend that when you are running disruptive tests that you only run the tests when connected through console. When disruptive tests are complete a warning message on the console recommends that you reload the system to return to normal operation: strictly follow this warning.
- While tests are running, all ports are shut down as a stress test is being performed with looping ports internally and external traffic might affect the test results. The switch must be rebooted to bring the switch to normal operation. When you issue the command to reload the switch, the system will ask you if the configuration should be saved.
- Do not save the configuration.
- If you are running the tests on other modules, after the test is initiated and complete, you must reset the module.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Global Health-Monitoring Tests

TestAsicSync

This test periodically verifies the status of bus and port synchronization ASICs.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXH.
Corrective action:	Reset the module. After the module has ten consecutive failures or three consecutive resets, it powers down.
Hardware support:	All modules.

TestEARLInternalTables

This test detects most PFC and DFC hardware table problems by running consistency checks on the PFC and DFC hardware tables. The test runs every 5 minutes.

A failure of the test for the PFC results in one of these actions:

- Failover to the redundant supervisor engine.
- If a redundant supervisor engine is not installed, shutdown of the supervisor engine.

A failure of the test for the DFC results in one of these actions:

- Up to two resets of the DFC-equipped switching module.
- Shutdown following a third failure.

A CallHome message is generated if CallHome is configured on the system.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXI2.
Corrective action:	Reset the affected module.
Hardware support:	PFC and DFCs.

TestErrorCounterMonitor

This test monitors the errors and interrupts that occur on each module in the system by periodically polling for the error counters maintained in the module. If the errors exceed a threshold value, a syslog message is displayed with detailed information including the error-counter identifier, port number, total failures, consecutive failures, and the severity of the error counter.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable. This test is automatically disabled during CPU-usage spikes to maintain accuracy.
Default:	On.
Intitial Release:	12.2(33)SXH.
Corrective action:	Display a syslog message indicating the error-counters detected on that port.
Hardware support:	All modules.

TestIntPortLoopback

This test uses the switching module internal port to run a non-disruptive loopback test. It can be used to detect fabric channel failure and also port ASIC failure. This test is similar to TestFabricCh0Health. The test runs every 15 seconds.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not turn this test off. Use as a health-monitoring test.
Default:	On.
Intitial Release:	12.2(33)SXH.
Corrective action:	The module resets after 10 consecutive failures. Three consecutive resets powers down the module.
Hardware support:	WS-X6148E-GE-45AT, WS-X6148A-GE-TX, WS-X6148A-GE-45AF, WS-X6148-FE-SFP, WS-X6148A-RJ-45, WS-X6148A-45AF, WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6548-GE-45AF.

TestLtlFpoeMemoryConsistency

This test verifies that the LTL and FPOE memories are working properly. The test runs every 15 seconds. Self-correction is applied if an error is detected. If self-correction fails, corrective action is triggered to reset the module. The module is powered-down on the third consecutive module reset. If self-correction passes, no action is taken. If too many self-corrections occur within a short period of time (more than three self-corrections in less than 300 seconds), the module is reset.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXI2.
Corrective action:	Failure of this test causes the module to reset and power down after two resets.
Hardware support:	WS-SVC-ASA-SM1-K9 and WS-SVC-NAM3-6G-K9.

TestMacNotification

This test verifies the data and control path between DFC-equipped modules and supervisor engines. This test also ensures Layer 2 MAC address consistency across Layer 2 MAC address tables. The test runs every six seconds. Ten consecutive failures causes the module to reset during bootup or runtime (default). After three consecutive resets, the module powers down. This test runs every 15 seconds.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(14)SX.
Corrective action:	Reset the module. After the module has ten consecutive failures or three consecutive resets, it powers down.
Hardware support:	DFC-equipped modules.

TestPortTxMonitoring

This test periodically polls the transmit counters on each port. The test displays a syslog message and error disables the port if no activity is seen for the configured time interval and failure threshold. You configure the time interval and threshold by entering the **diagnostic monitor interval** and **diagnostic monitor threshold** commands. The test does not source any packets, but leverages the CDP protocol that transmits packets periodically. If the CDP protocol is disabled, the polling for that port is not performed. The test runs every 75 seconds, and the failure threshold is set to five by default.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXH.
Corrective action:	Display a syslog message indicating the port(s) that failed. Error disable the port(s) that failed.
Hardware support:	All modules including supervisor engines.

TestScratchRegister

This test monitors the health of application-specific integrated circuits (ASICs) by writing values into registers and reading back the values from these registers. The test runs every 30 seconds. Five consecutive failures causes a supervisor engine to switchover (or reset), if you are testing the supervisor engine, or in the module powering down when testing a module.

Attribute	**Description**
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(14)SX.
Corrective action:	Reset the malfunctioning supervisor engine or power down the module.
Hardware support:	Active and standby supervisor engine, DFC-equipped modules, WS-X6148A-GE-TX, WS-X6148A-GE-45AF, WS-X6148-FE-SFP, WS-X6148A-RJ-45, WS-X6148A-45AF.

TestSnrMonitoring

This test monitors the SNR (signal-to-noise ratio) margin for a port, which varies between -12.7 dB to +12.7 dB. The test uses the following two threshold levels to compare SNR:

- Minor threshold at +1.0 dB
- Major threshold at 0.0 dB

When the SNR value drops below the minor threshold, the test logs a minor warning message. When the SNR value drops below the major threshold, the test logs a major warning message. Similarly, recovery messages are logged when SNR recovers the two threshold levels. The default interval for the test is 30 seconds and can be configured to as low as 10 seconds for faster monitoring. The TestSnrMonitoring is not a bootup test and cannot be run on demand.

Attribute	**Description**
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXI4.
Corrective action:	None.
Hardware support:	WS-X6716-10T.

TestSPRPInbandPing

This test detects most runtime software driver and hardware problems on supervisor engines by running diagnostic packet tests using the Layer 2 forwarding engine, the Layer 3 and 4 forwarding engine, and the replication engine on the path from the switch processor to the route processor. Packets are sent at 15-second intervals. Ten consecutive failures of the test results in failover to the redundant supervisor engine (default) or reload of the supervisor engine if a redundant supervisor engine is not installed.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable. This test is automatically disabled during CPU-usage spikes in order to maintain accuracy.
Default:	On.
Intitial Release:	12.2(14)SX.
Corrective action:	Reset the active supervisor engine.
Hardware support:	Active and standby supervisor engine.

TestUnusedPortIndexDirect

This test periodically verifies the data path between the supervisor engine and the network ports of a module in the runtime. In this test, a Layer 2 packet is index-directed to the test port from the supervisor's inband port. The packet is looped back in the test port and index-directed back to the supervisor's inband port. It's similar to TestPortIndexDirect but only runs on unused (admin down) network ports and only one unused port per port ASIC. This test substitutes the lack of a nondisruptive loopback test in current ASICs. This test runs every 60 seconds.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable. This test is automatically disabled during CPU-usage spikes to maintain accuracy.
Default:	On.
Intitial Release:	12.2(33)SXH.
Corrective action:	Display a syslog message indicating the port(s) that failed. For modules other than the supervisor engines, if all port groups fail (for example, at least one port per port ASIC fails more than the failure threshold for all port ASICs), the default action is to reset the module and power down the module after two resets.
Hardware support:	All modules including the supervisor engines.

TestUnusedPortLoopback

This test periodically verifies the data path between the supervisor engine and the network ports of a module in the runtime. In this test, a Layer 2 packet is flooded onto the VLAN associated with the test port and the inband port of the supervisor engine. The packet loops back in the test port and returns to the supervisor engine on the same VLAN. This test is similar to TestLoopback but only runs on unused (admin down) network ports and on only one unused port per port ASIC. This test substitutes the lack of a nondisruptive loopback test in current ASICs. This test runs every 60 seconds.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable. This test is automatically disabled during CPU-usage spikes to maintain accuracy.
Default:	On.
Intitial Release:	12.2(33)SXH.
Corrective action:	Display a syslog message indicating the port(s) that failed. For modules other than the supervisor engines, if all port groups fail (for example, at least one port per port ASIC fails more than the failure threshold for all port ASICs), the default action is to reset the module and power down the module after two resets.
Hardware support:	All modules including the supervisor engines.

Per-Port Tests

TestActiveToStandbyLoopback

This test verifies the data path between the active supervisor engine and the network ports of the standby supervisor engine. In this test, a Layer 2 packet is flooded onto a VLAN that consists of only the test port and the active supervisor engine's inband port. The test packets are looped back in the targeted port and are flooded back onto the bus with only the active supervisor engine's inband port listening in on the flooded VLAN.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the loopback port (for example, Spanning Tree Protocol).
Recommendation:	Schedule during downtime.
Default:	Runs at bootup or after OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	Error disable a port if the loopback test fails on the port. Reset the standby supervisor engine if all of the ports fail.
Hardware support:	Standby supervisor engine only.

TestCCPLoopback

This test checks the control plane data path. This test sends an online diagnostics packet from the supervisor engine to service or high availability port on the Wireless Services Module (WiSM2). The TestCCPLoopback checks whether the test packet loops back. If the test fails, a syslog message is displayed to indicate the error. This test also can be run as health monitoring, on-demand, and scheduled tests.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXJ.
Corrective action:	A syslog message is displayed after five consecutive failures.
Hardware support:	WS-SVC-WISM2-K9.

TestDataPortLoopback

This test sends a packet from the inband port of the supervisor to the data port on the Firewall or NAM service module to verify the data packet path. The packet is looped back to the supervisor in hardware. If the packet does not return from the supervisor, hardware counters are polled to isolate the faulty path. This test runs every 45 seconds.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable. If the test fails for 10 consecutive times, themodule is reset. If the test fails persistently, the module is powered down.
Default:	On.
Intitial Release:	12.2(33)SXJ.
Corrective action:	None.
Hardware support:	WS-SVC-ASA-SM1-K9 and WS-SVC-NAM3-6G-K9.

TestDCPLoopback

This test checks the data plane data path. This test sends an online diagnostics packet from the supervisor engine to data ports on the Wireless Services Module (WiSM2). This test checks whether the test packet loops back. If the test fails, a syslog message is displayed to indicate the error. This test also can be run as health monitoring, on-demand, and scheduled tests.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXJ.
Corrective action:	A syslog message is displayed after five consecutive failures.
Hardware support:	WS-SVC-WISM2-K9.

TestLoopback

This test verifies the data path between the supervisor engine and the network ports of a module. In this test, a Layer 2 packet is flooded onto a VLAN that consists of only the test port and the supervisor engine's inband port. The packet loops back in the port and returns to the supervisor engine on that same VLAN.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of looped-back port (for example, Spanning Tree Protocol).
Recommendation:	Schedule during downtime.
Default:	Runs at bootup or after online insertion and removal (OIR).
Intitial Release:	12.2(14)SX.
Corrective action:	Error disable a port if the loopback test fails on the port. Reset the module if all of the ports fail.
Hardware support:	All modules including supervisor engines.

TestMgmtPortsLoopback

This test sends a packet from the inband port of the supervisor to the Firewall or NAM service module to verify the health of the backplane ports. The packet is looped back to the supervisor in hardware. If the packet does not return from the supervisor, the service application is queried for the status of the packet and depending on the action suggested by the service module, a syslog message is displayed and the module is reset. This test runs every 30 seconds.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable. If the failure is isolated to the firewall module, then a syslog is printed indicating which port failed the test. If the test fails due to any other datapath issue for 10 consecutive times, the module is reset. If the test fails persistently, the module is powered down.
Default:	On.
Intitial Release:	12.2(33)SXJ.
Corrective action:	None.
Hardware support:	WS-SVC-ASA-SM1-K9 and WS-SVC-NAM3-6G-K9.

TestNetflowInlineRewrite

This test verifies the NetFlow lookup operation, the ACL permit and deny functionality, and the inline rewrite capabilities of the port ASIC. The test packet will undergo a NetFlow table lookup to obtain the rewrite information. The VLAN and the source and destination MAC addresses are rewritten when the packet reaches the targeted port.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on configuration of loopback port (for example, Spanning Tree Protocol).
Recommendation:	Schedule during downtime. Run this test during bootup only.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	All modules including supervisor engines.

TestNonDisruptiveLoopback

This test verifies the data path between the supervisor engine and the network ports of a module. In this test, a Layer 2 packet is flooded onto VLAN that contains a group of test ports. The test port group consists of one port per port ASIC channel. Each port in the test port group nondisruptively loops back the packet and directs it back to the supervisor engine's inband port. The ports in the test port group are tested in parallel.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(18)SXF.
Corrective action:	Error disable a port after 10 consecutive failures. Error disable a channel if all of its ports failed the test in one test cycle. Reset the module after a failure of all channels.
Hardware support:	WS-X6148-FE-SFP, WS-X6148A-GE-TX, WS-X6148A-RJ-45.

TestNPLoopback

This test checks the data path of the ACE30 module for data path errors. This test runs at bootup, and the default configuration is a health-monitoring test that runs every 15 seconds. If TestNPLoopback fails, an SCP (Switch-module Configuration Protocol) message is sent to the ACE30 module indicating which network processors have failed. Upon receipt of the SCP message, ACE30 will take corrective action. If the TestNPLoopback test fails for ten consecutive times, the ACE30 module is reset.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXJ.
Corrective action:	A syslog message is displayed to inform the ACE30 about the port(s) that failed the test on the failure code. Depending on the failure code, the ACE30 decides whether to take corrective action or not. The suggested action for ACE30 is to collect core dumps from all network processors and reset the ACE30 module.
Hardware support:	ACE30-MOD-K9.

TestPortIndexDirect

This test verifies the data path between the supervisor engine and the network ports of a module. In this test, a Layer 2 packet is index-directed to the test port from the supervisor engine inband port. The packet is looped back in the test port and index-directed back to the supervisor engine inband port.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	Schedule during downtime.
Default:	Off.
Intitial Release:	12.2(33)SXH.
Corrective action:	Error disable the port.
Hardware support:	All modules including supervisor engines.

TestTransceiverIntegrity

This security test is performed on the transceiver during transceiver online insertion and removal (OIR) or module bootup to make sure that the transceiver is supported.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Not applicable.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	Error disable the port.
Hardware support:	All modules with transceivers.

PFC Layer 2 Tests

TestBadBpduTrap

This test is a combination of the TestTrap and the TestBadBpdu tests, which are described in the "DFC Layer 2 Tests" section on page B-16.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	If you experience problems with the Layer 2 forwarding engine learning capability, run this test on-demand to verify the Layer 2 learning functionality. This test can also be used as a health-monitoring test.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines only.

TestDontConditionalLearn

This test is a combination of the TestDontLearn and the TestConditionalLearn tests, which are described in the "DFC Layer 2 Tests" section on page B-16.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	If you experience problems with the Layer 2 forwarding engine learning capability, run this test on-demand to verify the Layer 2 learning functionality. This test can also be used as a health monitoring test.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestMatchCapture

This test is a combination of the TestProtocolMatchChannel and the TestCapture tests, which are described in the "DFC Layer 2 Tests" section on page B-16.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Run this test on-demand to verify the Layer 2 learning functionality. This test can also be used as a health-monitoring test.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines only.

TestNewIndexLearn

This test is a combination of the TestNewLearn and the TestIndexLearn tests, which are described in the "DFC Layer 2 Tests" section on page B-16.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	If you experience problems with the Layer 2 forwarding engine learning capability, run this test on-demand to verify the Layer 2 learning functionality. This test can also be used as a health-monitoring test.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines only.

DFC Layer 2 Tests

TestBadBpdu

This test verifies the ability to trap or redirect packets to the switch processor. This test verifies that the Trap feature of the Layer 2 forwarding engine is working properly. When running the test on the supervisor engine, the diagnostic packet is sent from the supervisor engine's inband port and performs a packet lookup using the supervisor engine's Layer 2 forwarding engine. For DFC-equipped modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The BPDU feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestCapture

This test verifies that the capture feature of Layer 2 forwarding engine is working properly. The capture functionality is used for multicast replication. When running the test on the supervisor engine, the diagnostic packet is sent from the supervisor engine's inband port and performs a packet lookup using the supervisor engine's Layer 2 forwarding engine. For DFC-equipped modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Capture feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	Schedule during downtime.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestConditionalLearn

This test verifies the ability to learn a Layer 2 source MAC address under specific conditions. When running the test on the supervisor engine, the diagnostic packet is sent from the supervisor engine's inband port and performs a packet lookup using the supervisor engine Layer 2 forwarding engine. For DFC-equipped modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Conditional Learn feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestDontLearn

This test verifies that new source MAC addresses are not populated in the MAC address table when they should not be learned. This test verifies that the "don't learn" feature of the Layer 2 forwarding engine is working properly. For DFC-equipped modules, the diagnostic packet is sent from the supervisor engine inband port through the switch fabric and looped back from one of the ports on the DFC-enabled module. The "don't learn" feature is verified during diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	Schedule during downtime.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestIndexLearn

This test ensures that existing MAC address table entries can be updated. This test verifies the Index Learn feature of the Layer 2 forwarding engine is working properly. When running the test on the supervisor engine, the diagnostic packet is sent from the supervisor engine's inband port and performs a packet lookup using the supervisor engine Layer 2 forwarding engine. For DFC-equipped modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Index Learn feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestNewLearn

This test verifies the Layer 2 source MAC address learning functionality of the Layer 2 forwarding engine. For supervisor engines, a diagnostic packet is sent from the supervisor engine inband port to verify that the Layer 2 forwarding engine is learning the new source MAC address from the diagnostic packet. For DFC-equipped modules, a diagnostic packet is sent from the supervisor engine inband port through the switch fabric and looped backed from one of the ports on the DFC-enabled module. The Layer 2 learning functionality is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestPortSecurity

This test verifies the ability to redirect packets to the CPU if a secure MAC address is transmitting the packets from a different port. For the supervisor engine, a diagnostic packet is sent from the supervisor engine's inband port and the port security feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine. For DFC-equipped modules, a diagnostic packet is sent from the supervisor engine inband port through the fabric and is looped back in one of the ports on the DFC-equipped module. The port security feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	None.
Default:	Off.
Initial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestProtocolMatchChannel

This test verifies the ability to match specific Layer 2 protocols in the Layer 2 forwarding engine. When running the test on the supervisor engine, the diagnostic packet is sent from the supervisor engine's inband port and performs a packet lookup using the supervisor engine's Layer 2 forwarding engine. For DFC-equipped modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Match feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Initial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestStaticEntry

This test verifies the ability to populate static entries in the Layer 2 MAC address table. For DFC-equipped modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Static Entry feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

TestTrap

This test verifies the ability to trap or redirect packets to the switch processor. This test verifies that the Trap feature of the Layer 2 forwarding engine is working properly. When running the test on the supervisor engine, the diagnostic packet is sent from the supervisor engine's inband port and performs a packet lookup using the supervisor engine's Layer 2 forwarding engine. For DFC-equipped modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Trap feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	DFC-equipped modules.

PFC Layer 3 Tests

TestAclDeny

This test verifies that the ACL deny feature of the Layer 2 and Layer 3 forwarding engine is working properly. The test uses different ACL deny scenarios such as input, output, Layer 2 redirect, Layer 3 redirect, and Layer 3 bridges to determine whether or not the ACL deny feature is working properly.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	Run this test on-demand.
Default:	On.
Intitial Release:	12.2(14)SX.
Corrective action:	Automatic ASIC reset for recovery.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestAclPermit

This test verifies that the ACL permit functionality is working properly. An ACL entry permitting a specific diagnostics packet is installed in the ACL TCAM. The corresponding diagnostic packet is sent from the supervisor engine and looked up by the Layer 3 forwarding engine to make sure that it hits the ACL TCAM entry and gets permitted and forwarded appropriately.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	Run this test on-demand.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestFibDevices

This test verifies whether the FIB TCAM and adjacency devices are functional. One FIB entry is installed on each FIB TCAM device. A diagnostic packet is sent to make sure that the diagnostic packet is switched by the FIB TCAM entry installed on the TCAM device. This is not an exhaustive TCAM device test; only one entry is installed on each TCAM device.

Note Compared to the IPv4FibShortcut and IPv6FibShortcut tests, this test tests all FIB and adjacency devices using IPv4 or IPv6 packets, depending on your configuration.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Run this test on-demand to verify the Layer 3 forwarding functionality if you experience problems with the routing capability. This test can also be used as a health-monitoring test.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestIPv4FibShortcut

This test does the following:

- Verifies whether the IPv4 FIB forwarding of the Layer 3 forwarding engine is working properly. One diagnostic IPv4 FIB and an adjacency entry are installed, and a diagnostic packet is sent to make sure that the diagnostic packet is forwarded according to rewritten MAC and VLAN information.
- Verifies whether the FIB TCAM and adjacency devices are functional. One FIB entry is installed on each FIB TCAM device. A diagnostic packet is sent to make sure that the diagnostic packet is switched by the FIB TCAM entry installed on the TCAM device. This is not an exhaustive TCAM device test; only one entry is installed on each TCAM device.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Run this test on-demand to verify the Layer 3 forwarding functionality if you experience problems with the routing capability. This test can also be used as a health-monitoring test.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestIPv6FibShortcut

This test verifies that the IPV6 FIB forwarding of the Layer 3 forwarding engine is working properly. One diagnostic IPV6 FIB and an adjacency entry is installed, and a diagnostic IPv6 packet is sent to make sure the diagnostic packet is forwarded according to rewritten MAC and VLAN information.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Run this test on-demand to verify the Layer 3 forwarding functionality if you experience problems with the routing capability. This test can also be used as a health-monitoring test.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestL3Capture2

This test verifies that the Layer 3 capture (capture 2) feature of the Layer 3 forwarding engine is working properly. This capture feature is used for ACL logging and VACL logging. One diagnostic FIB and an adjacency entry with a capture 2 bit set is installed, and a diagnostic packet is sent to make sure that the diagnostic packet is forwarded according to the capture bit information.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	This test can also be used as a health-monitoring test. Use as a health-monitoring test if you are using ACL or VACL logging.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestMPLSFibShortcut

This test does the following:

- Verifies that the MPLS forwarding of the Layer 3 forwarding engine is working properly. One diagnostic MPLS FIB and an adjacency entry is installed, and a diagnostic MPLS packet is sent to make sure that the diagnostic packet is forwarded according to the MPLS label from the adjacency entry.
- Verifies the EoMPLS forwarding of the Layer 3 forwarding engine. One diagnostic EoMPLS Layer 2 FIB and an adjacency entry are installed and a diagnostic Layer 2 packet is sent to the forwarding engine to make sure it is forwarded accordingly with the MPLS labels and the encapsulated Layer 2 packet.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	This test can also be used as a health-monitoring test. Use as a health-monitoring test if you are routing MPLS traffic.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestNATFibShortcut

This test verifies the ability to rewrite a packet based on the NAT adjacency information (rewrite destination IP address). One diagnostic NAT FIB and an adjacency entry is installed, and the diagnostic packet is sent to make sure that the diagnostic packet is forwarded according to the rewritten IP address.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	This test can also be used as a health-monitoring test. Use as a health-monitoring test if the destination IP address is being rewritten (for example, if you are using NAT).
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestNetflowShortcut

This test verifies that the NetFlow forwarding functionality of the Layer 3 forwarding engine is working properly. One diagnostic NetFlow entry and an adjacency entry is installed, and a diagnostic packet is sent to make sure it is forwarded according to the rewritten MAC and VLAN information.

Attributes	Description
Disruptive or Nondisruptive:	Disruptive for looped back ports. The disruption is 500 ms.
Recommendation:	Run this test on-demand if you suspect that NetFlow is not working properly.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestQoSTcam

This test performs exhaustive memory tests for QoS TCAM devices.

Attributes	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is several minutes and can vary depending on the version of the PFC.
Recommendation:	Use this test only if you suspect a problem with the hardware or before putting the hardware into a live network. Do not run any traffic in the background on the module that you are testing. The supervisor engine must be rebooted after running this test.
Default:	Off.
Intitial Release:	12.2(18)SXD.
Corrective action:	Not applicable.
Hardware support:	All modules including supervisor engines.

DFC Layer 3 Tests

- TestAclDeny, page B-27
- TestAclFpgaMonitor, page B-27
- TestAclPermit, page B-28
- TestFibDevices, page B-28
- TestIPv4FibShortcut, page B-29
- TestIPv6FibShortcut, page B-29
- TestL3Capture2, page B-30

TestAclDeny

This test verifies that the ACL deny feature of the Layer 2 and Layer 3 forwarding engine is working properly. The test uses different ACL deny scenarios such as input and output Layer 2 redirect, Layer 3 redirect, and Layer 3 bridges.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	Schedule during downtime if you are using ACLs.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestAclFpgaMonitor

This test monitors the ACL FPGA for an invalid ACL TCAM reply and takes recovery action if an invalid reply is detected.

Attribute	Description
Disruptive or Nondisruptive:	NonDisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(18)SXF17a.
Corrective action:	Reset the module and optionally admin-down all the ports on the module.
Hardware support:	WS-X6748-GE-TX, WS-X6704-10GE, WS-X6724-SFP, WS-X6748-SFP modules with a DFC3B or DFC3BXL.

TestAclPermit

This test verifies that the ACL permit functionality is working properly. An ACL entry permitting a specific diagnostics packet is installed in the ACL TCAM. The corresponding diagnostic packet is sent from the supervisor engine and is looked up by the Layer 3 forwarding engine to make sure it hits the ACL TCAM entry and gets permitted and forwarded correctly.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestFibDevices

This test verifies whether the FIB TCAM and adjacency devices are functional. One FIB entry is installed on each FIB TCAM device. A diagnostic packet is sent to make sure that the diagnostic packet is switched by the FIB TCAM entry installed on the TCAM device. This is not an exhaustive TCAM device test; only one entry is installed on each TCAM device.

Note Compared to the IPv4FibShortcut and IPv6FibShortcut tests, this test tests all FIB and adjacency devices using IPv4 or IPv6 packets, depending on your configuration.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Run this test on-demand to verify the Layer 3 forwarding functionality if you experience problems with the routing capability. This test can also be used as a health-monitoring test.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestIPv4FibShortcut

These tests do the following:

- Verifies whether the IPv4 FIB forwarding of the Layer 3 forwarding engine is working properly. One diagnostic IPv4 FIB and an adjacency entry is installed, and a diagnostic packet is sent to make sure that the diagnostic packet is forwarded according to rewritten MAC and VLAN information.
- Verifies whether the FIB TCAM and adjacency devices are functional. One FIB entry is installed on each FIB TCAM device. A diagnostic packet is sent to make sure that the diagnostic packet is switched by the FIB TCAM entry installed on the TCAM device. This is not an exhaustive TCAM device test; only one entry is installed on each TCAM device.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestIPv6FibShortcut

This test verifies that the IPv6 FIB forwarding functionality of the Layer 3 forwarding engine is working properly. One diagnostic IPv6 FIB and an adjacency entry is installed, and a diagnostic IPv6 packet is sent to make sure that the diagnostic packet is forwarded according to rewritten MAC and VLAN information.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestL3Capture2

This test verifies that the Layer 3 capture (capture 2) feature of the Layer 3 forwarding engine is working properly. This capture feature is used for ACL logging and VACL logging. One diagnostic FIB and an adjacency entry with a capture 2-bit set is installed, and a diagnostic packet is sent to make sure that the diagnostic packet is forwarded according to capture bit information.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestMPLSFibShortcut

This test does the following:

- Verifies that the MPLS forwarding of the Layer 3 forwarding engine is working properly. One diagnostic MPLS FIB and an adjacency entry is installed, and a diagnostic MPLS packet is sent to make sure that the diagnostic packet is forwarded according to the MPLS label from the adjacency entry.
- Verifies the EoMPLS forwarding of the Layer 3 forwarding engine. One diagnostic EoMPLS Layer 2 FIB and an adjacency entry are installed and a diagnostic Layer 2 packet is sent to the forwarding engine to make sure it is forwarded accordingly with the MPLS labels and the encapsulated Layer 2 packet.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestNATFibShortcut

This test verifies the ability to rewrite a packet based on NAT adjacency information, such as the rewrite destination IP address. One diagnostic NAT FIB and an adjacency entry is installed, and a diagnostic packet is sent to the forwarding engine to make sure the diagnostic packet is forwarded according to the rewritten IP address.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of the looped-back port (for example, Spanning Tree Protocol).
Recommendation:	This test runs by default during bootup or after a reset or OIR.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestNetflowShortcut

This test verifies that the NetFlow forwarding functionality of the Layer 3 forwarding engine is working properly. One diagnostic NetFlow entry and an adjacency entry is installed, and a diagnostic packet is sent to make sure it is forwarded according to the rewritten MAC and VLAN information.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for looped-back ports. Disruption is typically less than one second.
Recommendation:	Run this test on-demand if you suspect that NetFlow is not working properly.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestQoSTcam

This test performs exhaustive memory tests for QoS TCAM devices.

Attributes	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is several minutes and can vary depending on the version of the PFC.
Recommendation:	Use this test only if you suspect a problem with the hardware or before putting the hardware into a live network. Do not run any traffic in the background on the module that you are testing. The supervisor engine must be rebooted after running this test.
Default:	Off.
Intitial Release:	12.2(18)SXD.
Corrective action:	Not applicable.
Hardware support:	All modules including supervisor engines.

Replication Engine Tests

TestEgressSpan

This test verifies that the egress SPAN replication functionality of the rewrite engine for both SPAN queues is working properly.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for both SPAN sessions. Disruption is typically less than one second.
Recommendation:	Run this test on-demand.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines, DFC-equipped modules, WS-65xx modules.

TestIngressSpan

This test ensures that the port ASIC is able to tag packets for ingress SPAN. This test also verifies that the ingress SPAN operation of the rewrite engine for both SPAN queues is working properly.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive for both SPAN sessions. Also disruptive for the loopback port on modules. Duration of the disruption depends on the configuration of the loopback port (for example, Spanning Tree Protocol).
Recommendation:	Run this test on-demand.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

TestL3VlanMet

This test verifies that the multicast functionality of the replication engine is working properly. The replication engine is configured to perform multicast replication of a diagnostic packet onto two different VLANs. After the diagnostic packet is sent out from the supervisor engine's inband port, the test verifies that two packets are received back in the inband port on the two VLANs configured in the replication engine.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive for supervisor engines. Disruptive for DFC-equipped modules. Disruption is typically less than one second on looped-back ports.
Recommendation:	Run this test on-demand to test the multicast replication abilities of the replication engine.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	None. See the system message guide for more information.
Hardware support:	Supervisor engines and DFC-equipped modules.

Fabric Tests

TestFabricCh0Health

This test constantly monitors the health of the ingress and egress data paths for fabric channel 0 on 10-gigabit modules. The test runs every five seconds. Ten consecutive failures are treated as fatal and the module resets; three consecutive reset cycles may result in a fabric switchover.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not turn this test off. Use as a health-monitoring test.
Default:	On.
Intitial Release:	12.2(14)SX.
Corrective action:	The module resets after 10 consecutive failures. Three consecutive resets powers down the module.
Hardware support:	WS-X6704-10GE.

TestFabricCh1Health

This test constantly monitors the health of the ingress and egress data paths for fabric channel 1 on 10-gigabit modules. The test runs every five seconds. Ten consecutive failures are treated as fatal and the module resets; three consecutive reset cycles might result in a fabric switchover.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not turn this test off. Use as a health-monitoring test.
Default:	On.
Intitial Release:	12.2(14)SX.
Corrective action:	The module resets after 10 consecutive failures. Three consecutive failures resets powers down the module.
Hardware support:	WS-X6704-10GE module.

TestFabricFlowControlStatus

This test reads the switch fabric ASIC registers to detect flow-control status for each fabric channel. Flow-control events are logged into the diagnostic events queue. By default, this test is disabled as a health-monitoring test, and when enabled, this test runs every 15 seconds. This test reports per-slot or per-channel rate reduction, current fabric channel utilization, peak fabric-channel utilization, and SP CPU utilization in both ingress and egress directions.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Use as a health-monitoring test. Use this test when you suspect a problem with the fabric channel.
Default:	Off.
Intitial Release:	12.2(33)SXH.
Corrective action:	Flow control events are logged into the diagnostic event log.
Hardware support:	Supervisor engines.

TestFabricSnakeBackward

This test consists of two test cases: the internal snake test and the external snake test. The internal snake test generates the test packets inside the fabric ASIC, and the test data path is limited so that it stays inside the fabric ASIC. The external snake test generates the test packet using the supervisor engine inband port and the test data path involves the port ASIC, the rewrite engine ASIC inside the supervisor engine, and the fabric ASIC. Whether or not the supervisor engine local channel is synchronized to the fabric ASIC determines which test is used. If it is synchronized, the external snake test is used; if it is not, internal snake test is used. For both tests, only the channels that are not synchronized to any modules are involved in the test. The backward direction indicates that the snaking direction is from the high-numbered channel to the low-numbered channel.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Run on-demand. This test can result in high CPU utilization.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	Supervisor engines crash to ROMMON; SFMs reset.
Hardware support:	Supervisor Engines.

TestFabricSnakeForward

This test consists of two test cases: the internal snake test and the external snake test. The internal snake test generates the test packets inside the fabric ASIC and the test data path is limited so that it stays inside the fabric ASIC. The external snake test generates the test packet using the supervisor engine inband port; the test data path involves the port ASIC, the rewrite engine ASIC inside the supervisor engine, and the fabric ASIC. Whether or not the supervisor engine local channel is synchronized to the fabric ASIC determines which test is used. If it is synchronized, the external snake test is used; if it is not, the internal snake test is used. For both tests, only the channels that are not synchronized to any modules are involved in the test. The Forward direction indicates that the snaking direction is from the low-numbered channel to the high-numbered channel.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Run on-demand. This test can result in high CPU utilization.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(14)SX.
Corrective action:	Supervisor engines crash to ROMMON; SFMs reset.
Hardware support:	Supervisor Engines.

TestSynchedFabChannel

This test periodically checks the fabric synchronization status for both the module and the fabric. This test is available only for fabric-enabled modules. This test is not a packet-switching test so it does not involve the data path. This test sends an SCP control message to the module and fabric to query the synchronization status.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not turn this test off. Use as a health-monitoring test.
Default:	On.
Intitial Release:	12.2(14)SX.
Corrective action:	The module resets after five consecutive failures. Three consecutive reset cycles results in the module powering down. A fabric switchover may be triggered, depending on the type of failure.
Hardware support:	All fabric-enabled modules.

Exhaustive Memory Tests

- TestAsicMemory, page B-37
- TestFibTcamSSRAM, page B-37

Note Because the supervisor engine must be rebooted after running memory tests, run memory tests on the other modules before running them on the supervisor engine. For more information about running on-demand online diagnostic tests see the "Configuring On-Demand Online Diagnostics" section on page 12-3.

TestAsicMemory

This test uses an algorithm to test the memory on a module.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is approximately one hour.
Recommendation:	Use this test only if you suspect a problem with the hardware or before putting the hardware into a live network. Do not run any traffic in the background on the module that you are testing. The supervisor engine must be rebooted after running this test.
Default:	Off.
Intitial Release:	12.2(17a)SX.
Corrective action:	Not applicable.
Hardware support:	All modules including supervisor engines.

TestFibTcamSSRAM

This test verifies the FIB TCAM and Layer 3 Adjacency SSRAM memory.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is several hours.
Recommendation:	Use this test only if you suspect a problem with the hardware or before putting the hardware into a live network. Do not run any traffic in the background on the module that you are testing. The supervisor engine must be rebooted after running this test.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	Not applicable.
Hardware support:	All modules including supervisor engines.

TestNetflowTcam

This test tests all the bits and checks the location of the Netflow TCAM.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is several minutes and can vary depending on the version of the PFC.
Recommendation:	Use this test only if you suspect a problem with the hardware or before putting the hardware into a live network. Do not run any traffic in the background on the module that you are testing. The supervisor engine must be rebooted after running this test.
Default:	Off.
Intitial Release:	12.2(18)SXD.
Corrective action:	Not applicable.
Hardware support:	All modules including supervisor engines.

TestQoSTcam

This test performs exhaustive memory tests for QoS TCAM devices.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is several minutes and can vary depending on the version of the PFC.
Recommendation:	Use this test only if you suspect a problem with the hardware or before putting the hardware into a live network. Do not run any traffic in the background on the module that you are testing. The supervisor engine must be rebooted after running this test.
Default:	Off.
Intitial Release:	12.2(18)SXD.
Corrective action:	Not applicable.
Hardware support:	All modules including supervisor engines.

Service Module Tests

- TestHapiEchoPkt, page B-39
- TestIPSecBaseComponents, page B-39
- TestIPSecClearPkt, page B-40
- TestIPSecEncryptDecryptPkt, page B-40
- TestIPSecSPAComponents, page B-41
- TestPcLoopback, page B-41
- TestPortASICLoopback, page B-41

TestHapiEchoPkt

This test sends a Hapi Echo packet to the crypto engine using the control path. After the Hapi Echo packet is sent to the crypto engine, it is echoed back from the crypto engine. The packet is sent from the supervisor engine inband port to the crypto engine using index-direct and is sent back using broadcast to a diagnostic VLAN.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	Run this test on-demand. This test cannot be run from on-demand CLI.
Default:	On.
Intitial Release:	12.2(18)SXE2.
Corrective action:	None. See the system message guide for more information.
Hardware support:	VPN service module.

TestIPSecBaseComponents

This test verifies components in 7600-SSC-400 modules in the run-time environment for hardware functionality and integrity.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	This test runs automatically during bootup.
Default:	On.
Intitial Release:	12.2(18)SXE2.
Corrective action:	None. See the system message guide for more information.
Hardware support:	7600-SSC-400.

TestIPSecClearPkt

This test sends a packet through the switch fabric or bus from the supervisor engine inband port through to the crypto engine. The packet is sent back without encryption from the crypto engine to the supervisor engine in-band port. The packet is checked to verify that the encryption is not done and that the packet data fields are reserved. The Layer 2 lookup drives the packet between the supervisor in-band port and the crypto engine.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Run this test on-demand.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(18)SXE2.
Corrective action:	None. See the system message guide for more information.
Hardware support:	VPN service module.

TestIPSecEncryptDecryptPkt

This test checks the encryption functionality by exchanging a packet between the supervisor engine in-band port and the crypto engine of the IPsec services modules (WS-SVC-IPSEC, SPA-IPSEC) using the switch fabric or bus (whichever is applicable). After several exchanges, the packet is checked to verify that the original data is preserved after the encryption and decryption process performed by the crypto engine. The Layer 2 lookup drives the packet between the supervisor in-band port and the crypto engine.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive. Test runs every minute by default.
Recommendation:	This test can only be run at bootup.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(18)SXE2.
Corrective action:	None. See the system message guide for more information.
Hardware support:	VPN services module.

TestIPSecSPAComponents

This test verifies components in SPA-IPSEC-2G modules in the run-time environment for hardware functionality and integrity.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	This test runs automatically during bootup.
Default:	On.
Intitial Release:	12.2(18)SXE2.
Corrective action:	None. See the system message guide for more information.
Hardware support:	SPA-IPSEC-2G.

TestPcLoopback

This test verifies the longest datapath between the supervisor and the NAM service module. A packet is sent from the supervisor to the module and is looped back by the PC to the supervisor engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	This test runs automatically during bootup.
Default:	On.
Intitial Release:	12.2(14)SX1.
Corrective action:	None. See the system message guide for more information.
Hardware support:	WS-SVC-NAM-1, WS-SVC-NAM-2.

TestPortASICLoopback

This test verifies the health of the ASIC ports on the NAM service module. A packet is sent from the supervisor engine and looped back at the ASIC.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	This test runs automatically during bootup.
Default:	On.
Intitial Release:	12.2(14)SX1.
Corrective action:	None. See the system message guide for more information.
Hardware support:	WS-SVC-NAM-1, WS-SVC-NAM-2.

Stress Tests

TestEobcStressPing

This test stresses a module's EOBC link with the supervisor engine. The test is started when the supervisor engine initiates a number of sweep-ping processes (the default is one). The sweep-ping process pings the module with 20,000 SCP-ping packets. The test passes if all 20,000 packets respond before each packet-ping timeout, which is two seconds. If unsuccessful, the test allows five retries to account for traffic bursts on the EOBC bus during the test.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is several minutes.
Recommendation:	Use this test to qualify hardware before installing it in your network.
Default:	Off.
Intitial Release:	12.2(18)SXD.
Corrective action:	Not applicable.
Hardware support:	Supervisor engines.

TestTrafficStress

This test stress tests the switch and the installed modules by configuring all of the ports on the modules into pairs, which then pass packets between each other. After allowing the packets to pass through the switch for a predetermined period, the test verifies that the packets are not dropped.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is several minutes.
Recommendation:	Use this test to qualify hardware before installing it in your network.
Default:	Off.
Intitial Release:	12.2(18)SXF.
Corrective action:	Not applicable.
Hardware support:	Supervisor engines.

General Tests

BusConnectivityTest

This test verifies the bus connectivity on WAN modules by sending packets from the supervisor engine to the module, where they are looped back by the bus ASIC and returned to the supervisor engine.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	This test runs automatically during bootup.
Default:	On.
Intitial Release:	12.2(18)SXF5.
Corrective action:	None. See the system message guide for more information.
Hardware support:	7600-SIP-400, 7600-SIP-200, WS-X6582-2PA.

ScheduleSwitchover

This test allows you to trigger a switchover at any time using the online diagnostics scheduling capability.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	Schedule this test during downtime to test the ability of the standby supervisor engine to take over after a switchover.
Default:	Off.
Intitial Release:	12.2(17b)SXA.
Corrective action:	None
Hardware support:	Supervisor engines.

TestCFRW

This test verifies the CompactFlash disk or disks on the supervisor engine. This test is performed during system boot-up or whenever a disk is inserted. A 128-byte temporary file is written to each disk present in the slot and read back. The content read back is checked and the temporary file is deleted. You can also execute this test from the CLI.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not disable. No traffic is affected.
Default:	On.
Intitial Release:	12.2(33)SXH.
Corrective action:	Format or replace the failed CompactFlash.
Hardware support:	Removable CompactFlash devices.

TestFirmwareDiagStatus

This test displays the results of the power-on diagnostic tests run by the firmware during the module bootup.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	This test can only be run at bootup.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(18)SXD.
Corrective action:	None. See the system message guide.
Hardware support:	All modules.

TestOBFL

This test verifies the on-board failure logging capabilities. During this test a diagnostic message is logged on the module.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	This test is run automatically during bootup and cannot be run on-demand.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(33)SXH.
Corrective action:	Not applicable.
Hardware support:	Supervisor engines, DFC-equipped switching modules, WS-SVC-WISM2.

TestRwEngineOverSubscription

This is a health-monitoring test that is not enabled by default. This test runs on the module every one second and checks if the rewrite engine gets oversubscribed by retrieving drop counters and generates a syslog message if the drops exceed the set threshold.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	This test is run only as a health-monitoring test.
Default:	Off.
Intitial Release:	12.2(33)SXI.
Corrective action:	Not applicable.
Hardware support:	Supervisor engines, DFC-equipped modules.

TestSpuriousIsrDetection

This test is run when an interrupt is detected on a fabric ASIC.This test is not a bootup test and cannot be run on demand. Failure of this test is treated as fatal, leading to supervisor engine crash.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	This test runs only when there is a interrupt detected.
Default:	Off.
Intitial Release:	12.2(33)SXH.
Corrective action:	Not applicable.
Hardware support:	Supervisor engines.

TestVDB

This test is available on PoE-equipped modules. This test queries the result of diagnostic tests that run on the PoE daughter card.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	This test is run automatically during bootup.
Default:	Off.
Intitial Release:	12.2(17)SXA.
Corrective action:	Not applicable.
Hardware support:	Modules with a PoE daughter card.

Critical Recovery Tests

- TestAclFpgaMonitor, page B-46
- TestL3HealthMonitoring, page B-47
- TestTxPathMonitoring, page B-47

Note These tests are also considered critical recovery tests:

- TestFabricCh0Health, page B-34
- TestFabricCh1Health, page B-34
- TestSynchedFabChannel, page B-36

TestAclFpgaMonitor

This test monitors the ACL FPGA for an invalid ACL TCAM reply status and takes recovery action if an invalid reply is detected.

Attribute	Description
Disruptive or Nondisruptive:	NonDisruptive.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(18)SXF17a.
Corrective action:	Reset the module and optionally admin-down all the ports on the module.
Hardware support:	DFC-equipped WS-X6748-GE-TX, WS-X6704-10GE, WS-X6724-SFP, WS-X6748-SFP modules with WS-F6700-DFC3B or WS-F6700-DFC3BXL DFC.

TestL3HealthMonitoring

This test triggers a set of diagnostic tests involving IPv4 and IPv6 packet switching on a DFC whenever the system tries to self-recover from a detected hardware fault. The tests shut down the front panel port (usually port 1) for testing purposes. If the diagnostic tests are not passing, it is an indication that the hardware fault cannot be fixed and a self-recovery sequence will be applied again.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive. Disruption is typically less than one second. Duration of the disruption depends on the configuration of looped-back port (for example, Spanning Tree Protocol). Forwarding and port functions are disrupted during the test.
Recommendation:	Do not disable.
Default:	Off.
Intitial Release:	12.2(14)SX.
Corrective action:	Not applicable.
Hardware support:	DFC-equipped modules.

TestTxPathMonitoring

This test sends index-directed packets periodically to each port on the supervisor engine and supported modules to verify ASIC synchronization and correct any related problems. The test runs every two seconds.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive.
Recommendation:	Do not change the default settings.
Default:	On.
Intitial Release:	12.2(14)SX.
Corrective action:	Not applicable (self-recovering).
Hardware support:	Supervisor engines and DFC-equipped modules.

ViSN Tests

TestRslHm

This test monitors the data and control links between the remote switch and core switches. A diagnostic packet is sent from the supervisor engine inband port on the remote switch to the supervisor engine inband port on the core switch and is pinged back along the reverse data path. This tests each RSL link between the remote switch and both active and standby core switches.

Attribute	Description
Disruptive or Nondisruptive:	Nondisruptive health monitoring test.
Recommendation:	Do not disable.
Default:	On.
Intitial Release:	12.2(33)SXH1.
Corrective action:	None. See the system message guide for more information.
Hardware support:	VSL-capable modules.

TestVSActiveToStandbyLoopback

This test is the only GOLD test that tests the full data path across the virtual switch links. This test selects an uplink port in the standby virtual switch supervisor engine as the loopback point and sends the VLAN flood packet from the active virtual switch supervisor engine inband port to the system. Due to the configuration of the FPOE and LTL VLAN flood region for all VSL modules and VSL interfaces in the active and standby virtual switch, the packet goes across VSL and arrives at the uplink port of the standby virtual switch supervisor engines, and loopbacks from there. The packet comes back to the inband port of the active supervisor engine due to the preconfiguration of FPOE and LTL in the standby and active virtual switches. In case of a test failure, the error check is executed for SP CPU, fabric flow control, and other errors in both active and standby virtual swtiches.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	Disable all health monitoring tests before executing this test. This test is run only for on-demand diagnostic testing.
Default:	Off.
Intitial Release:	12.2(33)SXH1.
Corrective action:	Not applicable.
Hardware support:	VSL-capable modules.

TestVslBridgeLink

This test provides diagnostic coverage for VSL-capable modules and the supervisor engine during module bootup. The data path of this test picks only one port corresponding to the local and remote bridge inband port as the loopback points. A diagnostic packet is sent from the inband port of the supervisor engine to the loopback points on the VSL module, and the packet traverses the bridge link between two fabric data path complexes to verify the hardware bridge link functionality.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	This test is run automatically during bootup.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(33)SXH1.
Corrective action:	Not applicable.
Hardware support:	VSL-capable modules.

TestVslLocalLoopback

This test verifies the hardware functionality of each port on the VSL module before the VSL link interface is up. The data path of this test is constrained with the VSL module. A diagnostic packet is sent from the local inband port of the VSL module to each port to run a loopback test. This test is run only during module bootup.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	This test is run automatically during bootup and cannot be run on-demand.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(33)SXH1.
Corrective action:	Not applicable.
Hardware support:	VSL-capable modules.

TestVslStatus

This test reports the status change detected by the VSLP protocol. When any link problem is detected by the VSLP protocol, the status of the link is changed and the result is updated accordingly. This test also triggers the loopback test to check the hardware status requested by the VSLP protocol.

Attribute	Description
Disruptive or Nondisruptive:	Disruptive.
Recommendation:	This test is effective once the VSL modules are online.
Default:	This test runs by default during bootup or after a reset or OIR.
Intitial Release:	12.2(33)SXH1.
Corrective action:	Not applicable.
Hardware support:	VSL-capable modules.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Acronyms

Table C-1 defines the acronyms used in this publication.

Table C-1 List of Acronyms

Acronym	Expansion
AAL	ATM adaptation layer
ACE	access control entry
ACL	access control list
AFI	authority and format identifier
Agport	aggregation port
ALPS	Airline Protocol Support
AMP	Active Monitor Present
APaRT	Automated Packet Recognition and Translation
ARP	Address Resolution Protocol
ATA	Analog Telephone Adaptor
ATM	Asynchronous Transfer Mode
AV	attribute value
BDD	binary decision diagrams
BECN	backward explicit congestion notification
BGP	Border Gateway Protocol
BPDU	bridge protocol data unit
BRF	bridge relay function
BSC	Bisync
BSTUN	Block Serial Tunnel
BUS	broadcast and unknown server
BVI	bridge-group virtual interface
CAM	content-addressable memory
CAR	committed access rate
CCA	circuit card assembly
CDP	Cisco Discovery Protocol

Table C-1 List of Acronyms (continued)

Acronym	Expansion
CEF	Cisco Express Forwarding
CHAP	Challenge Handshake Authentication Protocol
CIR	committed information rate
CIST	Common and internal spanning tree
CLI	command-line interface
CLNS	Connection-Less Network Service
CMNS	Connection-Mode Network Service
COPS	Common Open Policy Server
COPS-DS	Common Open Policy Server Differentiated Services
CoS	class of service
CPLD	Complex Programmable Logic Device
CRC	cyclic redundancy check
CRF	concentrator relay function
CST	Common Spanning Tree
CUDD	University of Colorado Decision Diagram
DCC	Data Country Code
dCEF	distributed Cisco Express Forwarding
DDR	dial-on-demand routing
DE	discard eligibility
DEC	Digital Equipment Corporation
DFC	Distributed Forwarding Card
DFI	Domain-Specific Part Format Identifier
DFP	Dynamic Feedback Protocol
DISL	Dynamic Inter-Switch Link
DLC	Data Link Control
DLSw	Data Link Switching
DMP	data movement processor
DNS	Domain Name System
DoD	Department of Defense
DOS	denial of service
dot1q	802.1Q
DRAM	dynamic RAM
DRiP	Dual Ring Protocol
DSAP	destination service access point
DSCP	differentiated services code point
DSPU	downstream SNA Physical Units

Table C-1 List of Acronyms (continued)

Acronym	Expansion
DTP	Dynamic Trunking Protocol
DTR	data terminal ready
DWRR	deficit weighted round robin
DXI	data exchange interface
EAP	Extensible Authentication Protocol
EARL	Enhanced Address Recognition Logic
EEPROM	electrically erasable programmable read-only memory
EHSA	enhanced high system availability
EIA	Electronic Industries Association
ELAN	Emulated Local Area Network
EOBC	Ethernet out-of-band channel
EOF	end of file
ESI	end-system identifier
FAT	File Allocation Table
FECN	forward explicit congestion notification
FM	feature manager
FRU	field replaceable unit
fsck	file system consistency check
FSM	feasible successor metrics
GARP	General Attribute Registration Protocol
GMRP	GARP Multicast Registration Protocol
GVRP	GARP VLAN Registration Protocol
HSRP	Hot Standby Routing Protocol
ICC	Inter-card Communication
ICD	International Code Designator
ICMP	Internet Control Message Protocol
IDB	interface descriptor block
IDP	initial domain part or Internet Datagram Protocol
IDSM	Intrusion Detection System Module
IFS	IOS File System
IGMP	Internet Group Management Protocol
IGRP	Interior Gateway Routing Protocol
ILMI	Integrated Local Management Interface
IP	Internet Protocol
IPC	interprocessor communication
IPX	Internetwork Packet Exchange

Table C-1 List of Acronyms (continued)

Acronym	Expansion
IS-IS	Intermediate System-to-Intermediate System Intradomain Routing Protocol
ISL	Inter-Switch Link
ISO	International Organization of Standardization
ISR	Integrated SONET router
IST	Internal spanning tree
LAN	local area network
LANE	LAN Emulation
LAPB	Link Access Procedure, Balanced
LCP	Link Control Protocol
LDA	Local Director Acceleration
LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LEM	link error monitor
LER	link error rate
LES	LAN Emulation Server
LLC	Logical Link Control
LTL	Local Target Logic
MAC	Media Access Control
MD5	Message Digest 5
MDS	multicast distributed switching
MDT	multicast distribution tree
MEC	multichassis EtherChannel
MFD	multicast fast drop
MIB	Management Information Base
MII	media-independent interface
MLD	Multicast Listener Discovery
MLS	Multilayer Switching
MLSE	maintenance loop signaling entity
MLSE	maintenance loops signaling entity
MOP	Maintenance Operation Protocol
MOTD	message-of-the-day
MRM	multicast routing monitor
MSDP	Multicast Source Discovery Protocol
MSFC	Multilayer Switch Feature Card
MSFC	Multilayer Switching Feature Card

Table C-1 List of Acronyms (continued)

Acronym	Expansion
MSM	Multilayer Switch Module
MST	multiple spanning tree
MTI	multicast tunnel interface
MTU	maximum transmission unit
MVAP	multiple VLAN access port
MVPN	multicast virtual private network
MVR	mulitcast VLAN registration
MVRF	multicast VRF
NAC	Network Admission Control
NAM	Network Analysis Module
NBAR	Network-Based Application Recognition
NBP	Name Binding Protocol
NCIA	Native Client Interface Architecture
NDE	NetFlow Data Export
NET	network entity title
NetBIOS	Network Basic Input/Output System
NFFC	NetFlow Feature Card
NMP	Network Management Processor
NSAP	network service access point
NSF	Nonstop Forwarding
NTP	Network Time Protocol
NVRAM	nonvolatile RAM
OAM	Operation, Administration, and Maintenance
ODM	order dependent merge
OSI	Open System Interconnection
OSM	Optical Services Module
OSPF	open shortest path first
PACL	port access control list
PAE	port access entity
PAgP	Port Aggregation Protocol
PBD	packet buffer daughterboard
PC	Personal Computer (formerly PCMCIA)
PCM	pulse code modulation
PCR	peak cell rate
PDP	policy decision point
PDU	protocol data unit

Table C-1 List of Acronyms (continued)

Acronym	Expansion
PEP	policy enforcement point
PFC	Policy Feature Card
PGM	Pragmatic General Multicast
PHY	physical sublayer
PIB	policy information base
PIM	protocol independent multicast
PPP	Point-to-Point Protocol
PRID	Policy Rule Identifiers
PVID	Port VLAN identifier
PVST+	Per VLAN Spanning Tree+
QDM	QoS device manager
QM	QoS manager
QoS	quality of service
RACL	router interface access control list
RADIUS	Remote Access Dial-In User Service
RAM	random-access memory
RCP	Remote Copy Protocol
RGMP	Router-Ports Group Management Protocol
RIB	routing information base
RIF	Routing Information Field
RMON	remote network monitor
ROM	read-only memory
ROMMON	ROM monitor
RP	route processor or rendezvous point
RPC	remote procedure call
RPF	reverse path forwarding
RPR	route processor redundancy
RPR+	route processor redundancy plus
RSPAN	remote SPAN
RST	reset
RSVP	ReSerVation Protocol
SAID	Security Association Identifier
SAP	service access point
SCM	service connection manager
SCP	Switch-Module Configuration Protocol
SDLC	Synchronous Data Link Control

Table C-1 List of Acronyms (continued)

Acronym	Expansion
SDM	Secure Device Manager
SEA	System Event Archive
SGBP	Stack Group Bidding Protocol
SIMM	single in-line memory module
SLB	server load balancing
SLCP	Supervisor Line-Card Processor
SLIP	Serial Line Internet Protocol
SMDS	Software Management and Delivery Systems
SMF	software MAC filter
SMP	Standby Monitor Present
SMRP	Simple Multicast Routing Protocol
SMT	Station Management
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
SPAN	Switched Port Analyzer
SREC	S-Record format, Motorola defined format for ROM contents
SRM	single router mode
SRR	shaped round robin
SSM	source-specific multicast
SSO	stateful switchover
SSTP	Cisco Shared Spanning Tree
STP	Spanning Tree Protocol
SVC	switched virtual circuit
SVI	switched virtual interface
TACACS+	Terminal Access Controller Access Control System Plus
TARP	Target Identifier Address Resolution Protocol
TCAM	Ternary Content Addressable Memory
TCL	table contention level
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
TIA	Telecommunications Industry Association
TLV	type-length-value
TopN	Utility that allows the user to analyze port traffic by reports
TOS	type of service
TTL	Time To Live
TVX	valid transmission

Table C-1 List of Acronyms (continued)

Acronym	Expansion
UDLD	UniDirectional Link Detection Protocol
UDP	User Datagram Protocol
UNI	User-Network Interface
URD	URL Rendezvous Directory
UTC	Coordinated Universal Time
UUFB	unknown unicast flood blocking
UUFRL	unknown unicast flood rate limiting
VACL	VLAN access control list
VCC	virtual channel circuit
VCI	virtual circuit identifier
VCR	Virtual Configuration Register
VINES	Virtual Network System
VLAN	virtual LAN
VMPS	VLAN Membership Policy Server
VPN	virtual private network
VRF	VPN routing and forwarding
VTP	VLAN Trunking Protocol
VVID	voice VLAN ID
WAN	wide area network
WCCP	Web Cache Communications Protocol
WFQ	weighted fair queueing
WoL	wake-on LAN
WRED	weighted random early detection
WRR	weighted round-robin
XNS	Xerox Network System

INDEX

Numerics

A

B

C

D

E

F

G

H

I

M

N

O

P

Q

R

S

T

U

V

W

X