Catalyst Supervisor Engine 32 PISA Cisco IOS Software Configuration Guide

Release 12.2(18)ZY 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-11439-03

Catalyst Supervisor Engine 32 PISA Cisco IOS Software Configuration Guide, Release 12.2ZY

CONTENTS

Preface

This preface describes who should read the *Catalyst Supervisor Engine 32 PISA Cisco IOS Software Configuration Guide*, Release 12.2ZY, and its document conventions.

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining Catalyst 6500 series switches.

Related Documentation

The following publications are available for the Catalyst 6500 series switches:

- *Catalyst 6500 Series Switch Installation Guide*
- *Catalyst 6500 Series Switch Module Installation Guide*
- *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY
- *Catalyst Supervisor Engine 32 PISA Cisco IOS System Message Guide*, Release 12.2ZY
- *Release Notes for Cisco IOS Release 12.2ZY on the Supervisor Engine 32 PISA*
- *Cisco IOS Configuration Guides and Command References*—Use these publications to help you configure Cisco IOS software features not described in the Catalyst 6500 series switch publications:
  - *Configuration Fundamentals Configuration Guide*
  - *Configuration Fundamentals Command Reference*
  - *Bridging and IBM Networking Configuration Guide*
  - *Bridging and IBM Networking Command Reference*
  - *Interface Configuration Guide*
  - *Interface Command Reference*
  - *Network Protocols Configuration Guide*, Part 1, 2, and 3
  - *Network Protocols Command Reference*, Part 1, 2, and 3
  - *Security Configuration Guide*
  - *Security Command Reference*
  - *Switching Services Configuration Guide*

- *Switching Services Command Reference*
- *Voice, Video, and Home Applications Configuration Guide*
- *Voice, Video, and Home Applications Command Reference*
- *Software Command Summary*
- *Software System Error Messages*
- *Debug Command Reference*
- *Internetwork Design Guide*
- *Internetwork Troubleshooting Guide*
- *Configuration Builder Getting Started Guide*

The Cisco IOS Configuration Guides and Command References are located at this URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_installation_and_configuration_guides_list.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*.
[ ]	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 and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

CHAPTER 1

Product Overview

This document provides configuration procedures for the Supervisor Engine 32 and Programmable Intelligent Services Accelerator (PISA). This chapter consists of these sections:

Supported Hardware and Software

For complete information about the chassis, modules, and software features supported by the Supervisor Engine 32 PISA, refer to the *Release Notes for Cisco IOS Release 12.2ZY on the Supervisor Engine 32 PISA*:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/release/notes/ol_13011.html

To configure Network-Based Application Recognition (NBAR) , see this publication:

http://www.cisco.com/en/US/docs/ios/12_4t/qos/configuration/guide/qsnbar1.html

To configure flexible packet matching (FPM), see these publications:

http://www.cisco.com/en/US/docs/ios/12_4t/12_4t4/ht_fpm.html

http://www.cisco.com/en/US/docs/ios/12_4t/12_4t11/ht_tcdf.html

User Interfaces

Release 12.2ZY supports configuration using the following interfaces:

- CLI—See Chapter 2, “Command-Line Interfaces.”
- SNMP—Refer to 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—Refer to “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

- Embedded CiscoView—See the "Configuring Embedded CiscoView Support" section on page 1-2.

Configuring Embedded CiscoView Support

These sections describe configuring Embedded CiscoView support:

Understanding Embedded CiscoView

The Embedded CiscoView network management system is a web-based interface that uses HTTP and SNMP to provide a graphical representation of the switch and to provide a GUI-based management and configuration interface.

Installing and Configuring Embedded CiscoView

To install and configure Embedded CiscoView, perform this task:

	Command	Purpose
Step 1	`Router#` **`dir`** *`device_name`*	Displays the contents of the device. If you are installing Embedded CiscoView for the first time, or if the CiscoView directory is empty, skip to Step 4.
Step 2	`Router#` **`delete`** *`device_name`***`:cv/*`**	Removes existing files from the CiscoView directory.
Step 3	`Router#` **`squeeze`** *`device_name`***`:`**	Recovers the space in the file system.
Step 4	`Router#` **`archive tar /xtract tftp://`** *`ip_address_of_tftp_server`***`/ciscoview.tar`** *`device_name`***`:cv`**	Extracts the CiscoView files from the tar file on the TFTP server to the CiscoView directory.
Step 5	`Router#` **`dir`** *`device_name`***`:`**	Displays the contents of the device. In a redundant configuration, repeat Step 1 through Step 5 for the file system on the redundant supervisor engine.
Step 6	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 7	`Router(config)#` **`ip http server`**	Enables the HTTP web server.
Step 8	`Router(config)#` **`snmp-server community`** *`string`* **`ro`**	Configures the SNMP password for read-only operation.
Step 9	`Router(config)#` **`snmp-server community`** *`string`* **`rw`**	Configures the SNMP password for read/write operation.

Note The default password for accessing the switch web page is the enable-level password of the switch.

For more information about web access to the switch, refer to "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

Displaying Embedded CiscoView Information

To display the Embedded CiscoView information, enter the following EXEC commands:

Command	Purpose
`Router# show ciscoview package`	Displays information about the Embedded CiscoView files.
`Router# show ciscoview version`	Displays the Embedded CiscoView version.

Software Features Supported in Hardware by the PFC3B

The PFC3B provides 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 PISA.

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

    Note Idle timeout is processed in software on the PISA.

  For more information about PFC3B support for ACLs, see Chapter 31, "Understanding Cisco IOS ACL Support."

  For complete information about configuring ACLs, refer to 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

- VLAN ACLs (VACLs)—To configure VACLs, see Chapter 32, "Configuring VLAN ACLs."

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

- Except on MPLS interfaces, TCP intercept—To configure TCP intercept, see the "Configuring TCP Intercept" section on page 30-2.
- Hardware-assisted NetFlow Aggregation—Refer to this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/configuration/guide/nde.html
- Bidirectional Protocol Independent Multicast (PIM) in hardware—See "Understanding How IPv4 Bidirectional PIM Works" section on page 25-6.
- Multiple-path Unicast Reverse Path Forwarding (RPF) Check—To configure Unicast RPF Check, see the "Configuring Unicast Reverse Path Forwarding Check" section on page 30-2.
- Except on MPLS interfaces, Network Address Translation (NAT) for IPv4 unicast and multicast traffic.

  Note the following information about hardware-assisted NAT:

  - The PFC3B does not support NAT of multicast traffic.
  - The PFC3B does not support NAT configured with a route-map that specifies length.
  - When you configure NAT and NDE on an interface, the PFC3B sends all traffic in fragmented packets to the PISA to be processed in software. (CSCdz51590)

  To configure NAT, refer to 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 PISA, 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/switches/lan/catalyst6500/ios/12.2ZY/command/reference/M1.html

  (CSCea23296)
- IPv4 Multicast over point-to-point generic route encapsulation (GRE) Tunnels—Refer to the publication at this URL:

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

Note The PFC3B does not provide hardware acceleration for tunnels configured with the **tunnel key** command.

- GRE Tunneling and IP in IP Tunneling—The PFC3B supports the following **tunnel** commands:
  - **tunnel destination**
  - **tunnel mode gre**
  - **tunnel mode ipip**
  - **tunnel source**
  - **tunnel ttl**
  - **tunnel tos**

  The PISA supports tunneling configured with any other tunnel commands.

  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, refer to these publications:

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

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/finter_r/irfshoip.htm

  To configure the **tunnel tos** and **tunnel ttl** commands, refer to this publication:

  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. (CSCdy72539)
  - Each tunnel interface uses one internal VLAN.
  - Each tunnel interface uses one additional router MAC address entry per router MAC address.
  - The PFC3B supports PFC QoS features on tunnel interfaces.
  - The PISA 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, CBAC, and encryption.

CHAPTER 2

Command-Line Interfaces

This chapter describes the command-line interfaces (CLIs) you use to configure the Catalyst 6500 series switches.

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

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

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

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 19-3).

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 "Securing the CLI" section on page 2-6.

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/12-2sx/sec-secure-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/sec_usr_ssh/configuration/12-2sx/sec-secure-copy.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 *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY.

CHAPTER 3

Configuring the Switch for the First Time

This chapter contains information about how to initially configure the Catalyst 6500 series switch, which supplements the administration information and procedures in these publications:

- *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2, at this URL:

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

- *Cisco IOS Configuration Fundamentals Configuration Command Reference*, Release 12.2, at this URL:

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

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

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

This chapter consists of these sections:

Default Configuration

Table 3-1 shows the default configuration.

Table 3-1 Default Configuration

Feature	Default Value
Administrative connection	Normal mode
Global information	No value for the following: • System name • System contact • Location
System clock	No value for system clock time
Passwords	No passwords configured for normal mode or enable mode (press the **Return** key)
Prompt	`Router>`

Configuring the Switch

These sections describe how to configure the switch:

Using the Setup Facility or the setup Command

These sections describe the setup facility and the **setup** command:

Setup Overview

At initial startup, the switch automatically defaults to the setup facility. (The **setup** command facility functions exactly the same as a completely unconfigured system functions when you first boot it up.) You can run the setup facility by entering the **setup** command at the enable prompt (#).

When you enter the **setup** command, current system configuration defaults are displayed in square brackets [ ] as you move through the **setup** command process and are queried by the system to make changes.

For example, you will see this display when you use the setup facility:

```
Configuring interface FastEtherent3/1:
```

```
  Is this interface in use?: yes
  Configure IP on this interface?: yes
```

When you use the **setup** command, you see this display:

```
Configuring interface FastEthernet4/1:
  Is this interface in use?[yes]: yes
  Configure IP on this interface?[yes]: yes
```

Configuring the Global Parameters

When you first start the setup facility or enter the **setup** command, you are queried by the system to configure the global parameters, which are used for controlling system-wide settings.

To boot the switch and enter the global parameters, follow these steps:

Step 1 Connect a console terminal to the console interface on the supervisor engine, and then boot the system to the user EXEC prompt (`Router>`).

The following display appears after you boot the Catalyst 6500 series switch (depending on your configuration, your display might not exactly match the example):

```
System Bootstrap, Version 6.1(2)
Copyright (c) 1994-2000 by cisco Systems, Inc.
c6k_sup2 processor with 131072 Kbytes of main memory

rommon 1 > boot disk0:c6sup22-jsv-mz.121-5c.EX.bin

Self decompressing the image : ##################################################
##############################################################################
##############################################################################
##############################################################################
##############################################################################
 [OK]

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco Internetwork Operating System Software
IOS (tm) c6sup2_sp Software (c6sup2_sp-SPV-M), Version 12.1(5c)EX, EARLY DEPLOYM
ENT RELEASE SOFTWARE (fc1)
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 18:36 by hqluong
Image text-base: 0x30020980, data-base: 0x306B8000

Start as Primary processor

00:00:05: %SYS-3-LOGGER_FLUSHING: System pausing to ensure console debugging out
```

```
put.

00:00:03: Currently running ROMMON from S (Gold) region
00:00:05: %OIR-6-CONSOLE: Changing console ownership to route processor



System Bootstrap, Version 12.1(3r)E2, RELEASE SOFTWARE (fc1)
Copyright (c) 2000 by cisco Systems, Inc.
Cat6k-MSFC2 platform with 131072 Kbytes of main memory

rommon 1 > boot

Self decompressing the image : ################################################
################################################################################
## [OK]

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco Internetwork Operating System Software
IOS (tm) MSFC2 Software (C6MSFC2-BOOT-M), Version 12.1(3a)E4, EARLY DEPLOYMENT R
ELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Sat 14-Oct-00 05:33 by eaarmas
Image text-base: 0x30008980, data-base: 0x303B6000

cisco Cat6k-MSFC2 (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
X.25 software, Version 3.0.0.
509K bytes of non-volatile configuration memory.

16384K bytes of Flash internal SIMM (Sector size 512K).


Press RETURN to get started!
```

Note The first two sections of the configuration script (the banner and the installed hardware) appear only at initial system startup. On subsequent uses of the **setup** command facility, the setup script begins with the following System Configuration Dialog.

```
--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: y

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
```

```
Default settings are in square brackets '[]'.


Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system
```

Note The examples in this section are intended as examples only. Your configuration might look differently depending on your system configuration.

Step 2 Enter **yes** or press **Return** when asked if you want to enter the configuration dialog and if you want to see the current interface summary. Press **Return** to accept the default (yes):

```
Would you like to enter the initial configuration dialog? [yes]:

First, would you like to see the current interface summary? [yes]:
```

This example of a **yes** response (displayed during the setup facility) shows a switch at first-time startup; that is, nothing has been configured:

```
Current interface summary

Interface                  IP-Address      OK? Method Status                Protocol
Vlan1                      unassigned      YES TFTP   administratively down down

GigabitEthernet1/1         unassigned      YES TFTP   administratively down down

GigabitEthernet1/2         unassigned      YES TFTP   administratively down down

GigabitEthernet3/1         unassigned      YES TFTP   administratively down down

GigabitEthernet3/2         unassigned      YES TFTP   administratively down down

GigabitEthernet3/3         unassigned      YES TFTP   administratively down down

GigabitEthernet3/4         unassigned      YES TFTP   administratively down down

GigabitEthernet3/5         unassigned      YES TFTP   administratively down down

GigabitEthernet3/6         unassigned      YES TFTP   administratively down down

GigabitEthernet3/7         unassigned      YES TFTP   administratively down down

GigabitEthernet3/8         unassigned      YES TFTP   administratively down down

(Additional displayed text omitted from this example.)
```

This example of a **yes** response (displayed during the setup command facility) shows a switch with some interfaces already configured:

```
Current interface summary

Interface                  IP-Address      OK? Method Status                Protocol
Vlan1                      unassigned      YES TFTP   administratively down down

GigabitEthernet1/1         172.20.52.34    YES NVRAM  up                    up

GigabitEthernet1/2         unassigned      YES TFTP   administratively down down

GigabitEthernet3/1         unassigned      YES TFTP   administratively down down
```

```
GigabitEthernet3/2         unassigned      YES TFTP   administratively down down

GigabitEthernet3/3         unassigned      YES TFTP   administratively down down

GigabitEthernet3/4         unassigned      YES TFTP   administratively down down

GigabitEthernet3/5         unassigned      YES TFTP   administratively down down

GigabitEthernet3/6         unassigned      YES TFTP   administratively down down

GigabitEthernet3/7         unassigned      YES TFTP   administratively down down

GigabitEthernet3/8         unassigned      YES TFTP   administratively down down

<...output truncated...>
```

Step 3 Choose which protocols to support on your interfaces. On IP installations only, you can accept the default values for most of the questions.

A typical minimal configuration using IP follows and continues through Step 8:

```
Configuring global parameters:

  Enter host name [Router]: Router
```

Step 4 Enter the enable secret password when the following is displayed (remember this password for future reference):

```
The enable secret is a password used to protect access to
privileged EXEC and configuration modes. This password, after
entered, becomes encrypted in the configuration.
Enter enable secret: barney
```

Step 5 Enter the enable password when the following is displayed (remember this password for future reference):

```
The enable password is used when you do not specify an
enable secret password, with some older software versions, and
some boot images.
Enter enable password: wilma
```

The commands available at the user EXEC level are a subset of those available at the privileged EXEC level. Because many privileged EXEC commands are used to set operating parameters, you should protect these commands with passwords to prevent unauthorized use.

You must enter the correct password to gain access to privileged EXEC commands. When you are running from the boot ROM monitor, the enable password might be the correct one to use, depending on your boot ROM level.

The enable and enable secret passwords need to be different for effective security. You can enter the same password for both enable and enable secret during the setup script, but you receive a warning message indicating that you should enter a different password.

Note An enable secret password can contain from 1 to 25 uppercase and lowercase alphanumeric characters; an enable password can contain any number of uppercase and lowercase alphanumeric characters. In both cases, a number cannot be the first character. Spaces are also valid password characters; for example, “two words” is a valid password. Leading spaces are ignored; trailing spaces are recognized.

Step 6 Enter the virtual terminal password when the following is displayed (remember this password for future reference):

```
The virtual terminal password is used to protect
access to the router over a network interface.
Enter virtual terminal password: bambam
```

Step 7 In most cases you will use IP routing. If so, you must also select an interior routing protocol, for example, the Enhanced Interior Gateway Routing Protocol (EIGRP).

Enter **yes** (the default) or press **Return** to configure IP, and then select EIGRP:

```
Configure IP? [yes]:
Configure EIGRP routing? [yes]:
Your IGRP autonomous system number [1]: 301
```

Step 8 Enter **yes** or **no** to accept or refuse SNMP management:

```
Configure SNMP Network Management? [yes]:
Community string [public]:
```

For complete SNMP information and procedures, refer to these publications:

- *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2, "Cisco IOS System Management," "Configuring SNMP Support," at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf014.html

- *Cisco IOS Configuration Fundamentals Configuration Command Reference*, Release 12.2, at this URL:

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

To provide a review of what you have done, a display similar to the following appears and lists all of the configuration parameters you selected in Steps 3 through 8. These parameters and their defaults are shown in the order in which they appeared on your console terminal:

```
The following configuration command script was created:

hostname router
enable secret 5 $1$S3Lx$uiTYg2UrFK1U0dgWdjvxw.
enable password lab
line vty 0 4
password lab
no snmp-server
!
ip routing eigrp 301

!
interface Vlan1
shutdown
no ip address
!
interface GigabitEthernet1/1
shutdown
no ip address
!
interface GigabitEthernet1/2
shutdown
no ip address
!
.
<...output truncated...>
```

```
.!
end


[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]: 2
% You can enter the setup, by typing setup at IOS command prompt
Router#
```

This completes the procedure on how to configure global parameters. The setup facility continues with the process to configure interfaces in the next section "Configuring Interfaces."

Configuring Interfaces

This section provides steps for configuring installed interfaces (using the setup facility or **setup** command facility) to allow communication over your external networks. To configure the interface parameters, you need your interface network addresses, subnet mask information, and which protocols you want to configure. (For additional interface configuration information on each of the modules available, refer to the individual configuration notes that shipped with your modules.)

Note The examples in this section are intended as examples only. Your configuration might look differently depending on your system configuration.

To configure interfaces, follow these steps:

Step 1 At the prompt for the Gigabit Ethernet interface configuration, enter the appropriate responses for your requirements, using your own address and subnet mask:

```
Do you want to configure GigabitEthernet1/1 interface? [no]: yes
  Configure IP on this interface? [no]: yes
    IP address for this interface: 172.20.52.34
    Subnet mask for this interface [255.255.0.0] : 255.255.255.224
    Class B network is 172.20.0.0, 27 subnet bits; mask is /27
```

Step 2 At the prompt for all other interface types, enter the appropriate responses for your requirements:

```
Do you want to configure FastEthernet5/1  interface? [no]: y
  Configure IP on this interface? [no]: y
    IP address for this interface: 172.20.52.98
    Subnet mask for this interface [255.255.0.0] : 255.255.255.248
    Class B network is 172.20.0.0, 29 subnet bits; mask is /29
```

Repeat this step for each interface you need to configure. Proceed to Step 3 to check and verify your configuration parameters.

When you reach and respond to the configuration dialog for the last installed interface, your interface configuration is complete.

Step 3 Check and verify the entire list of configuration parameters, which should display on your console terminal and end with the following query:

```
Use this configuration? [yes/no]:
```

A **no** response returns you to the enable prompt (#). You will need to reenter the **setup** command to reenter your configuration. A **yes** response saves the running configuration to NVRAM as follows:

```
Use this configuration? [yes/no]: yes
[OK]
Use the enabled mode 'configure' command to modify this configuration.
Press RETURN to get started!
```

After you press the **Return** key, this prompt appears:

```
Router>
```

This completes the procedures for configuring global parameters and interface parameters in your system. Your interfaces are now available for limited use.

If you want to modify the currently saved configuration parameters after the initial configuration, enter the **setup** command. To perform more complex configurations, enter configuration mode and use the **configure** command. Check the current state of the switch using the **show version** command, which displays the software version and the interfaces, as follows:

```
Router# show version
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, 33 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
Router#
```

For detailed interface configuration information, refer to the *Cisco IOS Interface Configuration Guide* at this URL:

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

Using Configuration Mode

If you prefer not to use the setup facility, you can configure the switch from configuration mode as follows:

Step 1 Connect a console terminal to the console interface of your supervisor engine.

Step 2 When you are asked if you want to enter the initial dialog, answer **no** to enter the normal operating mode as follows:

```
Would you like to enter the initial dialog? [yes]: no
```

Step 3 After a few seconds you will see the user EXEC prompt (`Router>`). Type **enable** to enter enable mode:

```
Router> enable
```

Note Configuration changes can only be made in enable mode.

The prompt will change to the privileged EXEC prompt (#) as follows:

```
Router#
```

Step 4 At the prompt (#), enter the **configure terminal** command to enter configuration mode as follows:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

At the prompt, enter the **interface** *type slot/interface* command to enter interface configuration mode as follows:

```
Router(config)# interface fastethernet 5/1
Router(config-if)#
```

In either of these configuration modes, you can enter any changes to the configuration. Enter the **end** command to exit configuration mode.

Step 5 Save your settings. (See the "Saving the Running Configuration Settings" section on page 3-11.)

Your switch is now minimally configured and can boot with the configuration you entered. To see a list of the configuration commands, enter **?** at the prompt or press the **help** key in configuration mode.

Checking the Running Configuration Before Saving

You can check the configuration settings you entered or changes you made by entering the **show running-config** command at the privileged EXEC prompt (#) as follows:

```
Router# show running-config
Building configuration...

Current configuration:
Current configuration : 3441 bytes
!
version 12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
```

```
no service password-encryption
!
hostname Router
!
boot buffersize 522200
boot system flash disk0:c6sup22-jsv-mz.121-5c.EX.bin
enable password lab
!
redundancy
 main-cpu
  auto-sync standard
ip subnet-zero
no ip finger
!
cns event-service server
!
<...output truncated...>
!
interface FastEthernet3/3
 ip address 172.20.52.19 255.255.255.224
!
<...output truncated...>
!
line con 0
 exec-timeout 0 0
 transport input none
line vty 0 4
 exec-timeout 0 0
 password lab
 login
 transport input lat pad mop telnet rlogin udptn nasi
!
end
Router#
```

Saving the Running Configuration Settings

To store the configuration or changes to your startup configuration in NVRAM, enter the **copy running-config startup-config** command at the privileged EXEC prompt (#) as follows:

```
Router# copy running-config startup-config
```

This command saves the configuration settings that you created in configuration mode. If you fail to do this step, your configuration will be lost the next time you reload the system.

Reviewing the Configuration

To display information stored in NVRAM, enter the **show startup-config** EXEC command. The display should be similar to the display from the **show running-config** EXEC command.

Configuring a Static Route

If your Telnet station or SNMP network management workstation is on a different network from your switch and a routing protocol has not been configured, you might need to add a static routing table entry for the network where your end station is located.

To configure a static route, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip route dest_IP_address mask {forwarding_IP | vlan vlan_ID}`	Configures a static route.
Step 2	`Router# show running-config`	Verifies the static route configuration.

This example shows how to use the **ip route** command to configure a static route to a workstation at IP address 171.10.5.10 on the switch with a subnet mask and IP address 172.20.3.35 of the forwarding router:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip route 171.10.5.10 255.255.255.255 172.20.3.35
Router(config)# end
Router#
```

This example shows how to use the **show running-config** command to confirm the configuration of the previously configured static route:

```
Router# show running-config
Building configuration...
.
<...output truncated...>
.
ip classless
ip route 171.10.5.10 255.255.255.255 172.20.3.35
no ip http server
!
line con 0
 transport input none
line vty 0 4
 exec-timeout 0 0
 password lab
 login
 transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end

Router#
```

This example shows how to use the **ip route** command to configure a static route to a workstation at IP address 171.20.5.3 on the switch with subnet mask and connected over VLAN 1:

```
Router# configure terminal
Router(config)# ip route 171.20.5.3 255.255.255.255 vlan 1
Router(config)# end
Router#
```

This example shows how to use the **show running-config** command to confirm the configuration of the previously configured static route:

```
Router# show running-config
Building configuration...
.
<...output truncated...>
.
ip classless
ip route 171.20.52.3 255.255.255.255 Vlan1
no ip http server
!
```

```
!
x25 host z
!
line con 0
 transport input none
line vty 0 4
 exec-timeout 0 0
 password lab
 login
 transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end

Router#
```

Configuring a BOOTP Server

The Bootstrap Protocol (BOOTP) automatically assigns an IP address by adding the MAC and IP addresses of the interface to the BOOTP server configuration file. When the switch boots, it automatically retrieves the IP address from the BOOTP server.

The switch performs a BOOTP request *only* if the current IP address is set to 0.0.0.0. (This address is the default address for a new switch or a switch that has had its startup-config file cleared using the **erase** command.)

To allow your switch to retrieve its IP address from a BOOTP server, you must first determine the MAC address of the switch and add that MAC address to the BOOTP configuration file on the BOOTP server. To create a BOOTP server configuration file, follow these steps:

Step 1 Install the BOOTP server code on the workstation, if it is not already installed.

Step 2 Determine the MAC address from the label on the chassis.

Step 3 Add an entry in the BOOTP configuration file (usually /usr/etc/bootptab) for each switch. Press **Return** after each entry to create a blank line between each entry. See the example BOOTP configuration file that follows in Step 4.

Step 4 Enter the **reload** command to reboot and automatically request the IP address from the BOOTP server.

This example BOOTP configuration file shows the added entry:

```
# /etc/bootptab: database for bootp server (/etc/bootpd)
#
# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#
#       first field -- hostname
#                       (may be full domain name and probably should be)
#
#       hd -- home directory
#       bf -- bootfile
#       cs -- cookie servers
#       ds -- domain name servers
#       gw -- gateways
#       ha -- hardware address
#       ht -- hardware type
#       im -- impress servers
#       ip -- host IP address
#       lg -- log servers
#       lp -- LPR servers
#       ns -- IEN-116 name servers
```

```
#       rl -- resource location protocol servers
#       sm -- subnet mask
#       tc -- template host (points to similar host entry)
#       to -- time offset (seconds)
#       ts -- time servers
#
<information deleted>
#
#######################################################################
# Start of individual host entries
#######################################################################
Router:         tc=netcisco0:   ha=0000.0ca7.ce00:      ip=172.31.7.97:
dross:          tc=netcisco0:   ha=00000c000139:        ip=172.31.7.26:
<information deleted>
```

Protecting Access to Privileged EXEC Commands

The following tasks provide a way to control access to the system configuration file and privileged EXEC commands:

- Setting or Changing a Static Enable Password, page 3-14
- Using the enable password and enable secret Commands, page 3-15
- Setting or Changing a Line Password, page 3-15
- Setting TACACS+ Password Protection for Privileged EXEC Mode, page 3-16
- Encrypting Passwords, page 3-16
- Configuring Multiple Privilege Levels, page 3-17

Setting or Changing a Static Enable Password

To set or change a static password that controls access to the privileged EXEC mode, perform this task:

Command	Purpose
`Router(config)# **enable password** *password*`	Sets a new password or changes an existing password for the privileged EXEC mode.

This example shows how to configure an enable password as “lab” at the privileged EXEC mode:

```
Router# configure terminal
Router(config)# enable password lab
Router(config)#
```

To display the password or access level configuration, see the “Displaying the Password, Access Level, and Privilege Level Configuration” section on page 3-18.

Using the enable password and enable secret Commands

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a TFTP server, you can use either the **enable password** or **enable secret** commands. Both commands configure an encrypted password that you must enter to access enable mode (the default) or to access a specified privilege level. We recommend that you use the **enable secret** command.

If you configure the **enable secret** command, it takes precedence over the **enable password** command; the two commands cannot be in effect simultaneously.

To configure the switch to require an enable password, perform either of these tasks:

Command	Purpose
`Router(config)# enable password [level level] {password | encryption-type encrypted-password}`	Establishes a password for the privileged EXEC mode.
`Router(config)# enable secret [level level] {password | encryption-type encrypted-password}`	Specifies a secret password, saved using a nonreversible encryption method. (If **enable password** and **enable secret** commands are both set, users must enter the enable secret password.)

Use either of these commands with the **level** option to define a password for a specific privilege level. After you specify the level and set a password, give the password only to users who need to have access at this level. Use the **privilege level** configuration command to specify commands accessible at various levels.

If you enable the **service password-encryption** command, the password you enter is encrypted. When you display it with the **more system:running-config** command, it displays in encrypted form.

If you specify an encryption type, you must provide an encrypted password that you copy from another Catalyst 6500 series switch configuration.

Note You cannot recover a lost encrypted password. You must clear NVRAM and set a new password. See the "Recovering a Lost Enable Password" section on page 3-18 if you lose or forget your password.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-18.

Setting or Changing a Line Password

To set or change a password on a line, perform this task:

Command	Purpose
`Router(config-line)# password password`	Sets a new password or change an existing password for the privileged level.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-18.

Setting TACACS+ Password Protection for Privileged EXEC Mode

For complete information about TACACS+, refer to these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.2, "Authentication, Authorization, and Accounting (AAA)," at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfaaa.html
- *Cisco IOS Security Command Reference*, Release 12.2, at this URL:

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

To set the TACACS+ protocol to determine whether or not a user can access privileged EXEC mode, perform this task:

Command	Purpose
`Router(config)# enable use-tacacs`	Sets the TACACS-style user ID and password-checking mechanism for the privileged EXEC mode.

When you set TACACS password protection at the privileged EXEC mode, the **enable** EXEC command prompts for both a new username and a password. This information is then sent to the TACACS+ server for authentication. If you are using the extended TACACS+, it also sends any existing UNIX user identification code to the TACACS+ server.

Caution If you enter the **enable use-tacacs** command, you must also enter **tacacs-server authenticate enable**, or you are locked out of the privileged EXEC mode.

Note When used without extended TACACS, the **enable use-tacacs** command allows anyone with a valid username and password to access the privileged EXEC mode, creating a potential security problem. This problem occurs because the switch cannot tell the difference between a query resulting from entering the **enable** command and an attempt to log in without extended TACACS.

Encrypting Passwords

Because protocol analyzers can examine packets (and read passwords), you can increase access security by configuring the Cisco IOS software to encrypt passwords. Encryption prevents the password from being readable in the configuration file.

To configure the Cisco IOS software to encrypt passwords, perform this task:

Command	Purpose
`Router(config)# service password-encryption`	Encrypts a password.

Encryption occurs when the current configuration is written or when a password is configured. Password encryption is applied to all passwords, including authentication key passwords, the privileged command password, console and virtual terminal line access passwords, and Border Gateway Protocol (BGP) neighbor passwords. The **service password-encryption** command keeps unauthorized individuals from viewing your password in your configuration file.

Caution The **service password-encryption** command does not provide a high level of network security. If you use this command, you should also take additional network security measures.

Although you cannot recover a lost encrypted password (that is, you cannot get the original password back), you can regain control of the switch after you lose or forget the encrypted password. See the "Recovering a Lost Enable Password" section on page 3-18 if you lose or forget your password.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-18.

Configuring Multiple Privilege Levels

By default, the Cisco IOS software has two modes of password security: user EXEC mode and privileged EXEC mode. You can configure up to 16 hierarchical levels of commands for each mode. By configuring multiple passwords, you can allow different sets of users to have access to specified commands.

For example, if you want many users to have access to the **clear line** command, you can assign it level 2 security and distribute the level 2 password widely. If you want more restricted access to the **configure** command, you can assign it level 3 security and distribute that password to more restricted users.

These tasks describe how to configure additional levels of security:

Setting the Privilege Level for a Command

To set the privilege level for a command, perform this task:

	Command	Purpose
Step 1	`Router(config)# privilege mode level level command`	Sets the privilege level for a command.
Step 2	`Router(config)# enable password level level [encryption-type] password`	Specifies the enable password for a privilege level.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-18.

Changing the Default Privilege Level for Lines

To change the default privilege level for a given line or a group of lines, perform this task:

Command	Purpose
`Router(config-line)# privilege level level`	Changes the default privilege level for the line.

To display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-18.

Logging In to a Privilege Level

To log in at a specified privilege level, perform this task:

Command	Purpose
`Router# enable level`	Logs into a specified privilege level.

Exiting a Privilege Level

To exit to a specified privilege level, perform this task:

Command	Purpose
`Router# disable level`	Exits to a specified privilege level.

Displaying the Password, Access Level, and Privilege Level Configuration

To display the password, access level, and privilege level configuration, perform this task:

	Command	Purpose
Step 1	`Router# show running-config`	Displays the password and the access level configuration.
Step 2	`Router# show privilege`	Shows the privilege level configuration.

This example shows how to display the password and access level configuration:

```
Router# show running-config
<...output truncated...>
enable password lab
<...output truncated...>
```

This example shows how to display the privilege level configuration:

```
Router# show privilege
Current privilege level is 15
Router#
```

Recovering a Lost Enable Password

To recover a lost enable password, follow these steps:

Step 1 Connect to the console interface.

Step 2 Configure the switch to boot up without reading the configuration memory (NVRAM).

Step 3 Reboot the system.

Step 4 Access enable mode (which can be done without a password when one is not configured).

Step 5 View or change the password, or erase the configuration.

Step 6 Reconfigure the switch to boot up and read the NVRAM as it normally does.

Step 7 Reboot the system.

Note Password recovery requires the Break signal. You must be familiar with how your terminal or PC terminal emulator issues this signal. For example, in ProComm, the Alt-B keys generate the Break signal. In a Windows terminal session, you press the **Break** or **Ctrl** and **Break** keys simultaneously.

Modifying the Supervisor Engine Startup Configuration

These sections describe how the startup configuration on the supervisor engine works and how to modify the configuration register and BOOT variable:

Understanding the Supervisor Engine Boot Configuration

These next sections describe how the boot configuration works on the supervisor engine.

Understanding the Supervisor Engine Boot Process

The supervisor engine boot process involves two software images: ROM monitor and supervisor engine software. When the switch is powered up or reset, the ROM-monitor code is executed. Depending on the NVRAM configuration, the supervisor engine either stays in ROM-monitor mode or loads the supervisor engine software.

Two user-configurable parameters determine how the switch boots: the configuration register and the BOOT environment variable. The configuration register is described in the "Modifying the Boot Field and Using the boot Command" section on page 3-21. The BOOT environment variable is described in the "Specifying the Startup System Image" section on page 3-23.

Understanding the ROM Monitor

The ROM monitor executes upon power-up, reset, or when a fatal exception occurs. The switch enters ROM-monitor mode if the switch 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 ROM-monitor mode, you can manually load a software image from bootflash or a Flash PC card.

Note For complete syntax and usage information for the ROM monitor commands, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, publication.

You can also enter ROM-monitor mode by restarting and then pressing the **Break** key during the first 60 seconds of startup. If you are connected through a terminal server, you can escape to the Telnet prompt and enter the **send break** command to enter ROM-monitor mode.

Note The **Break** key is always enabled for 60 seconds after rebooting, regardless of whether the configuration-register setting has the **Break** key disabled.

The ROM monitor has these features:

- Power-on confidence test
- Hardware initialization
- Boot capability (manual boot and autoboot)
- Debug utility and crash analysis
- Monitor call interface (EMT calls—the ROM monitor provides information and some functionality to the running software images through EMT calls)
- File system (the ROM monitor knows the simple file system and supports the newly developed file system through the dynamic linked file system library [MONLIB])
- Exception handling

Configuring the Software Configuration Register

The switch uses a 16-bit software configuration register, which allows you to set specific system parameters. Settings for the software configuration register are written into NVRAM.

Following are some reasons for changing the software configuration register settings:

- To select a boot source and default boot filename.
- To enable or disable the Break function.
- To control broadcast addresses.
- To set the console terminal baud rate.
- To load operating software from flash memory.
- To recover a lost password.
- To allow you to manually boot the system using the **boot** command at the bootstrap program prompt.
- To force an automatic boot from the system bootstrap software (boot image) or from a default system image in onboard flash memory, and read any **boot system** commands that are stored in the configuration file in NVRAM.

Table 3-2 lists the meaning of each of the software configuration memory bits, and Table 3-3 defines the boot field.

Caution The recommended configuration register setting is 0x2102. If you configure a setting that leaves break enabled and you send a break sequence over a console connection, the switch drops into ROMMON.

Table 3-2 Software Configuration Register Bit Meaning

Bit Number[1]	Hexadecimal	Meaning
00 to 03	0x0000 to 0x000F	Boot field (see Table 3-3)
06	0x0040	Causes system software to ignore NVRAM contents
07	0x0080	OEM[2] bit enabled
08	0x0100	Break disabled
09	0x0200	Use secondary bootstrap
10	0x0400	Internet Protocol (IP) broadcast with all zeros
11 to 12	0x0800 to 0x1000	Console line speed (default is 9600 baud)
13	0x2000	Boot default flash software if network boot fails
14	0x4000	IP broadcasts do not have network numbers
15	0x8000	Enable diagnostic messages and ignore NVRAM contents

1. The factory default value for the configuration register is 0x2102.
2. OEM = original equipment manufacturer.

Table 3-3 Explanation of Boot Field (Configuration Register Bits 00 to 03)

Boot Field	Meaning
00	Stays at the system bootstrap prompt
01	Boots the first system image in onboard flash memory
02 to 0F	Specifies a default filename for booting over the network; enables boot system commands that override the default filename

Modifying the Boot Field and Using the boot Command

The configuration register boot field determines whether or not the switch loads an operating system image, and if so, where it obtains this system image. The following sections describe using and setting the configuration register boot field, and the tasks you must perform to modify the configuration register boot field.

Bits 0 through 3 of the software configuration register form the boot field.

Note The factory default configuration register setting for systems and spares is 0x2102.

When the boot field is set to either 0 or 1 (0-0-0-0 or 0-0-0-1), the system ignores any boot instructions in the system configuration file and the following occurs:

- When the boot field is set to 0, you must boot the operating system manually by entering the **boot** command to the system bootstrap program or ROM monitor.
- When the boot field is set to 1, the system boots the first image in the onboard bootflash single in-line memory module (SIMM).
- When the entire boot field equals a value between 0-0-1-0 and 1-1-1-1, the switch loads the system image specified by **boot system** commands in the startup configuration file.

You can enter the **boot** command only, or enter the command and include additional boot instructions, such as the name of a file stored in flash memory, or a file that you specify for booting from a network server. If you use the **boot** command without specifying a file or any other boot instructions, the system boots from the default flash image (the first image in onboard flash memory). Otherwise, you can instruct the system to boot from a specific flash image (using the **boot system flash** *filename* command).

You can also use the **boot** command to boot images stored in the Flash PC cards located in Flash PC card slot 0 or slot 1 on the supervisor engine. If you set the boot field to any bit pattern other than 0 or 1, the system uses the resulting number to form a filename for booting over the network.

You must set the boot field for the boot functions you require.

Modifying the Boot Field

You modify the boot field from the software configuration register. To modify the software configuration register boot field, perform this task:

	Command	Purpose
Step 1	`Router# show version`	Determines the current configuration register setting.
Step 2	`Router# configure terminal`	Enters configuration mode, selecting the terminal option.
Step 3	`Router(config)# config-register value`	Modifies the existing configuration register setting to reflect the way in which you want the switch to load a system image.
Step 4	`Router(config)# end`	Exits configuration mode.
Step 5	`Router# reload`	Reboots to make your changes take effect.

To modify the configuration register while the switch is running Cisco IOS, follow these steps:

Step 1 Enter the **enable** command and your password to enter privileged level as follows:

```
Router> enable
Password:
Router#
```

Step 2 Enter the **configure terminal** command at the EXEC mode prompt (#) as follows:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Step 3 Configure the configuration register to0x2102 as follows:

```
Router(config)# config-register 0x2102
```

Set the contents of the configuration register by entering the **config-register** *value* configuration command, where *value* is a hexadecimal number preceded by 0x (see Table 3-2 on page 3-21).

Step 4 Enter the **end** command to exit configuration mode. The new value settings are saved to memory; however, the new settings do not take effect until the system software is reloaded by rebooting the system.

Step 5 Enter the **show version** EXEC command to display the configuration register value currently in effect and that will be used at the next reload. The value is displayed on the last line of the screen display, as in this example:

```
Configuration register is 0x141 (will be 0x2102 at next reload)
```

Step 6 Save your settings.

See the "Saving the Running Configuration Settings" section on page 3-11. However, note that configuration register changes take effect only after the system reloads, such as when you enter a **reload** command from the console.

Step 7 Reboot the system.

The new configuration register value takes effect with the next system boot.

Verifying the Configuration Register Setting

Enter the **show version** EXEC command to verify the current configuration register setting. In ROM-monitor mode, enter the **o** command to verify the value of the configuration register boot field.

To verify the configuration register setting, perform this task:

Command	Purpose
`Router# show version | include Configuration register`	Displays the configuration register setting.

In this example, the **show version** command indicates that the current configuration register is set so that the switch does not automatically load an operating system image. Instead, it enters ROM-monitor mode and waits for user-entered ROM monitor commands. The new setting instructs the switch to load a system image from commands in the startup configuration file or from a default system image stored on a network server.

```
Router1# show version | include Configuration register
Configuration register is 0x2102
Router#
```

Specifying the Startup System Image

You can enter multiple boot commands in the startup configuration file or in the BOOT environment variable to provide backup methods for loading a system image.

Note

- Store the system software image in the **sup-bootdisk:** or **disk0:**.
- Do not store the system software image in the **bootdisk:** device, which is on the PISA and is not accessible at boot time.

The BOOT environment variable is also described in the "Specify the Startup System Image in the Configuration File" section in the "Loading and Maintaining System Images and Microcode" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Understanding Flash Memory

The following sections describe flash memory:

- Flash Memory Features, page 3-24
- Security Features, page 3-24
- Flash Memory Configuration Process, page 3-24

Note The descriptions in the following sections applies to both the bootflash device and to removable flash memory cards.

Flash Memory Features

The flash memory components allow you to do the following:

- Copy the system image to flash memory using TFTP.
- Copy the system image to flash memory using rcp.
- Boot the system from flash memory either automatically or manually.
- Copy the flash memory image to a network server using TFTP or rcp.
- Boot manually or automatically from a system software image stored in flash memory.

Security Features

The flash memory components support the following security features:

- Flash memory cards contain a write-protect switch that you can use to protect data. You must set the switch to *unprotected* to write data to the Flash PC card.
- The system image stored in flash memory can be changed only from privileged EXEC level on the console terminal.

Flash Memory Configuration Process

To configure your switch to boot from flash memory, follow these steps:

Step 1 Copy a system image to flash memory using TFTP or rcp (refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2, "Cisco IOS File Management," "Loading and Maintaining System Images," at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf008.html

Step 2 Configure the system to boot automatically from the file in flash memory. You might need to change the configuration register value. See the "Modifying the Boot Field and Using the boot Command" section on page 3-21, for more information on modifying the configuration register.

Step 3 Save your configurations.

Step 4 Power cycle and reboot your system to ensure that all is working as expected.

CONFIG_FILE Environment Variable

For class A flash file systems, the CONFIG_FILE environment variable specifies the file system and filename of the configuration file to use for initialization (startup). Valid file systems can include **nvram:**, **disk0:**, and **sup-bootflash:**.

For detailed file management configuration information, refer to 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

After you save the CONFIG_FILE environment variable to your startup configuration, the switch checks the variable upon startup to determine the location and filename of the configuration file to use for initialization.

The switch uses the NVRAM configuration during initialization when the CONFIG_FILE environment variable does not exist or when it is null (such as at first-time startup). If the switch detects a problem with NVRAM or a checksum error, the switch enters **setup** mode. See the "Using the Setup Facility or the setup Command" section on page 3-2 for more information on the **setup** command facility.

Controlling Environment Variables

Although the ROM monitor controls environment variables, you can create, modify, or view them with certain commands. To create or modify the BOOT, BOOTLDR, and CONFIG_FILE environment variables, use the **boot system**, **boot bootldr**, and **boot config** global configuration commands.

Refer to the "Specify the Startup System Image in the Configuration File" section in the "Loading and Maintaining System Images and Microcode" chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the BOOT environment variable. Refer to the "Specify the Startup Configuration File" section in the "Modifying, Downloading, and Maintaining Configuration Files" chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the CONFIG_FILE variable.

Note

- When you use the **boot system**, **boot bootldr**, and **boot config** global configuration commands, you affect only the running configuration. You must save the environment variable settings to your startup configuration to place the information under ROM monitor control and for the environment variables to function as expected. Enter the **copy system:running-config nvram:startup-config** command to save the environment variables from your running configuration to your startup configuration.
- Cisco IOS software supports the **boot bootldr** global configuration command and the ROM monitor supports the BOOTLDR environment variable, but because Release 12.2ZY does not require use of a bootloader image, there are no Release 12.2ZY bootloader images.

You can view the contents of the BOOT, BOOTLDR, and the CONFIG_FILE environment variables using the **show bootvar** command. This command displays the settings for these variables as they exist in the startup configuration as well as in the running configuration if a running configuration setting differs from a startup configuration setting.

This example shows how to check the BOOT, BOOTLDR, and the CONFIG_FILE environment variables:

```
Router# show bootvar
BOOT variable = disk0:c6sup22-jsv-mz.121-5c.EX.bin,1;
```

```
CONFIG_FILE variable does not exist
BOOTLDR variable = bootflash:c6msfc2-boot-mz.121-3a.E4
Configuration register is 0x2
Router#
```

To display the contents of the configuration file pointed to by the CONFIG_FILE environment variable, enter the **more nvram:startup-config** command.

CHAPTER 4

Configuring a Supervisor Engine 32 PISA

This chapter describes how to configure a Supervisor Engine 32 PISA (Supervisor Engine 32 with a programmable intelligent services accelerator) in a Catalyst 6500 series switch. This chapter contains these sections:

Note

- For complete syntax and usage information for the commands used in this chapter, see the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- This is the minimum required Supervisor Engine 32 PISA memory:
  - 512-MB DRAM on the Supervisor Engine 32
  - 1-GB DRAM on the PISA daughterboard
- Supervisor Engine 32 PISA has a PFC3B and operates in PFC3B mode.
- With a 3-slot or a 4-slot chassis, install the Supervisor Engine 32 PISA in either slot 1 or 2.
- With a 6-slot or a 9-slot chassis, install the Supervisor Engine 32 PISA in either slot 5 or 6.
- With a 13-slot chassis, install the Supervisor Engine 32 PISA in either slot 7 or 8.
- Supervisor Engine 32 PISA does not support switch fabric connectivity.
- For information about the hardware and software features supported by the Supervisor Engine 32 PISA, see the *Release Notes for Cisco IOS Release 12.2ZY on the Supervisor Engine 32 PISA* at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/release/notes/ol_13011.html

Flash Memory on a Supervisor Engine 32 PISA

The Supervisor Engine 32 PISA has these flash memory devices:

- **disk0:**
  - One external CompactFlash Type II slot
  - Supports CompactFlash Type II flash PC cards
- **sup-bootdisk:**
  - Supervisor Engine 32 256-MB internal CompactFlash flash memory
  - From the Supervisor Engine 32 ROMMON, it is **bootdisk:**
- **bootdisk:**
  - PISA 256-MB internal CompactFlash flash memory
  - Not accessible from the Supervisor Engine 32 ROMMON

Supervisor Engine 32 PISA Ports

These sections describe the ports on a Supervisor Engine 32 PISA:

- Supervisor Engine 32 PISA Management Ports, page 4-2
- Supervisor Engine 32 PISA Data Ports, page 4-2

Supervisor Engine 32 PISA Management Ports

The console port for the Supervisor Engine 32 PISA port is an EIA/TIA-232 (RS-232) port. The Supervisor Engine 32 PISA also has two Universal Serial Bus (USB) 2.0 ports that currently are not enabled.

Supervisor Engine 32 PISA Data Ports

The WS-S32-10GE-PISA has these ports:

- Ports 1 and 2: XENPAK 10 Gigabit Ethernet
- Port 3: 10/100/1000 Mbps RJ-45

Note

- To avoid unexpected application of QoS to the PISA EtherChannel, do not configure QoS on WS-S32-10GE-PISA ports.
- You can disable Port 3 and reallocate its port ASIC capacity to the PISA EtherChannel (see the "Configuring Full PISA EtherChannel Bandwidth" section on page 4-3 section).

The WS-S32-GE-PISA has these ports:

- Ports 1 through 8: Small form-factor pluggable (SFP) Gigabit Ethernet
- Port 9: 10/100/1000 Mbps RJ-45 port

Note

- To avoid unexpected application of QoS to the PISA EtherChannel, do not configure QoS on WS-S32-GE-PISA ports.
- You can disable port 9 and reallocate its port ASIC capacity to the PISA EtherChannel (see the "Configuring Full PISA EtherChannel Bandwidth" section on page 4-3 section).

Configuring Full PISA EtherChannel Bandwidth

A Supervisor Engine 32 PISA automatically creates an EtherChannel (port channel interface 256) that the Supervisor Engine 32 and the PISA daughterboard use to communicate with each other. By default, the PISA EtherChannel bandwidth is 1 Gbps. To increase the PISA EtherChannel bandwidth, you can disable supervisor engine ports and reallocate the port ASIC capacity to the PISA EtherChannel.

To increase the PISA EtherChannel bandwidth, perform this task:

	Command	Purpose

Step 1	`Router(config)# interface gigabitethernet slot/3`
	Or:
	`Router(config)# interface gigabitethernet slot/[8 | 9]`
		On a WS-S32-10GE-PISA, selects the Ethernet port to be configured.
		On a WS-S32-GE-PISA, selects the Ethernet port to be configured.
		Note On a WS-S32-GE-PISA, you can allocate both ports 8 and 9 to the PISA EtherChannel.

Step 2	`Router(config-if)# channel-group 256 mode on`	Disables the port and allocates its port ASIC capacity to the PISA EtherChannel.
	`Router(config-if)# no channel-group 256 mode on`	Reverts to the default port ASIC capacity allocation.

Note

- You cannot enter any configuration under port channel interface 256.
- After the port becomes a member of the PISA EtherChannel, only the **no channel-group 256 mode on** command has any effect on the port until the port is no longer a member of the PISA EtherChannel. While the port is a member of the PISA EtherChannel, all port configuration commands except the [**no**] **channel-group 256 mode on** command are ignored.
- The PISA EtherChannel MTU size is 4,096 bytes.

This example shows how to allocate the port ASIC capacity of port 3 to the PISA EtherChannel on a WS-S32-10GE-PISA that is installed in slot 5:

```
Router# configure terminal
Router(config)# interface gigabitethernet 5/3
Router(config-if)# channel-group 256 mode on
Router(config-if)# end
```

This example shows how to allocate the port ASIC capacity of port 9 to the PISA EtherChannel on a WS-S32-GE-PISA that is installed in slot 5:

```
Router# configure terminal
Router(config)# interface gigabitethernet 5/9
Router(config-if)# channel-group 256 mode on
Router(config-if)# end
```

Displaying PISA Platform Statistics

To display platform statistics for the Supervisor Engine 32 PISA, perform this task:

Command	Purpose
`Router# show platform pisa np` *`counter-type`* `counters`	Displays platform statistics for the Supervisor Engine 32 PISA. • *counter-type*—See Table 4-1 for the list of valid values.

Table 4-1 shows the available counters.

Table 4-1 PISA Platform Counters

Counter-type	Description
me *num*	Microengine information (valid *num* values are from 0 to 15)
acl	ACL counter information
all	All Supervisor Engine 32 PISA-specific counters
all pps	Packets per second (pps) for all Supervisor Engine 32 PISA-specific counters
fpm	Flexible packet matching (FPM) counters
mqc	Modular QoS CLI information
nbar	Network-based application recognition (NBAR) counter information
rx	Receive engine counters
tx	Transmit engine counters

For examples of the **show platform pisa np** command output, see the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

To clear platform counters for the Supervisor Engine 32 PISA, perform this task:

Command	Purpose

Cisco IOS Release 12.2(33)ZYA and earlier releases
`Router# clear platform pisa ixp counters counter-type`
 	
Cisco IOS Release 12.2(33)ZYA1 and later releases
`Router# clear platform pisa np counter-type counters`
	Displays platform statistics for the Supervisor Engine 32 PISA.
	• *counter-type*—See Table 4-1 for the list of valid values.

This example shows how to clear the ACL counters in Cisco IOS Release 12.2(33)ZYA1 and later releases:

```
Router# clear platform pisa np acl counters
```

CHAPTER 5

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).

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- For information about RPR redundancy, see Chapter 6, "Configuring RPR Supervisor Engine Redundancy."
- NSF with SSO does not support IPv6 multicast traffic.

This chapter consists of these sections:

Understanding NSF with SSO Supervisor Engine Redundancy

These sections describe supervisor engine redundancy using NSF with SSO:

NSF with SSO Supervisor Engine Redundancy Overview

Note When a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.

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 information about RPR modes, see Chapter 6, "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 line cards 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 line cards, the line cards 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 line cards 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 line cards 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 PISA 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 PISA of the active supervisor engine, and they receive routing updates from their neighbor routers. Routing protocols do not run on the PISA 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 check-pointed 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 HSRP is not SSO-aware, meaning state information is not maintained between the active and redundant supervisor engine during normal operation. HSRP and SSO can coexist but both features work independently. Traffic that relies on HSRP may switch to the HSRP standby in the event of a supervisor engine switchover.
- The Gateway Load Balancing Protocol (GLBP) is not SSO-aware, meaning state information is not maintained between the active and redundant supervisor engine during normal operation. GLBP and SSO can coexist but both features work independently. Traffic that relies on GLBP may switch to the GLBP standby in the event of a supervisor engine switchover.
- The Virtual Redundancy Routing Protocols (VRRP) is not SSO-aware, meaning state information is not maintained between the active and redundant 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 engine switchover.
- Multiprotocol Label Switching (MPLS) is not suported with Cisco NSF with SSO; however, MPLS and NSF with SSO can coexist. If NSF with SSO is configured in the same chassis with MPLS, the failover performance of MPLS protocols will be at least equivalent to RPR while the supported NSF with SSO protocols still retain the additional benefits of NSF with SSO.
- 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 PFC3B 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 5-9
- Redundancy Configuration Guidelines and Restrictions, page 5-9

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 5-9
- Hardware Configuration Guidelines and Restrictions, page 5-9
- Configuration Mode Restrictions, page 5-10

Redundancy Configuration Guidelines and Restrictions

These guidelines and restrictions apply to all redundancy modes:

- 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 supervisor engine and the PISA supports redundant configurations where the supervisor engines and PISA routers are identical. If they are not identical, one will boot first and become active and hold the other supervisor engine and PISA 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 5-19).

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.
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 and display the redundancy state:

```
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
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
Router#
```

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 mutlitcast 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.
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.

	Command	Purpose
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 **bootdisk:** device on a redundant supervisor engine:

```
Router# copy source_device:source_filename slavesup-bootdisk:target_filename
```

Enter this command to copy a file to the **bootdisk:** device on a redundant PISA:

```
Router# copy source_device:source_filename slavebootdisk:target_filename
```

CHAPTER 6

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, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- For information about nonstop forwarding (NSF) with stateful switchover (SSO), see Chapter 5, "Configuring NSF with SSO Supervisor Engine Redundancy."

This chapter consists of these sections:

Understanding RPR

These sections describe supervisor engine redundancy using RPR:

Supervisor Engine Redundancy Overview

Note When a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.

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 6-6.)

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 PISA and PFC3B 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 PISA (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 6-3
- Hardware Configuration Guidelines and Restrictions, page 6-3
- Configuration Mode Restrictions, page 6-4

Redundancy Guidelines and Restrictions

These guidelines and restrictions apply to RPR redundancy modes:

- 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 supervisor engine and the PISA supports redundant configurations where the supervisor engines and PISA routers are identical. If they are not identical, one will boot first and become active and hold the other supervisor engine and PISA 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 6-7).

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 in the startup-config must be set to autoboot (see the "Modifying the Boot Field" section on page 3-22).

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, page 6-4
- Synchronizing the Supervisor Engine Configurations, page 6-5
- Displaying the Redundancy States, page 6-5

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 mode. 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 mode is enabled.
Step 4	`Router# show redundancy states`	Displays the operating redundancy mode.

This example shows how to configure the system for RPR and display the redundancy state:

```
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
Router# show redundancy states
       my state = 13 -ACTIVE
     peer state = 1  -DISABLED
           Mode = Simplex
           Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Route Processor Redundancy
Redundancy Mode (Configured)  = Route Processor Redundancy
     Split Mode = Disabled
   Manual Swact = Disabled  Reason: Simplex mode
 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

Router#
```

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 Plus
Redundancy Mode (Configured)  = Route Processor Redundancy Plus
     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

Router#
```

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.

To perform an FSU, perform this task:

	Command	Purpose
Step 1	Router# **copy** *source_device*:*source_filename* {**disk0** | **disk1**}:*target_filename* Or: Router# **copy** *source_device*:*source_filename* **sup-bootflash:***target_filename* Or: Router# **copy** *source_device*:*source_filename* **slavedisk0:***target_filename* Or: Router# **copy** *source_device*:*source_filename* **slavesup-bootflash:***target_filename*	Copies the new Cisco IOS image to bootflash on both supervisor engines.
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 reset
Router# redundancy force-switchover
Router#
```

Copying Files to the Redundant Supervisor Engine

Use the following command to copy a file to the **disk0:** device on a redundant supervisor engine:

```
Router# copy source_device:source_filename slavedisk0:target_filename
```

Use the following command to copy a file to the **bootdisk:** device on a redundant supervisor engine:

```
Router# copy source_device:source_filename slavesup-bootdisk:target_filename
```

Use the following command to copy a file to the **bootdisk:** device on a redundant PISA:

```
Router# copy source_device:source_filename slavebootdisk:target_filename
```

CHAPTER 7

Configuring Interfaces

This chapter describes how to configure interfaces on the Catalyst 6500 series switches. This chapter consists of these sections:

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

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

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:
  - Ethernet (use the **ethernet** keyword)
  - Fast Ethernet (use the **fastethernet** keyword)
  - Gigabit Ethernet (use the **gigabitethernet** keyword)
  - 10-Gigabit Ethernet (use the **tengigabitethernet** keyword)

Note For WAN interfaces, refer to the configuration note for the WAN module.

- Slot number—The slot in which the module is installed. On the Catalyst 6500 series switch, slots are numbered starting with 1, from top to bottom.
- Port number—The physical port number on the module. On the Catalyst 6500 series switch, 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
Router#
```

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

Router#
```

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 7-16.

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* = **ethernet**, **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 7-5.
- 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.
`Router(config)# no define interface-range macro_name`	Deletes a macro.

1. *type* = **ethernet**, **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 7-7
- Configuring Jumbo Frame Support, page 7-10
- Configuring IEEE 802.3x Flow Control, page 7-13
- Configuring the Port Debounce Timer, page 7-14
- Adding a Description for an Interface, page 7-15

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 the Catalyst 6500 series switch to negotiate the speed and duplex mode between ports. If you decide to configure the port speed and duplex modes manually, consider the following information:

- If you set the Ethernet port speed to auto, the switch automatically sets the duplex mode to auto.
- If you enter the **no speed** command, the switch automatically configures both speed and duplex to auto.
- 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 Catalyst 6500 series switches 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.

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.
	`Router(config-if)# no speed`	Reverts to the default configuration (speed auto).

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.
	`Router(config-if)# no duplex`	Reverts to the default configuration (duplex auto).

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 7-1 shows the four possible link negotiation configurations and the resulting link status for each configuration.

Table 7-1 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.
	`Router(config-if)# no speed nonegotiate`	Reverts to the default configuration (link negotiation enabled).

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`	Displays the speed and duplex mode configuration.

1. *type* = **ethernet**, **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#
```

Configuring Jumbo Frame Support

These sections describe jumbo frame support:

- Understanding Jumbo Frame Support, page 7-10
- Configuring MTU Sizes, page 7-12

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 7-10
- Ethernet Ports, page 7-11
- VLAN Interfaces, page 7-12

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 PISA.
- 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 7-13).

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 PFC3B

For traffic that needs to be routed, Jumbo frame support on the PFC3B 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 PFC3B sends the traffic to the PISA to be fragmented and routed in software. If the "do not fragment bit" is set, the PFC3B 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 7-13).

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 7-13).

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 7-13).

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` `Router(config-if)# no mtu`	Configures the MTU size. Reverts to the default MTU size (1500 bytes).
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* = **ethernet**, **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 7-13).

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.
	`Router(config)# no system jumbomtu`	Reverts to the default global egress LAN port MTU size (9216 bytes).
Step 2	`Router(config)# end`	Exits configuration mode.

Configuring IEEE 802.3x Flow Control

Gigabit Ethernet and 10-Gigabit Ethernet ports on the Catalyst 6500 series switches 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 transmits an IEEE 802.3x pause frame that requests remote ports to delay sending frames for a specified time. All Ethernet ports (10 Gbps, 1 Gbps, 100 Mbps, and 10 Mbps) can receive and 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.
	`Router(config-if)# no flowcontrol {receive | send}`	Reverts to the default flow control settings.
Step 3	`Router# show interfaces [type[1] slot/port] flowcontrol`	Displays the flow-control configuration for all ports.

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

When configuring flow control, note the following information:

- 10-Gigabit Ethernet ports are permanently configured to respond to pause frames.
- When the configuration of the remote ports is unknown, use the **receive desired** keywords to configure a Gigabit Ethernet port to respond to received pause frames.
- Use the **receive on** keywords to configure a Gigabit Ethernet port to respond to received pause frames.
- Use the **receive off** keywords to configure a Gigabit Ethernet port to ignore received pause frames.
- When configuring transmission of pause frames, note the following information:
  - When the configuration of the remote ports is unknown, use the **send desired** keywords to configure a port to send pause frames.
  - 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, which can decrease traffic loss due to network reconfiguration. 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.
	`Router(config-if)# no link debounce`	Reverts to the default setting.
Step 3	`Router# show interfaces debounce`	Verifies the configuration.

1. *type* = **ethernet**, **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.
- 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 recognize 10 Gbps copper media and detects media-only changes.

Table 7-2 lists the time delay that occurs before notification of a link change.

Table 7-2 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	10 milliseconds[1]	100 milliseconds
WS-X6502-10GE 10-Gigabit ports	1000 milliseconds	3100 milliseconds

1. 10 milliseconds with Release 12.2(18)SXF13 and later releases.

Note On all 10-Gigabit Ethernet ports, the Debounce Timer Disabled is 10 milliseconds and the Debounce Timer Enabled is 1 second.

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.
`Router(config-if)# no description`	Deletes a description from 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, refer to 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 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 7-17
- Clearing Counters on an Interface, page 7-17
- Resetting an Interface, page 7-18
- Shutting Down and Restarting an Interface, page 7-18

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.
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* = **ethernet**, **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* = **ethernet**, **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* = **ethernet**, **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.

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.
```

Note

- TDR can test cables up to a maximum length of 115 meters.
- See the *Release Notes for Cisco IOS Release 12.2ZY on the Supervisor Engine 32 PISA* for information about which modules support the TDR.

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 #
```

CHAPTER 8

Configuring LAN Ports for Layer 2 Switching

This chapter describes how to use the command-line interface (CLI) to configure Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet LAN ports for Layer 2 switching on the Catalyst 6500 series switches. The configuration tasks in this chapter apply to LAN ports on LAN switching modules and to the LAN ports on the supervisor engine.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- To configure Layer 3 interfaces, see Chapter 19, "Configuring Layer 3 Interfaces."

This chapter consists of these sections:

Understanding How Layer 2 Switching Works

These sections describe how Layer 2 switching works on the Catalyst 6500 series switches:

Understanding Layer 2 Ethernet Switching

These sections describe Layer 2 Ethernet switching:

Layer 2 Ethernet Switching Overview

Catalyst 6500 series 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.

Catalyst 6500 series switches 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 LAN port on a Catalyst 6500 series switch 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 Address Table

Catalyst 6500 series switches build the address table by using the source address of the frames received. When the switch receives a frame for a destination 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 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.

The address table can store at least 32,000 address entries without flooding any entries. The switch uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

Understanding VLAN Trunks

These sections describe VLAN trunks on the Catalyst 6500 series switches:

Trunking Overview

Note For information about VLANs, see Chapter 12, "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 10, "Configuring EtherChannels."

Ethernet trunk ports support several trunking modes (see Table 8-2 on page 8-4). 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 11, "Configuring VTP."

Encapsulation Types

Table 8-1 lists the Ethernet trunk encapsulation types.

Table 8-1 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 8-3).

Table 8-1 Ethernet Trunk Encapsulation Types (continued)

Encapsulation	Function
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 8-2 lists the Layer 2 LAN port modes and describes how they function on LAN ports.

Table 8-2 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 8-3 shows the Layer 2 LAN port default configuration.

Table 8-3 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 12-1 on page 12-2)
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 on the Catalyst 6500 series switches:

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.
	`Router(config-if)# no switchport`	Clears Layer 2 LAN port configuration.
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* = **ethernet**, **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.

Configuring a Layer 2 Switching Port as a Trunk

These section 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 8-7 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.
`Router(config-if)# no switchport trunk encapsulation`	Reverts to the default trunk encapsulation mode (**negotiate**).

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 8-9) 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 8-12 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 8-7 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.
`Router(config-if)# no switchport mode`	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 8-2 on page 8-4 for information about trunking modes.

Note Complete the steps in the "Completing Trunk Configuration" section on page 8-12 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 8-7 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.
	`Router(config-if)# no switchport mode`	Reverts to the default trunk trunking mode (**switchport mode dynamic desirable**).
Step 2	`Router(config-if)# switchport nonegotiate`	(Optional) Configures the trunk not to use DTP.
	`Router(config-if)# no switchport nonegotiate`	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 8-8).

- To support the **switchport nonegotiate** command, you must enter the **switchport mode trunk** command.
- Enter the **switchport mode dynamic trunk** command. See Table 8-2 on page 8-4 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 8-8) 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 8-9).

Note Complete the steps in the "Completing Trunk Configuration" section on page 8-12 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 8-7 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 12-1 on page 12-2).
`Router(config-if)# no switchport access vlan`	Reverts to the default value (VLAN 1).

Note Complete the steps in the "Completing Trunk Configuration" section on page 8-12 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 8-7 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.
`Router(config-if)# no switchport trunk native vlan`	Reverts to the default value (VLAN 1).

When configuring the native VLAN, note the following information:

- The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 12-1 on page 12-2).
- The access VLAN is not automatically used as the native VLAN.

Note Complete the steps in the "Completing Trunk Configuration" section on page 8-12 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 8-7 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.
`Router(config-if)# no switchport trunk allowed vlan`	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.
- 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 8-12 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 8-7 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[,...]]}}`	(Optional) Configures the list of prune-eligible VLANs on the trunk (see the "Understanding VTP Pruning" section on page 11-3).
`Router(config-if)# no switchport trunk pruning vlan`	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 12-1 on page 12-2), 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 Catalyst 6500 series switches with 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 8-12 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* = **ethernet**, **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 12-10).

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)# no switchport`	Clears Layer 2 LAN port configuration.
Step 5	`Router(config-if)# switchport mode access` `Router(config-if)# no switchport mode`	Configures the LAN port as a Layer 2 access port. Reverts to the default switchport mode (**switchport mode dynamic desirable**).
Step 6	`Router(config-if)# switchport access vlan vlan_ID` `Router(config-if)# no switchport access vlan`	Places the LAN port in a VLAN. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 12-1 on page 12-2). Reverts to the default access VLAN (VLAN 1).
Step 7	`Router(config-if)# no shutdown`	Activates the interface. (Required only if you shut down the interface.)
Step 8	`Router(config-if)# end`	Exits configuration mode.
Step 9	`Router# show running-config interface [type[1] slot/port]`	Displays the running configuration of the interface.
Step 10	`Router# show interfaces [type[1] slot/port] switchport`	Displays the switch port configuration of the interface.

1. *type* = **ethernet**, **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 a Custom IEEE 802.1Q 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.
`Router(config-if)# no switchport dot1q ethertype`	Reverts to the default 802.1Q EtherType field value (0x8100).

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 12.2ZY on the Supervisor Engine 32 PISA* 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)#
```

CHAPTER 9

Configuring Flex Links

This chapter describes how to configure Flex Links on the Catalyst 6500 series switch.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

The chapter consists of these sections:

Understanding Flex Links

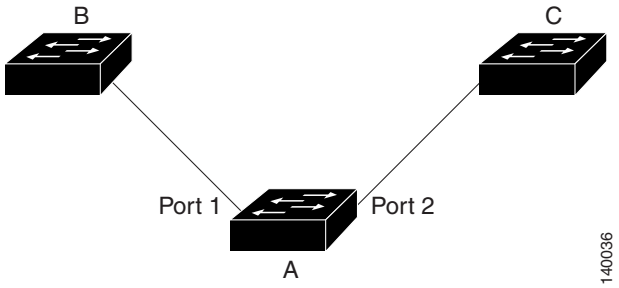
Flex Links are a pair of a Layer 2 interfaces (switchports or port channels), where one interface is configured to act as a backup to the other. Flex Links are typically configured in service-provider or enterprise networks where customers do not want to run STP. Flex Links provide link-level redundancy that is an alternative to Spanning Tree Protocol (STP). STP is automatically disabled on Flex Links interfaces.

The Catalyst 6500 series switches support a maximum of 16 Flex Links.

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 9-1, 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 9-1 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.

Flex Links are supported only on Layer 2 ports and port channels, not on VLANs or on Layer 3 ports.

Configuring Flex Links

These sections contain this configuration information:

- Flex Links Default Configuration, page 9-2
- Flex Links Configuration Guidelines and Restrictions, page 9-2
- Configuring Flex Links, page 9-3

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

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.
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* = **ethernet**, **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

Table 9-1 shows the privileged EXEC command for monitoring the Flex Links configuration.

Table 9-1 Flex Links Monitoring Command

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* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

CHAPTER 10

Configuring EtherChannels

This chapter describes how to configure EtherChannels on the Catalyst 6500 series switch Layer 2 or Layer 3 LAN ports.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding How EtherChannels Work

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. The maximumnumber of EtherChannels is 128.

You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6500 series switch. 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 Catalyst 6500 series 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 How EtherChannels Are Configured

These sections describe how EtherChannels are configured:

- EtherChannel Configuration Overview, page 10-2
- Understanding Manual EtherChannel Configuration, page 10-3
- Understanding PAgP EtherChannel Configuration, page 10-3
- Understanding IEEE 802.3ad LACP EtherChannel Configuration, page 10-3

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.

Table 10-1 lists the user-configurable EtherChannel modes.

Table 10-1 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.
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.

Understanding Manual EtherChannel Configuration

Manually configured EtherChannel ports do not exchange EtherChannel protocol packets. A manually configured EtherChannel forms only when you enter 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 10-10). 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 10-7). 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 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 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 pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel.

EtherChannel load balancing can use MAC addresses or IP addresses. EtherChannel load balancing can also use Layer 4 port numbers. EtherChannel load balancing can use either 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 use MPLS Layer 2 information.

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 LAN ports in Catalyst 6500 series switches, 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 Ethernet LAN 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.
- 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.
- 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 and without strict-priority queues.

Configuring EtherChannels

These sections describe how to configure EtherChannels:

- Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels, page 10-6
- Configuring Channel Groups, page 10-7
- Configuring EtherChannel Load Balancing, page 10-10
- Configuring the EtherChannel Min-Links Feature, page 10-11

Note Make sure that the LAN ports are configured correctly (see the "EtherChannel Feature Configuration Guidelines and Restrictions" section on page 10-5).

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 10-7).
- 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 10-7).
- 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 number`	Creates the port channel interface.
	`Router(config)# no interface port-channel number`	Deletes 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 10-6), 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.
	`Router(config-if)# no channel-protocol`	Removes the restriction.
Step 4	`Router(config-if)# channel-group number mode {active | auto | desirable | on | passive}`	Configures the LAN port in a port channel and specifies the mode (see Table 10-1 on page 10-2). PAgP supports only the auto and desirable modes. LACP supports only the active and passive modes.
	`Router(config-if)# no channel-group`	Removes the LAN port from the channel group.
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.
	`Router(config-if)# no lacp port-priority`	Reverts to the default.
Step 6	`Router(config-if)# end`	Exits configuration mode.
Step 7	`Router# show running-config interface type[1] slot/port` `Router# show interfaces type[1] slot/port etherchannel`	Verifies the configuration.

1. *type* = **ethernet**, **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)# end
```

Note See the "Configuring a Range of Interfaces" section on page 7-4 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.
	`Router(config)# no lacp system-priority`	Reverts to the default.
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 load-balance {src-mac | dst-mac | src-dst-mac | src-ip | dst-ip | src-dst-ip | src-port | dst-port | src-dst-port}`	Configures EtherChannel load balancing.
	`Router(config)# no port-channel load-balance`	Reverts to default EtherChannel load balancing.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show etherchannel load-balance`	Verifies the configuration.

The load-balancing 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**—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

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
Source XOR Destination IP address
Router#
```

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.

To configure the EtherChannel Min-Links feature, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface port-channel 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.
	`Router(config-if)# no port-channel min-links`	Reverts to the default number of active member ports (one).
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show running-config interface type[1] slot/port` `Router# show interfaces type[1] slot/port etherchannel`	Verifies the configuration.

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 2 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
```

CHAPTER 11

Configuring VTP

This chapter describes how to configure the VLAN Trunking Protocol (VTP) on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding How VTP Works

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 12, “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).

By default, the Catalyst 6500 series switch is in VTP server mode and is in the no-management domain state until the switch 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.

When you make a change to the VLAN configuration on a VTP server, the change is propagated to all network devices in the VTP domain. VTP advertisements are transmitted out all trunk connections.

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 a Catalyst 6500 series switch to operate in 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, transparent network devices do forward VTP advertisements that they receive out their trunking LAN ports.

Note Catalyst 6500 series switches automatically change 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 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 maximum transmission unit (MTU) size for each VLAN
- Frame format

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 How VLANs Work" section on page 12-1.
- Unrecognized Type-Length-Value (TLV) Support—A VTP server or client propagates configuration changes to its other trunks, even for TLVs 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 in the supervisor engine software, 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 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.

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 11-1 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 Catalyst 6500 series switch (see the "Enabling VTP Pruning" section on page 11-7). You configure pruning on Layer 2 trunking LAN ports (see the "Configuring a Layer 2 Switching Port as a Trunk" section on page 8-8).

Figure 11-1 Flooding Traffic without VTP Pruning

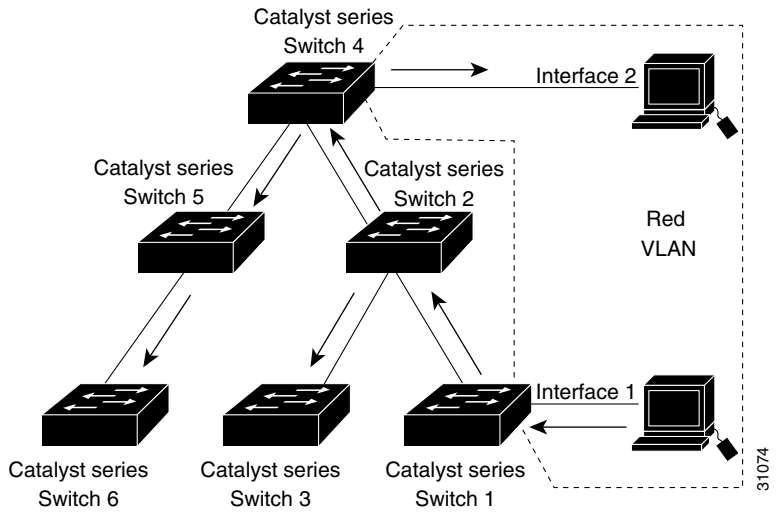
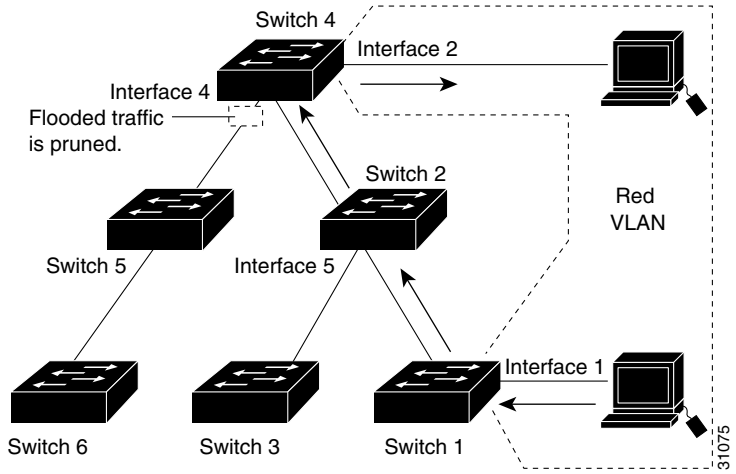


Figure 11-2 shows the same switched network with VTP pruning enabled. The broadcast traffic from Switch 1 is not forwarded to Switches 3, 5, and 6 because traffic for the Red VLAN has been pruned on the links indicated (port 5 on Switch 2 and port 4 on Switch 4).

Figure 11-2 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 8-8). 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.

VTP Default Configuration

Table 11-1 shows the default VTP configuration.

Table 11-1 VTP Default Configuration

Feature	Default Value
VTP domain name	Null
VTP mode	Server
VTP version 2 enable 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 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.
- 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 running VTP version 1 provided 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.
- 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 Catalyst 6500 series switches with 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 8-11.

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 a VTP Password

To configure the VTP 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.

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 8-11.

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#
```

Configuring the VTP Mode

To configure the VTP mode, perform this task:

	Command	Purpose
Step 1	`Router(config)# vtp mode {client | server | transparent}`	Configures the VTP mode.
	`Router(config)# no vtp mode`	Reverts to the default VTP mode (server).
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 verify the configuration:

```
Router# show vtp status
VTP Version                     : 2
Configuration Revision          : 247
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 33
VTP Operating Mode              : Server
VTP Domain Name                 : Lab_Network
VTP Pruning Mode                : Enabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x45 0x52 0xB6 0xFD 0x63 0xC8 0x49 0x80
Configuration last modified by 0.0.0.0 at 8-12-99 15:04:49
Local updater ID is 172.20.52.34 on interface Gi1/1 (first interface found)
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 verify the configuration:

```
Router# show vtp status
VTP Version                     : 2
Configuration Revision          : 247
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 33
VTP Operating Mode              : Client
VTP Domain Name                 : Lab_Network
VTP Pruning Mode                : Enabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x45 0x52 0xB6 0xFD 0x63 0xC8 0x49 0x80
Configuration last modified by 0.0.0.0 at 8-12-99 15:04:49
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 verify the configuration:

```
Router# show vtp status
VTP Version                     : 2
Configuration Revision          : 247
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 33
VTP Operating Mode              : Transparent
VTP Domain Name                 : Lab_Network
VTP Pruning Mode                : Enabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x45 0x52 0xB6 0xFD 0x63 0xC8 0x49 0x80
Configuration last modified by 0.0.0.0 at 8-12-99 15:04:49
Router#
```

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
```

CHAPTER 12

Configuring VLANs

This chapter describes how to configure VLANs on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding How VLANs Work

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 17-2).

Catalyst 6500 series switches support 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 12-1 describes the VLAN ranges.

Table 12-1 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).
- Switches running the Catalyst operating system do not support configuration of VLANs 1006–1024. If you configure VLANs 1006–1024, ensure that the VLANs do not extend to any switches running Catalyst software.
- You must enable the extended system ID to use extended range VLANs (see the "Understanding the Bridge ID" section on page 17-2).

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

Understanding Token Ring VLANs

The following section describes the two Token Ring VLAN types supported on network devices running VTP version 2:

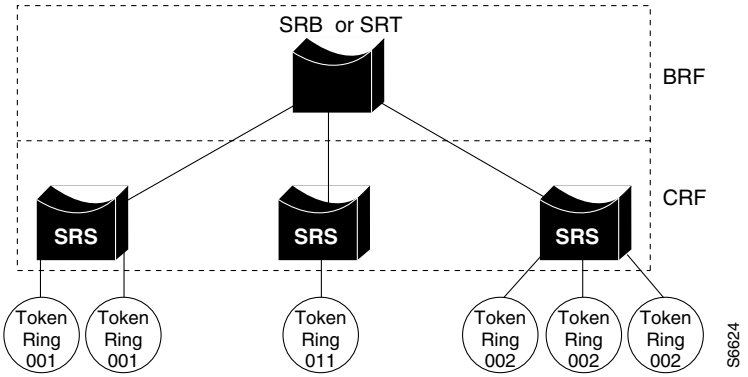
- Token Ring TrBRF VLANs, page 12-3
- Token Ring TrCRF VLANs, page 12-4

Note Catalyst 6500 series switches do not support Inter-Switch Link (ISL)-encapsulated Token Ring frames. When a Catalyst 6500 series switch is configured as a VTP server, you can configure Token Ring VLANs from the switch.

Token Ring TrBRF VLANs

Token Ring Bridge Relay Function (TrBRF) VLANs interconnect multiple Token Ring Concentrator Relay Function (TrCRF) VLANs in a switched Token Ring network (see Figure 12-1). The TrBRF can be extended across a network devices interconnected via trunk links. The connection between the TrCRF and the TrBRF is referred to as a *logical port*.

Figure 12-1 Interconnected Token Ring TrBRF and TrCRF VLANs



For source routing, the Catalyst 6500 series switch appears as a single bridge between the logical rings. The TrBRF can function as a source-route bridge (SRB) or a source-route transparent (SRT) bridge running either the IBM or IEEE STP. If an SRB is used, you can define duplicate MAC addresses on different logical rings.

The Token Ring software runs an instance of STP for each TrBRF VLAN and each TrCRF VLAN. For TrCRF VLANs, STP removes loops in the logical ring. For TrBRF VLANs, STP interacts with external bridges to remove loops from the bridge topology, similar to STP operation on Ethernet VLANs.

Caution Certain parent TrBRF STP and TrCRF bridge mode configurations can place the logical ports (the connection between the TrBRF and the TrCRF) of the TrBRF in a blocked state. For more information, see the "VLAN Configuration Guidelines and Restrictions" section on page 12-8.

To accommodate IBM System Network Architecture (SNA) traffic, you can use a combination of SRT and SRB modes. In a mixed mode, the TrBRF determines that some ports (logical ports connected to TrCRFs) operate in SRB mode while other ports operate in SRT mode

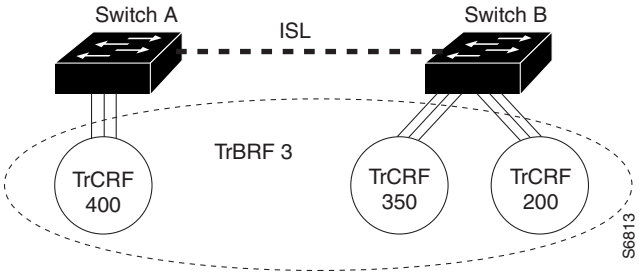
Token Ring TrCRF VLANs

Token Ring Concentrator Relay Function (TrCRF) VLANs define port groups with the same logical ring number. You can configure two types of TrCRFs in your network: undistributed and backup.

TrCRFs typically are undistributed, which means each TrCRF is limited to the ports on a single network device. Multiple undistributed TrCRFs on the same or separate network devices can be associated with a single parent TrBRF (see Figure 12-2). The parent TrBRF acts as a multiport bridge, forwarding traffic between the undistributed TrCRFs.

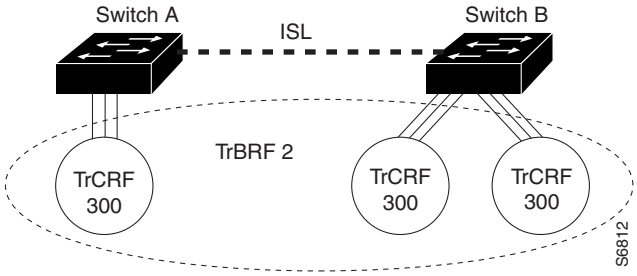
Note To pass data between rings located on separate network devices, you can associate the rings to the same TrBRF and configure the TrBRF for an SRB.

Figure 12-2 Undistributed TrCRFs



By default, Token Ring ports are associated with the default TrCRF (VLAN 1003, trcrf-default), which has the default TrBRF (VLAN 1005, trbrf-default) as its parent. In this configuration, a distributed TrCRF is possible (see Figure 12-3), and traffic is passed between the default TrCRFs located on separate network devices if the network devices are connected through an ISL trunk.

Figure 12-3 Distributed TrCRF



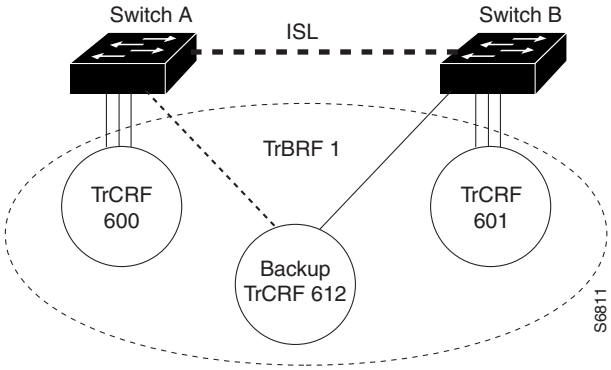
Within a TrCRF, source-route switching forwards frames based on either MAC addresses or route descriptors. The entire VLAN can operate as a single ring, with frames switched between ports within a single TrCRF.

You can specify the maximum hop count for All-Routes and Spanning Tree Explorer frames for each TrCRF. When you specify the maximum hop count, you limit the maximum number of hops an explorer is allowed to traverse. If a port determines that the explorer frame it is receiving has traversed more than the number of hops specified, it does not forward the frame. The TrCRF determines the number of hops an explorer has traversed by the number of bridge hops in the route information field.

If the ISL connection between network devices fails, you can use a backup TrCRF to configure an alternate route for traffic between undistributed TrCRFs. Only one backup TrCRF for a TrBRF is allowed, and only one port per network device can belong to a backup TrCRF.

If the ISL connection between the network devices fails, the port in the backup TrCRF on each affected network device automatically becomes active, rerouting traffic between the undistributed TrCRFs through the backup TrCRF. When the ISL connection is reestablished, all but one port in the backup TrCRF is disabled. Figure 12-4 illustrates the backup TrCRF.

Figure 12-4 Backup TrCRF



VLAN Default Configuration

Tables 12-2 through 12-6 show the default configurations for the different VLAN media types.

Table 12-2 Ethernet VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1	1–4094
VLAN name	“default” for VLAN 1 “VLAN*vlan_ID*” for other Ethernet VLANs	—
802.10 SAID	10*vlan_ID*	100001–104094
MTU size	1500	1500–18190
Translational bridge 1	0	0–1005
Translational bridge 2	0	0–1005
VLAN state	active	active, suspend
Pruning eligibility	VLANs 2–1001 are pruning eligible; VLANs 1006–4094 are not pruning eligible.	—

Table 12-3 FDDI VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1002	1–1005
VLAN name	“fddi-default”	—
802.10 SAID	101002	1–4294967294
MTU size	1500	1500–18190
Ring number	0	1–4095
Parent VLAN	0	0–1005
Translational bridge 1	0	0–1005

Table 12-3 FDDI VLAN Defaults and Ranges (continued)

Parameter	Default	Range
Translational bridge 2	0	0–1005
VLAN state	active	active, suspend

Table 12-4 Token Ring (TrCRF) VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1003	1–1005
VLAN name	“token-ring-default”	—
802.10 SAID	101003	1–4294967294
Ring Number	0	1–4095
MTU size	VTPv1 default 1500 VTPv2 default 4472	1500–18190
Translational bridge 1	0	0–1005
Translational bridge 2	0	0–1005
VLAN state	active	active, suspend
Bridge mode	srb	srb, srt
ARE max hops	7	0–13
STE max hops	7	0–13
Backup CRF	disabled	disable; enable

Table 12-5 FDDI-Net VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1004	1–1005
VLAN name	“fddinet-default”	—
802.10 SAID	101004	1–4294967294
MTU size	1500	1500–18190
Bridge number	1	0–15
STP type	ieee	auto, ibm, ieee
VLAN state	active	active, suspend

Table 12-6 Token Ring (TrBRF) VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1005	1–1005
VLAN name	“trnet-default”	—
802.10 SAID	101005	1–4294967294

Table 12-6 Token Ring (TrBRF) VLAN Defaults and Ranges (continued)

Parameter	Default	Range
MTU size	VTPv1 1500; VTPv2 4472	1500–18190
Bridge number	1	0–15
STP type	ibm	auto, ibm, ieee
VLAN state	active	active, suspend

VLAN Configuration Guidelines and Restrictions

When creating and modifying VLANs in your network, follow these guidelines and restrictions:

- 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.
- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode. See the "VLAN Configuration Options" section on page 12-9.
- Before you can create a VLAN, the Catalyst 6500 series switch must be in VTP server mode or VTP transparent mode. For information on configuring VTP, see Chapter 11, "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 *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, publication.
- To do a complete backup of your configuration, include the vlan.dat file in the backup.
- The Cisco IOS **end** command is not supported in VLAN database mode.
- You cannot enter **Ctrl-Z** to exit VLAN database mode.
- Catalyst 6500 series switches do not support Token Ring or FDDI media. The switch does not forward FDDI, FDDI-Net, TrCRF, or TrBRF traffic, but it can propagate the VLAN configuration through VTP.
- When a Catalyst 6500 series switch is configured as a VTP server, you can configure FDDI and Token Ring VLANs from the switch.
- You must configure a TrBRF before you configure the TrCRF (the parent TrBRF VLAN you specify must exist).
- In a Token Ring environment, the logical interfaces (the connection between the TrBRF and the TrCRF) of the TrBRF are placed in a blocked state if either of these conditions exists:
  - The TrBRF is running the IBM STP, and the TrCRF is in SRT mode.
  - The TrBRF is running the IEEE STP, and the TrCRF is in SRB mode.

Configuring VLANs

These sections describe how to configure VLANs:

Note VLANs support a number of parameters that are not discussed in detail in this section. For complete information, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference, Release 12.2ZY*, publication.

VLAN Configuration Options

These sections describe the VLAN configuration options:

VLAN Configuration in Global Configuration Mode

If the switch is in VTP server or transparent mode (see the "Configuring VTP" section on page 11-6), 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.

Note

- When the switch boots, if the VTP domain name and VTP mode in the startup-config 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. You cannot configure extended-range VLANs in VLAN database mode.

VLAN Configuration in VLAN Database Mode

Note You cannot configure extended-range VLANs in VLAN database mode. You can configure extended-range VLANs only in global configuration mode.

If the switch is in VTP server or transparent mode, you can configure VLANs in the VLAN database mode. When you configure VLANs in VLAN database mode, the VLAN configuration is saved in the vlan.dat files. To display the VLAN configuration, enter the **show vlan** command.

You use the interface configuration command mode to define the port membership mode and add and remove ports from a VLAN. The results of these commands are written to the running-config file, and you can display the file by entering the **show running-config** command.

Creating or Modifying an Ethernet VLAN

User-configured VLANs have unique IDs from 1 to 4094, except for reserved VLANs (see Table 12-1 on page 12-2). 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 "VLAN Default Configuration" section on page 12-6 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).

	`Router(config)# no vlan vlan_ID`
	`Router(config-vlan)#`
	or
	`Router(vlan)# no vlan vlan_ID`
		Deletes a VLAN.

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 8-6.

Configuring the Internal VLAN Allocation Policy

For more information about VLAN allocation, see the "VLAN Ranges" section on page 12-2.

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}` `Router(config)# no vlan internal allocation policy`	Configures the internal VLAN allocation policy. Returns to the default (ascending).
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:

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.
- The following table lists:
  - The modules that support VLAN translation
  - 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

Note LAN ports on OSMs support VLAN translation. LAN ports on OSMs are in a single port group.

Product Number	Number of Ports	Number of Port Groups	Port Ranges per Port Group	Translations per Port Group	VLAN Translation Trunk-Type Support
WS-SUP32-10GE	3	2	1, 2–3	16	ISL 802.1Q
WS-SUP32-GE	9	1	1–9	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

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-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 8-8.

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.
	`Router(config-if)# no switchport vlan mapping {all | original_vlan_ID translated_vlan_ID}`	Deletes the 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* = **ethernet**, **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.
	`Router(config-if)# no switchport vlan mapping enable`	Disables 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* = **ethernet**, **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 on the Catalyst 6500 series switch.
- 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 Catalyst 6500 series switch. Make sure 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.
	`Router(config)# no vlan mapping dot1q {all | dot1q_vlan_ID}`	Deletes the mapping.
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.

CHAPTER 13

Configuring Private VLANs

This chapter describes how to configure private VLANs on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding How Private VLANs Work

These sections describe how private VLANs work:

Private VLAN Domains

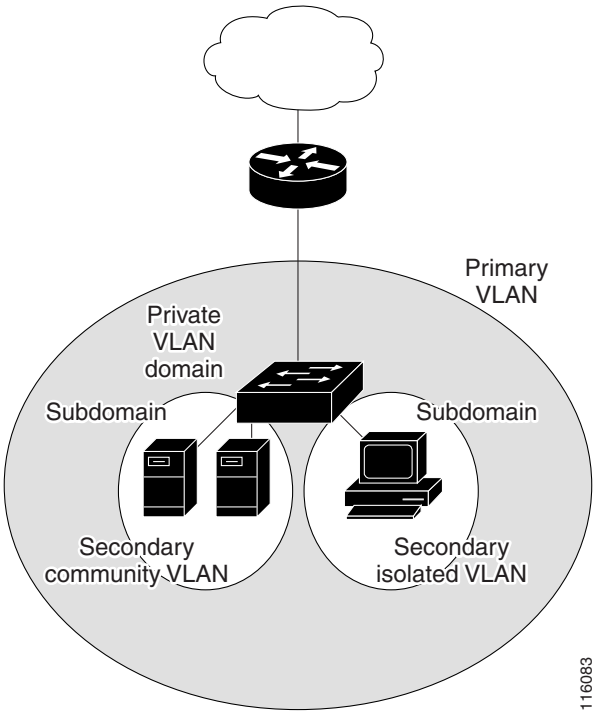
The private VLAN feature addresses two problems that service providers encounter when using VLANs:

- 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 13-1).

Figure 13-1 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:

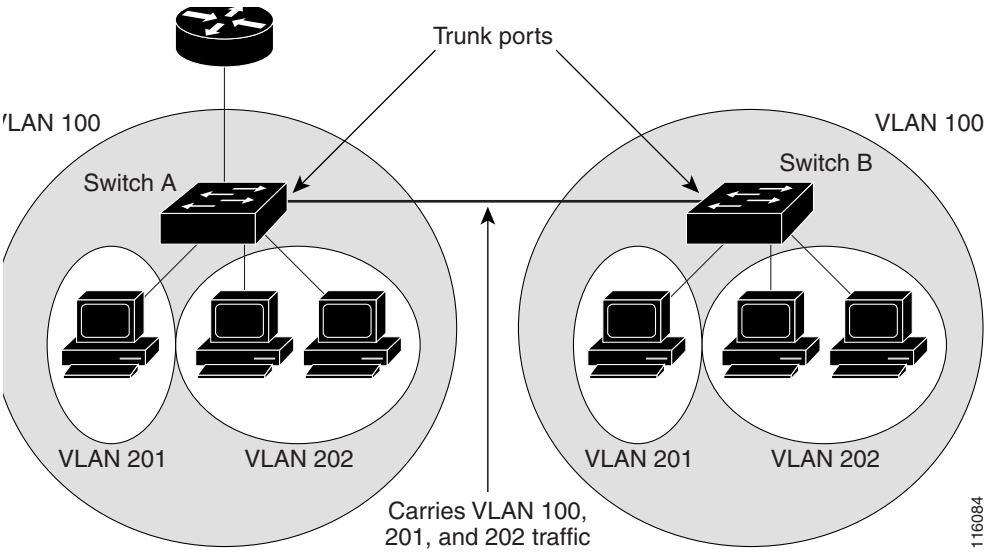
- Assigning a block of addresses to a customer VLAN can result in unused IP addresses.
- If the number of devices in the VLAN increases, the number of assigned addresses might not be large enough to accommodate them.

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 13-2.)

Figure 13-2 Private VLANs Across Switches



'LAN 100 = Primary VLAN
'LAN 201 = Secondary isolated VLAN
'LAN 202 = Secondary community VLAN

Because VTP does 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.

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 13-6
- Private VLANs and SVIs, page 13-6

See also the "Private VLAN Configuration Guidelines and Restrictions" section on page 13-6.

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 13-7
- Private VLAN Port Configuration, page 13-9
- Limitations with Other Features, page 13-9

Secondary and Primary VLAN Configuration

When configuring private VLANs consider these guidelines:

- After you configure a private VLAN and set VTP to transport mode, you are not allowed to change the VTP mode to client or server. For information about VTP, see Chapter 11, "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 "VLAN Configuration Options" section on page 12-9.
- 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.
- VTP does not propagate a private VLAN configuration. You must configure private VLANs on each device where you want private VLAN ports.
- You cannot configure VLAN 1 or VLANs 1002 to 1005 as primary or secondary VLANs. Extended VLANs (VLAN IDs 1006 to 4094) can 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 32, "Configuring 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 38, "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 33-25.

- 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 33-25. 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 32-8.) 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 30, "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 48, "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 18, "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.

- 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.
- You can configure port security on ports that are in a private VLAN.
- Do not configure a remote SPAN (RSPAN) VLAN as a private VLAN primary or secondary VLAN. For more information about SPAN, see Chapter 48, "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.
- When protocol filtering is enabled on a Supervisor Engine 1, all the required Local Target Logic (LTL) buckets of a private VLAN port should be programmed with the appropriate secondary VLAN indexes.
- 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.
- Here are some restrictions for configuring 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. (CSCea67876).

  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.

- Here are some restrictions for configuring 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:

- Configuring a VLAN as a Private VLAN, page 13-11
- Associating Secondary VLANs with a Primary VLAN, page 13-12
- Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN, page 13-13
- Configuring a Layer 2 Interface as a Private VLAN Host Port, page 13-14
- Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port, page 13-15

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}` `Router(config-vlan)# no private-vlan {community | isolated | primary}`	Configures a VLAN as a private VLAN. Clears the private VLAN configuration. **Note** These commands do not take effect until you exit VLAN configuration submode.
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.
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.
	`Router(config-if)# no switchport mode private-vlan`	Clears private VLAN port configuration.

	Command	Purpose
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.
	`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* = **ethernet**, **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 type[1] 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.

	Command	Purpose
Step 3	`Router(config-if)# switchport mode private-vlan {host | promiscuous}`	Configures the Layer 2 port as a private VLAN promiscuous port.
	`Router(config-if)# no switchport mode private-vlan`	Clears the private VLAN port configuration.
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.
	`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 [type[1] slot/port] switchport`	Verifies the configuration.

1. *type* = **ethernet**, **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.
- 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 13-1 shows the privileged EXEC commands for monitoring private VLAN activity.

Table 13-1 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          Fa2/1, Gi3/1, Gi3/2
10      502       community         Fa2/11, Gi3/1, Gi3/4
10      503       non-operational
```

CHAPTER 14

Configuring Cisco IP Phone Support

This chapter describes how to configure support for Cisco IP phones on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding Cisco IP Phone Support

These sections describe Cisco IP phone support:

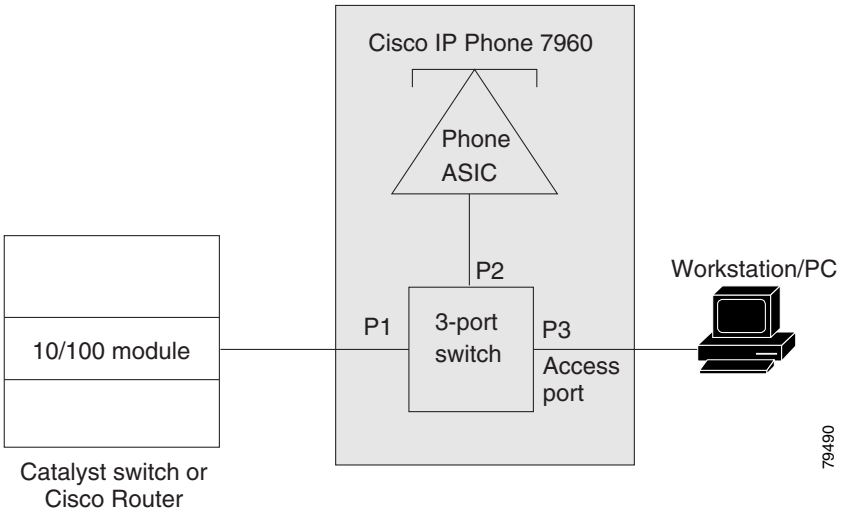
Cisco IP Phone Connections

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 14-1 shows a Cisco IP phone connected between a switch and a PC.

Figure 14-1 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. To provide more predictable voice traffic flow, you can configure QoS to trust the Layer 3 IP precedence or Layer 2 CoS value in the voice traffic (refer to Chapter 38, "Configuring PFC QoS").

Note 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. Configure QoS policies that use the Layer 3 IP precedence value on other switching modules.

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.

You can configure Layer 2 access ports on the switch to send Cisco Discovery Protocol (CDP) packets that instruct 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).

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.

Cisco IP Phone Data Traffic

Note Untagged 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 14-1), 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 enter one 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.

Cisco IP Phone Power Configurations

These sections describe Cisco IP phone power configurations:

- Locally Powered Cisco IP Phones, page 14-3
- Inline-Powered Cisco IP Phones, page 14-3

Locally Powered Cisco IP Phones

There are two varieties of local power:

- From a power supply connected to the Cisco IP phone
- From a power supply through a patch panel over the twisted-pair Ethernet cable to the Cisco IP phone

When a locally powered Cisco IP phone is present on a switching module port, the switching module cannot detect its presence. The supervisor engine discovers the Cisco IP phone through CDP messaging with the Cisco IP phone.

If a locally powered Cisco IP phone loses local power and the mode is set to **auto**, the switching module discovers the Cisco IP phone and informs the supervisor engine, which then supplies inline power to the Cisco IP phone.

Inline-Powered Cisco IP Phones

Switching modules that support an inline power daughtercard can supply power over the twisted-pair Ethernet cable to external devices such as IP phones, IP cameras, and wireless access points. Cisco inline power modules are available to support one or both of the two most common implementations of Power over Ethernet (PoE):

- Cisco prestandard inline power
- IEEE 802.3af standard

With an inline power card installed, a switching module can automatically detect and provision a powered device that adheres to a PoE implementation supported by the card. The switching module can supply power to devices supporting other PoE implementations only through manual configuration.

Only one device can be powered per port, and the device must be connected directly to the switch port. For example, if a second IP phone is daisy-chained off a phone that is connected to the switch port, the second phone cannot be powered by the switch.

Note For information about switching modules that support inline power, refer to the *Release Notes for Cisco IOS Release 12.2ZY on the Supervisor Engine 32 PISA* publication at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/release/notes/ol_13011.html

Inline Power Management

Each inline-powered device, such as an IP phone or a wireless access point, requires power to be allocated from the chassis power budget. Because each powered device 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.

With Release 12.2ZYA and later releases, you can configure the switching module to allocate and apply inline power to individual ports in these situations:

- When an attached inline-powered device is discovered, at a power level based on information sensed from the device or at a default or specified maximum power level (**auto** mode)
- At a fixed default or specified level, whether or not an inline-powered device is present on the port (**static** mode)

The Cisco prestandard PoE implementation defines a method to sense an attached inline-powered device and to apply an initial power level. After activation, a Cisco prestandard device that supports CDP can negotiate a lower or higher power allocation using CDP messaging.

The IEEE 802.3af PoE standard defines a method to sense an attached device and to immediately classify the device's power requirement into three power ranges:

- Class 1: Up to 4 W per port
- Class 2: Up to 7 W per port
- Class 3: Up to 15.4 W per port

The IEEE standard contains no provision for subsequent readjustment of a device's power allocation. Cisco inline-powered devices that support IEEE 802.3af and CDP can use CDP to override the IEEE 802.3af power classification.

A switching module whose inline power card supports both PoE implementations will attempt both detection methods in parallel. If the attached device responds to both detection methods, the module will consider the device to be an IEEE 802.3af device.

Caution When an IP phone 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.

Example: Cisco Prestandard IP Phone

When a switching module port detects an unpowered Cisco prestandard IP phone, the switching module reports to the supervisor engine that an unpowered Cisco IP phone is present and indicates which module and port the phone is on. If the port is configured in **auto** mode, the supervisor engine determines whether there is enough system power available to power up the Cisco IP phone. The power allocation will be the lower value of the default power or the configured port maximum power if a maximum has been specified. If there is sufficient power available, the supervisor engine removes the allocated power from the total available system power and sends a message to the switching module instructing it to provide power to the port. If there is not enough available power for the Cisco IP phone, the supervisor engine sends a message to the switching module indicating that power is denied to the port.

Cisco IP phones may have different power requirements. Unless a lower maximum power level has been configured for the port, the supervisor engine initially allocates the configured default of 7 W (167 mA at 42 V) to the Cisco IP phone. When the correct amount of power is determined from the CDP messaging with the Cisco IP phone, the supervisor engine reduces or increases the allocated power.

For example, the default allocated power is 7 W. A Cisco IP phone requiring 6.3 W is plugged into a port. The supervisor engine allocates 7 W for the Cisco IP phone and powers it up. Once the Cisco IP phone is operational, it sends a CDP message with the actual power requirement to the supervisor engine. The supervisor engine then decreases the allocated power to the required amount.

When you power off the Cisco IP phone through the CLI or SNMP or remove it, the supervisor engine sends a message to the switching module to turn off the power on the port. That power is then returned to the available system power.

Example: IEEE 802.3af IP Phone

When a switching module port detects an unpowered IEEE 802.3af-compliant IP phone, the module detects the phone's IEEE 802.3af power classification and notifies the supervisor engine of the phone's location and power requirement. If there is sufficient system power available, the supervisor engine allocates the power level indicated by the IEEE class and sends a message to the switching module approving power to the port. If there is not enough available power for the IP phone, the supervisor engine sends a message to the switching module indicating that power is denied to the port.

For example, an IEEE 802.3af-compliant IP phone consuming 7.1 W is plugged into a port. The switching module detects the phone and determines that its IEEE power classification is Class 3, which requires between 7.0 W and 15.4 W. The switching module notifies the supervisor engine of the port location and the IEEE classification of the phone. If there is sufficient power available, the supervisor engine removes 15.4 W from the system power and approves power to the port. For this phone, the system is required to reserve an unnecessary 8.3 W due to the broad ranges of the IEEE classification system. If the **auto** mode is selected for this port with a maximum power level lower than 15.4 W, the Class 3 phone will be denied power. Cisco inline-powered devices that support IEEE 802.3af and CDP can use CDP to override the IEEE 802.3af power classification. In this case, the actual power requirement will be negotiated through CDP to a lower value.

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 Catalyst 6500 series switch 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 (refer to Chapter 43, "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:

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 type[1] slot/port`	Selects an interface to configure.
Step 2	`Router(config-if)# switchport voice vlan {voice_vlan_ID | dot1p | none | untagged}`	Configures the way in which the Cisco IP phone transmits voice traffic.
	`Router(config-if)# no switchport voice vlan`	Clears the configuration.
Step 3	`Router(config)# end`	Exits configuration mode.
Step 4	`Router# show interface type slot/port switchport` `Router# show running-config interface type slot/port`	Verifies the configuration.

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

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).
- Refer to Chapter 38, "Configuring PFC QoS," for information about how to configure QoS.
- Refer to the "Configuring a LAN Interface as a Layer 2 Access Port" section on page 8-14 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

To configure the way in which the Cisco IP phone transmits data traffic, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects an interface to configure.
Step 2	`Router(config-if)# mls qos trust extend [cos cos_value]`	Configures the way in which the Cisco IP phone transmits data traffic.
	`Router(config-if)# no mls qos trust extend`	Clears the configuration.
Step 3	`Router(config)# end`	Exits configuration mode.
Step 4	`Router# show interface type slot/port switchport` `Router# show running-config interface type slot/port`	Verifies the configuration.

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

When configuring the way in which the Cisco IP phone transmits data traffic, note the following information:

- To send CDP packets that configure the 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 the 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
```

Configuring Inline Power Support

To configure inline power support, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects an interface to configure.
Step 2	`Router(config-if)# power inline {auto | static | never} [max milliwatts]`	Configures inline power support.
	`Router(config-if)# no power inline`	Clears the configuration.
Step 3	`Router(config)# end`	Exits configuration mode.
Step 4	`Router# show power inline {type slot/port | module slot}[detail]`	Verifies the configuration.

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

When configuring inline power support, note the following information:

- To configure auto-detection of an inline-powered device and auto-allocation of port inline power, enter the **auto** keyword.
- To configure auto-detection of an inline-powered device but reserve a fixed inline power 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, an inline-powered device that supports CDP can negotiate a different power level.
- To disable auto-detection of an inline-powered device, enter the **never** keyword.
- The following information applies to WS-F6K-48-AF and WS-F6K-GE48-AF inline power cards:
  - In Cisco IOS Release 12.2ZYA 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 defined.

  - In Cisco IOS Release 12.2ZYA 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 Fast Ethernet port 5/1:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
```

```
Router(config-if)# power inline never
```

This example shows how to enable inline power on Fast Ethernet port 5/1:

```
Router# configure terminal
Router(config)# interface fastethernet 5/1
Router(config-if)# power inline auto
```

This example shows how to verify the inline power configuration on Fast Ethernet port 5/1:

```
Router# show power inline fastethernet 5/1
Interface  Admin    Oper     Power        Device
                            (Watts)
---------- ----- ---------- ------- -------------------
Fa5/1      auto  on           6.3   cisco phone device
```

CHAPTER 15

Configuring IEEE 802.1Q Tunneling

This chapter describes how to configure IEEE 802.1Q tunneling on the Catalyst 6500 series switches.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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.

This chapter consists of these sections:

Understanding How 802.1Q Tunneling Works

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 15-1 on page 15-2 and Figure 15-2 on page 15-3.

Figure 15-1 IEEE 802.1Q Tunnel Ports in a Service-Provider Network

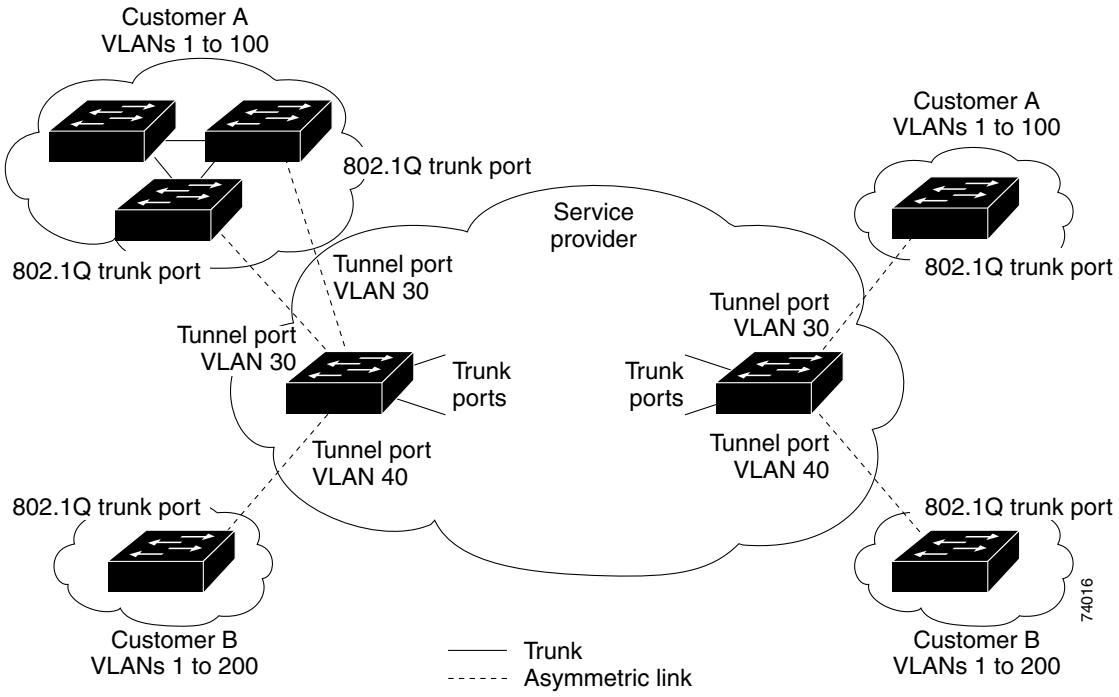
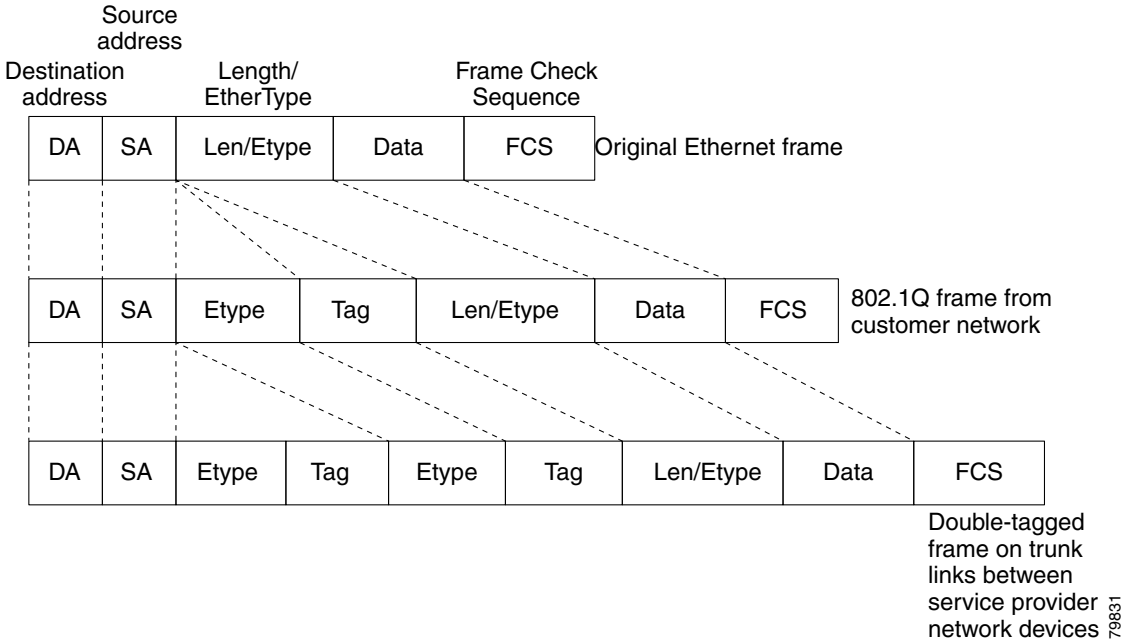


Figure 15-2 Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames



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 7-10.
  - 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 16, "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 15-6
- Configuring the Switch to Tag Native VLAN Traffic, page 15-6

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` `Router(config-if)# no switchport mode dot1q-tunnel`	Configures the Layer 2 port as a tunnel port. Clears the tunnel port configuration.
Step 4	`Router(config-if)# end`	Exits configuration mode.
Step 5	`Router# show dot1q-tunnel [{interface type interface-number}]`	Verifies the configuration.

1. *type* = **ethernet**, **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)# 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.

To configure the switch to tag traffic in the native VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# vlan dot1q tag native`	Configures the switch to tag native VLAN traffic.
	`Router(config)# no vlan dot1q tag native`	Clears the configuration.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show vlan dot1q tag native`	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
```

CHAPTER 16

Configuring Layer 2 Protocol Tunneling

This chapter describes how to configure Layer 2 protocol tunneling on the Catalyst 6500 series switches.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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.

This chapter consists of these sections:

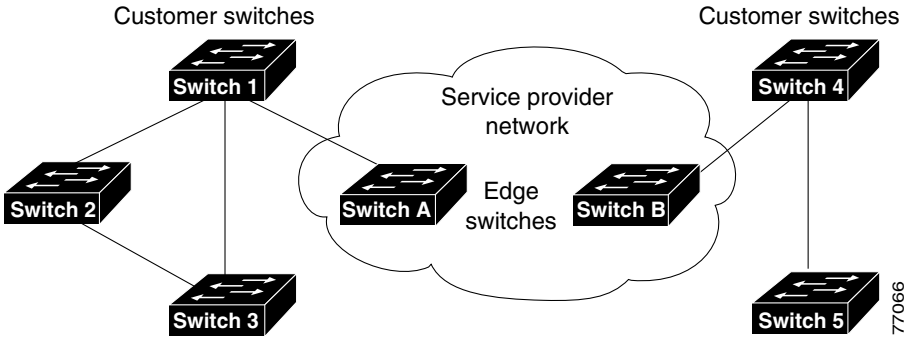
Understanding How Layer 2 Protocol Tunneling Works

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 16-1).
- 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 16-1) 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 16-1 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 disabled 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 7-10.
  - 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|drop-threshold [packets|shutdown-threshold [packets]|stp|vtp]` `Router(config-if)# no l2protocol-tunnel [cdp| drop-threshold|shutdown-threshold|stp|vtp]`	Configures the Layer 2 port as a Layer 2 protocol tunnel port for the protocols specified. Clears the configuration.
Step 4	`Router(config)# end`	Exits configuration mode.
Step 5	`Router# show l2protocol-tunnel [interface type[1] slot/port|summary]`	Verifies the configuration.

1. *type* = **ethernet**, **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 1-second period. If a shutdown threshold is not specified, the value is 0 (shutdown 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).

Note Refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY for more information about the **l2ptguard** keyword for the following commands:

- **errdisable detect cause**
- **errdisable recovery cause**

This example shows how to configure Layer 2 protocol tunneling and shutdown 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 10
Router(config-if)# l2protocol-tunnel shutdown-threshold stp 10
Router(config-if)# l2protocol-tunnel shutdown-threshold vtp 10
Router(config-if)# end
Router# show l2protocol-tunnel summary
```

```
Port   Protocol        Threshold
                       (cos/cdp/stp/vtp)
----------------------------------------
Fa5/1   cdp stp vtp     0/10  /10  /10          down trunk
Router#
```

This example shows how to display counter information for port 5/1:

```
Router# show l2protocol-tunnel interface fastethernet 5/1
Port   Protocol        Threshold           Counters
                       (cos/cdp/stp/vtp)   (cdp/stp/vtp/decap)
---------------------------------------------------------------
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 10
Router(config-if)# no l2protocol-tunnel shutdown-threshold stp 10
Router(config-if)# no l2protocol-tunnel shutdown-threshold vtp 10
Router(config-if)# no l2protocol-tunnel cdp
Router(config-if)# no l2protocol-tunnel stp
Router(config-if)# no l2protocol-tunnel vtp
Router(config-if)# end
Router# show l2protocol-tunnel summary
Port   Protocol        Threshold
                       (cos/cdp/stp/vtp)
----------------------------------------
Router#
```

This example shows how to clear Layer 2 protocol tunneling port counters:

```
Router# clear l2protocol-tunnel counters
Router#
```

CHAPTER 17

Configuring STP and MST

This chapter describes how to configure the Spanning Tree Protocol (STP) and Multiple Spanning Tree (MST) protocol on Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Note For information on configuring the PortFast, UplinkFast, and BackboneFast STP enhancements, see Chapter 18, "Configuring Optional STP Features."

Understanding How STP Works

These sections describe how STP works:

STP Overview

STP 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.

Catalyst 6500 series switches use STP (the IEEE 802.1D bridge protocol) 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

Note In Catalyst 6500 series switches, the extended system ID is always enabled.

The bridge priority is a 4-bit value when the extended system ID is enabled (see Table 17-1 on page 17-3 and the "Configuring the Bridge Priority of a VLAN" section on page 17-33).

Extended System ID

A 12-bit extended system ID field is part of the bridge ID (see Table 17-1 on page 17-3). Catalyst 6500 series switches have 64 MAC addresses and always use the 12-bit extended system ID.

Table 17-1 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 switches have 64 addresses available to support software features such as STP. To view the MAC address range, enter the **show catalyst6000 chassis-mac-address** command.

STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

With MAC address reduction enabled on any device, you should also enable MAC address reduction on all other Layer 2 connected network devices to avoid undesirable root bridge election and spanning tree topology issues.

When MAC address reduction is enabled, the root bridge priority becomes a multiple of 4096 plus the VLAN ID. With MAC address reduction 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 run the MAC address reduction feature, 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 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 value increases the probability; a lower 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 17-2 describes the STP protocol timers that affect STP performance.

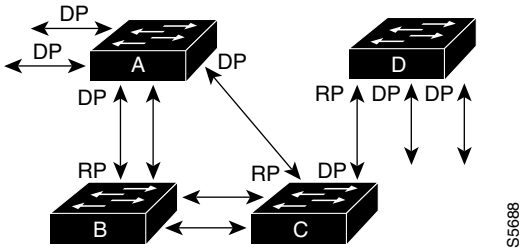
Table 17-2 STP Protocol Timers

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 17-1, 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 17-1 Spanning Tree Topology



RP = Root Port
DP = Designated Port

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, page 17-5
- Blocking State, page 17-7
- Listening State, page 17-7
- Learning State, page 17-8
- Forwarding State, page 17-9
- Disabled State, page 17-10

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 on a Catalyst 6500 series switch 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.

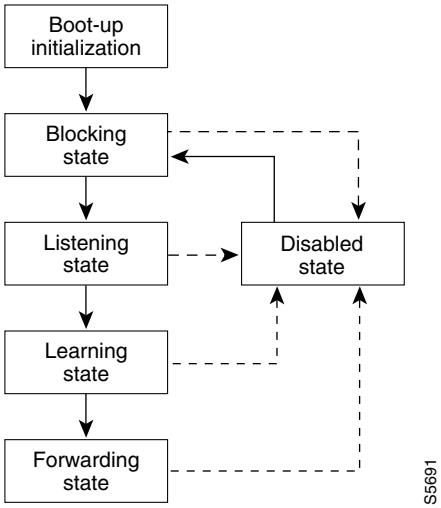
- Forwarding—The Layer 2 LAN port forwards frames.
- Disabled—The Layer 2 LAN port does not participate in STP and is not forwarding frames.

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 17-2 illustrates how a Layer 2 LAN port moves through the five states.

Figure 17-2 STP Layer 2 LAN Interface States



When you enable STP, every port in the Catalyst 6500 series switch, 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.

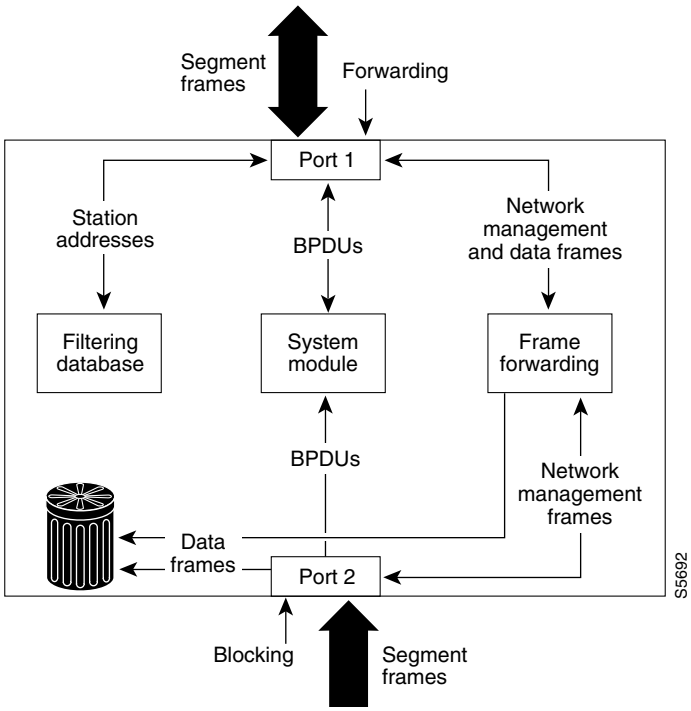
When the STP algorithm places a Layer 2 LAN port in the forwarding state, the following process occurs:

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 17-3. 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 17-3 Interface 2 in Blocking State



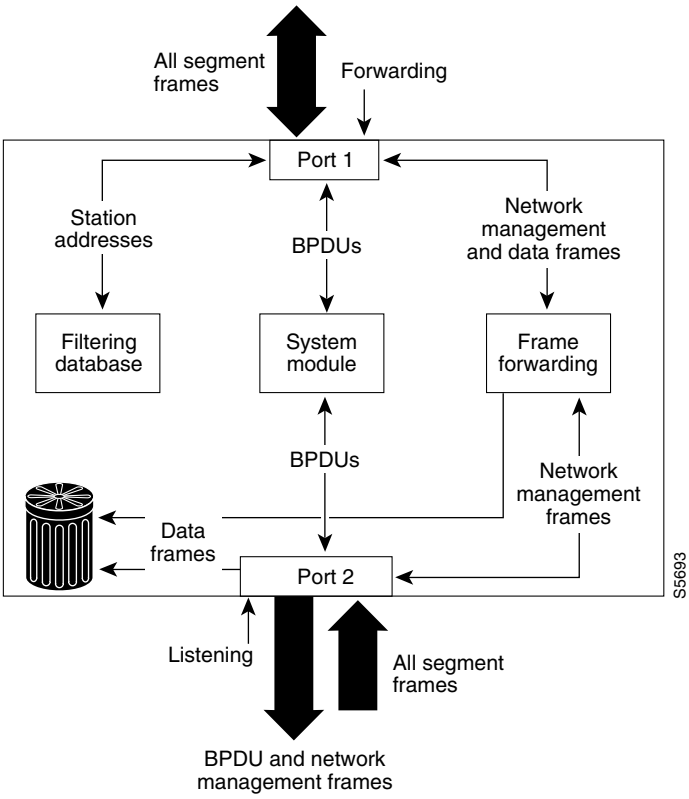
A Layer 2 LAN port in the blocking state performs as follows:

- 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 17-4 shows a Layer 2 LAN port in the listening state.

Figure 17-4 Interface 2 in Listening State



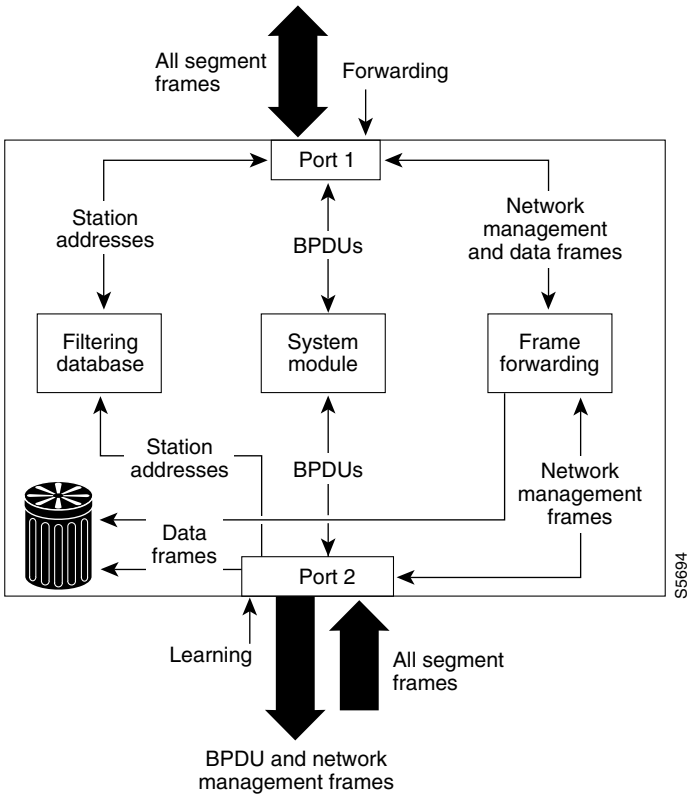
A Layer 2 LAN port in the listening state performs as follows:

- 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 17-5 shows a Layer 2 LAN port in the learning state.

Figure 17-5 Interface 2 in Learning State



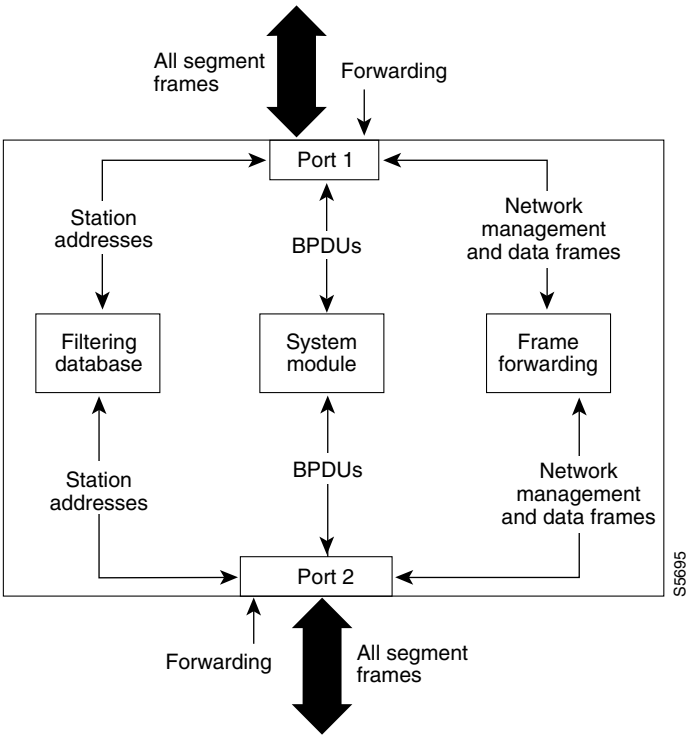
A Layer 2 LAN port in the learning state performs as follows:

- 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 17-6. The Layer 2 LAN port enters the forwarding state from the learning state.

Figure 17-6 Interface 2 in Forwarding State



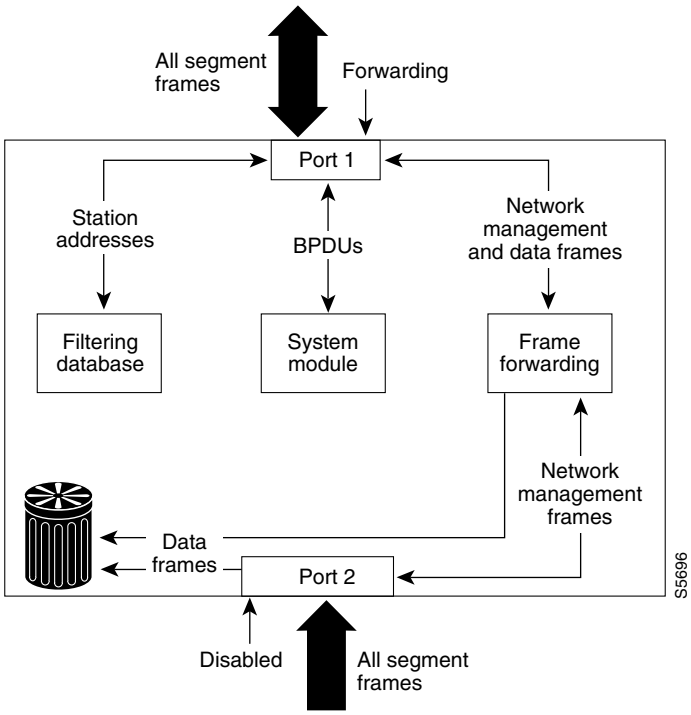
A Layer 2 LAN port in the forwarding state performs as follows:

- 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 17-7. A Layer 2 LAN port in the disabled state is virtually nonoperational.

Figure 17-7 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.

For more information on 802.1Q trunks, see Chapter 8, “Configuring LAN Ports for Layer 2 Switching.”

Understanding How IEEE 802.1w RSTP Works

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 17-4. 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 17-3 provides a comparison of 802.1D and RSTP port states.

Table 17-3 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 17-3 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 17-8, 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.

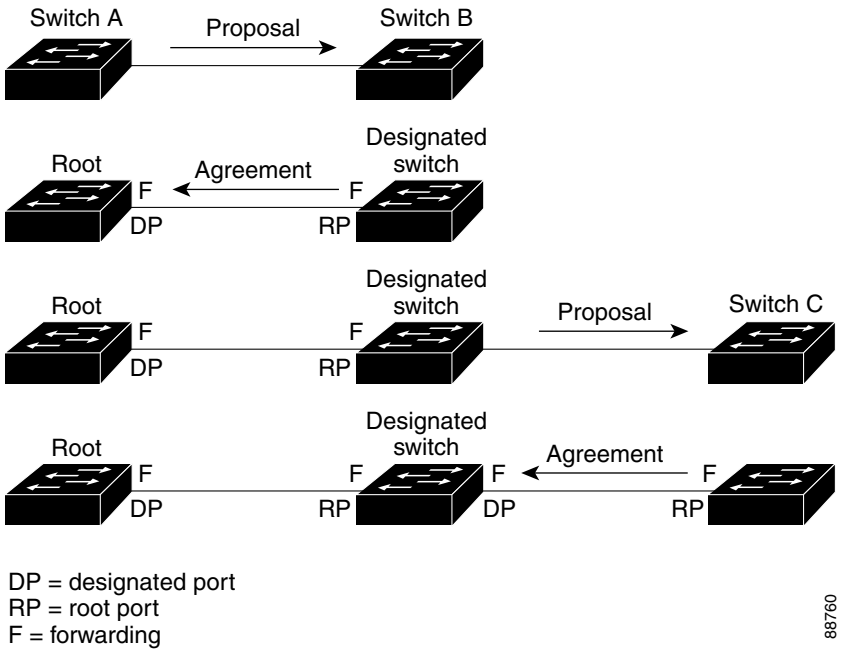
  After receiving the proposal message, switch B selects as its new root port the port from which the proposal message was received, forces all nonedge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

  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 17-8 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.

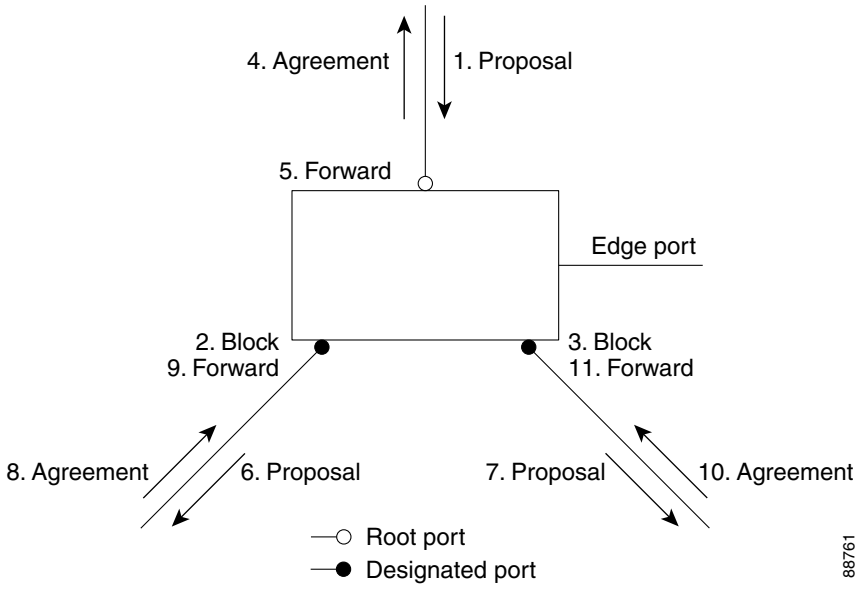
The switch is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the switch is synchronized if:

- 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 17-9.

Figure 17-9 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 17-15
- Processing Superior BPDU Information, page 17-16
- Processing Inferior BPDU Information, page 17-16

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 17-4 describes the RSTP flag fields.

Table 17-4 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
5	Forwarding

Table 17-4 RSTP BPDU Flags (continued)

Bit	Function
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.

Understanding MST

These sections describe MST:

- MST Overview, page 17-18
- MST Regions, page 17-18

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 Chapter 17, "Configuring STP and MST." For information about other spanning tree features such as Port Fast, UplinkFast, root guard, and so forth, see Chapter 18, "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 17-10 on page 17-21.

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 17-20 and the "Spanning Tree Operations Between MST Regions" section on page 17-20.

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 17-10 on page 17-21. 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.

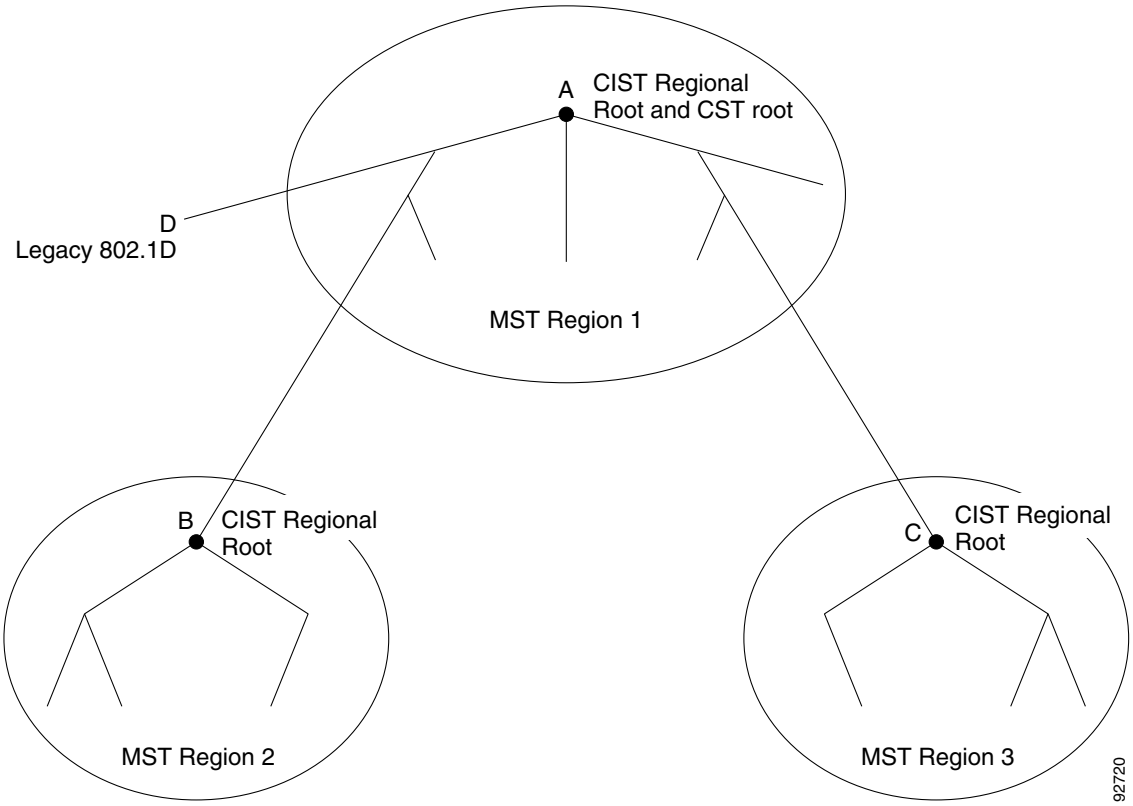
Spanning Tree Operations Between MST Regions

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 17-10 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 17-10 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 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 17-5 compares the IEEE standard and the Cisco prestandard terminology.

Table 17-5 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 17-23
- Spanning Tree Interoperation Between Legacy and Standard-Compliant Switches, page 17-24
- Detecting Unidirectional Link Failure, page 17-24

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:

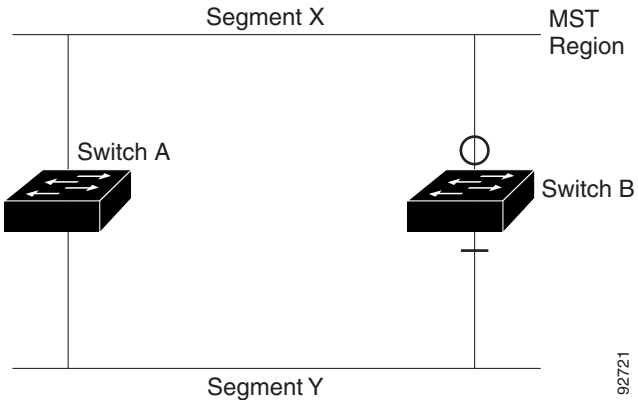
- The boundary port is the root port of the CIST regional root—When the CIST instance port is proposed and is synchronized, it can send back an agreement and move to the forwarding state only after all the corresponding MSTI ports are synchronized (and thus forwarding). The MSTI ports now have a special *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 17-11 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 17-11 Standard-Compliant and Prestandard Switch Interoperation



Note We recommend that you minimize the interaction between standard and prestandard MST implementations.

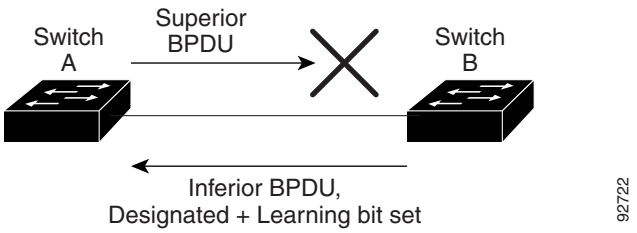
Detecting Unidirectional Link Failure

This feature is not yet present in the IEEE MST standard, but it is included in the standard-compliant implemtation. The software 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 state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

Figure 17-12 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 17-12 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:

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 17-6 shows the default STP configuration.

Table 17-6 STP Default Configuration

Feature	Default Value
Enable state	STP enabled for all VLANs
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)	• 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)	• Gigabit Ethernet: 4 • Fast Ethernet: 19 • Ethernet: 100
Hello time	2 seconds
Forward delay time	15 seconds
Maximum aging time	20 seconds
Mode	PVST

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 Catalyst 6500 series 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 17-6 on page 17-26).
	`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

The extended system ID is enabled permanently on Catalyst 6500 series switches. This example shows how to verify the configuration:

```
Router# show spanning-tree summary | include Extended
Extended system ID is enabled.
```

Configuring the Root Bridge

Catalyst 6500 series switches 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 17-1 on page 17-3.)

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 Catalyst 6500 series 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 Catalyst 6500 series switch as the root bridge.

To configure a Catalyst 6500 series 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 a Catalyst 6500 series switch as the root bridge. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 17-6 on page 17-26).
	`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 Catalyst 6500 series 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 Catalyst 6500 series 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 a Catalyst 6500 series 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 a Catalyst 6500 series switch as the secondary root bridge. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 12-1 on page 12-2).
	`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 Catalyst 6500 series 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 12-1 on page 12-2).
	`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.

	Command	Purpose
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 12-1 on page 12-2).
	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 12-1 on page 12-2).
	Router(config)# **no spanning-tree vlan** *vlan_ID* **priority**	Reverts to the default bridge priority value.

	Command	Purpose
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 12-1 on page 12-2).
	`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 12-1 on page 12-2).
	`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 12-1 on page 12-2).
	`Router(config)# no spanning-tree vlan vlan_ID max-age`	Reverts to the default maximum aging time.

	Command	Purpose
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 17-25.

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 17-7 shows the default MST configuration.

Table 17-7 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-CIST port basis)	128
spanning tree port cost (configurable on a per-CIST port basis)	1000 Mbps: 4 100 Mbps: 19 10 Mbps: 100
Hello time	2 seconds
Forward-delay time	15 seconds

Table 17-7 Default MST Configuration (continued)

Feature	Default Setting
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.

	Command	Purpose
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.
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 17-1 on page 17-3.)

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 17-40.
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.

	Command	Purpose
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.

	Command	Purpose
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	`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 17-13.

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 17-8.

Table 17-8 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.

CHAPTER 18

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, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Note For information on configuring the spanning tree protocol (STP), see Chapter 17, "Configuring STP and MST."

Understanding How PortFast Works

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.

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 ports. If you enable PortFast on a port connected to a switch, you might create a temporary bridging loop.

Understanding How BPDU Guard Works

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 state. In a valid configuration, PortFast Layer 2 LAN interfaces do not receive BPDUs. 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 state.

Understanding How PortFast BPDU Filtering Works

PortFast BPDU filtering allows the administrator to prevent the system from sending or even receiving BPDUs on specified ports.

When configured globally, PortFast BPDU filtering applies to all operational PortFast ports. Ports in an operational PortFast state are supposed to be connected to hosts, that typically drop BPDUs. If an operational PortFast port receives a BPDU, it immediately loses its operational PortFast status. 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 Explicate 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 go to forwarding.

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 18-10), 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 18-1 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 18-1 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.

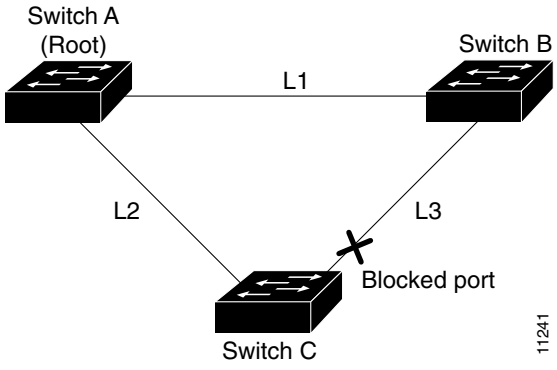
Understanding How UplinkFast Works

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.

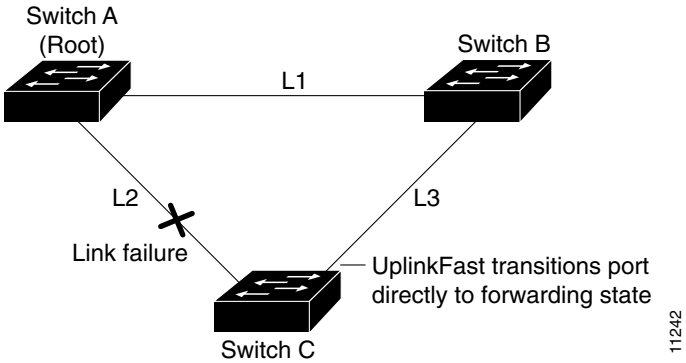
Figure 18-1 shows an example topology with no link failures. Switch A, the root bridge, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 18-1 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 18-2. This switchover takes approximately one to five seconds.

Figure 18-2 UplinkFast Example After Direct Link Failure



Understanding How BackboneFast Works

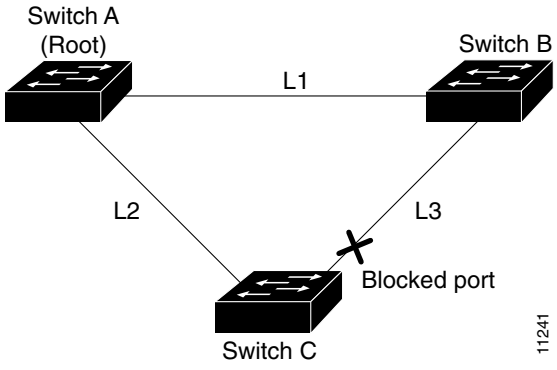
BackboneFast is initiated when a root port or blocked port on a network device receives inferior BPDUs from its designated bridge. An inferior BPDU identifies one network device as both the root bridge and the designated bridge. When a network device receives an inferior BPDU, it indicates that a link 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.

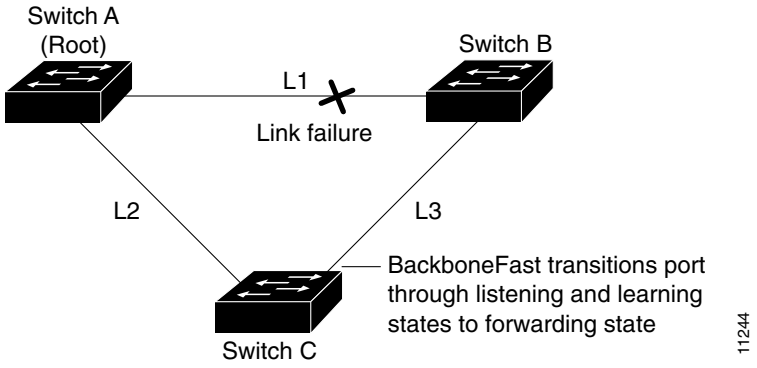
Figure 18-3 shows an example topology with no link failures. Switch A, the root bridge, connects directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that connects directly to Switch B is in the blocking state.

Figure 18-3 BackboneFast Example Before Indirect Link Failure



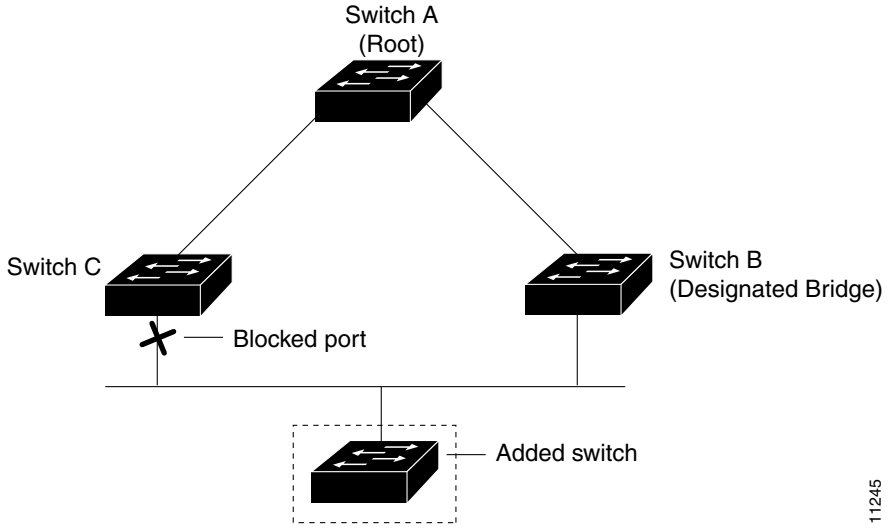
If link L1 fails, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root bridge over L1, it detects the failure and elects itself the root and begins sending BPDUs to Switch C indicating itself as the root. When 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 18-4 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 18-4 BackboneFast Example After Indirect Link Failure



If a new network device is introduced into a shared-medium topology as shown in Figure 18-5, 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 18-5 Adding a Network Device in a Shared-Medium Topology



Understanding How EtherChannel Guard Works

EtherChannel guard detects a misconfigured EtherChannel where interfaces on the Catalyst 6500 series switch are configured as an EtherChannel while interfaces on the other device are not or 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 Catalyst 6500 series switch into the errdisabled state.

Understanding How Root Guard Works

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 How Loop Guard Works

Loop guard helps prevent bridging loops that could occur because of a uni-directional 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 18-6 shows loop guard in a triangle switch configuration.

Figure 18-6 Triangle Switch Configuration with Loop Guard

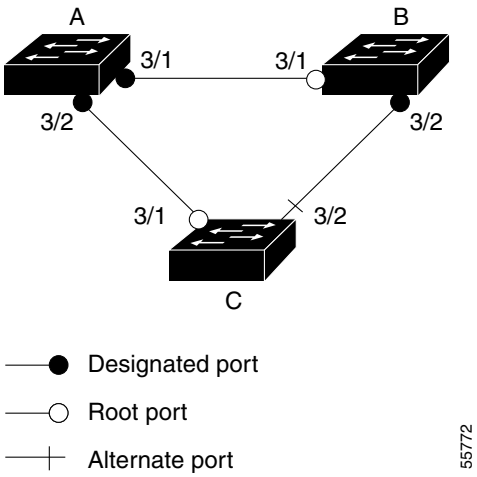


Figure 18-6 illustrates the following configuration:

- 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.

Enabling PortFast

Caution Use PortFast *only* when connecting a single end station to a Layer 2 access port. Otherwise, you might create a network loop.

To enable PortFast on a Layer 2 access 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 portfast**	Enables PortFast on a Layer 2 access port connected to a single workstation or server.
Step 3	Router(config-if)# **spanning-tree portfast default**	Enables PortFast.
Step 4	Router(config-if)# **end**	Exits configuration mode.
Step 5	Router# **show running interface** {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}	Verifies the configuration.

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

This example shows how to enable PortFast on Fast Ethernet interface 5/8:

```
Router# configure terminal
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree portfast
Router(config-if)# end
Router#
```

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 access vlan 200
 switchport mode access
 spanning-tree portfast
end

Router#
```

To enable the default PortFast configuration, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree portfast default`	Configures the PortFast default.
Step 2	`Router(config)# show spanning-tree summary totals`	Verifies the global configuration.
Step 3	`Router(config)# show spanning-tree interface x detail`	Verifies the effect on a specific port.
Step 4	`Router(config-if)# spanning-tree portfast trunk`	Enables the PortFast trunk on a port
Step 5	`Router# show spanning-tree interface fastEthernet x detail`	Verifies the configuration.

This example shows how to enable the default PortFast configuration:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# spanning-tree portfast default
Router(config)# ^Z

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#

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

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#
```

Enabling PortFast BPDU Filtering

These sections describe how to configure PortFast BPDU filtering.

To enable PortFast BPDU filtering globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree portfast bpdufilter default`	Enables BPDU filtering globally on the switch.
Step 2	`Router# show spanning-tree summary totals`	Verifies the configuration.

BPDU filtering is set to default on each port. This example shows how to enable PortFast BPDU filtering on the port and verify the configuration in PVST+ mode:

Note For PVST+ information, see Chapter 17, "Configuring STP and MST."

```
Router(config)# spanning-tree portfast bpdufilter default
Router(config)# ^Z

Router# show spanning-tree summary totals
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 enabled 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#
```

To enable PortFast BPDU filtering on a nontrunking port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface fastEthernet 4/4`	Selects the interface to configure.
Step 2	`Router(config-if)# spanning-tree bpdufilter enable`	Enables BPDU filtering.
Step 3	`Router# show spanning-tree interface fastEthernet 4/4`	Verifies the configuration.

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
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

To enable BPDU Guard globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree portfast bpduguard default` `Router(config)# no spanning-tree portfast bpduguard default`	Enables BPDU Guard globally. Disables BPDU Guard globally.
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 bpduguard
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary totals default
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 enabled 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#
```

Enabling UplinkFast

UplinkFast increases the bridge priority to 49152 and adds 3000 to the STP port cost of all Layer 2 LAN interfaces on the Catalyst 6500 series 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 Catalyst 6500 series switch. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree uplinkfast`	Enables UplinkFast.
	`Router(config)# spanning-tree uplinkfast [max-update-rate max_update_rate]`	Enables UplinkFast with an update rate in seconds.
	`Router(config)# no spanning-tree uplinkfast max-update-rate`	Reverts to the default rate.
	`Router(config)# no spanning-tree uplinkfast`	Disables UplinkFast.
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.
	`Router(config)# no spanning-tree backbonefast`	Disables BackboneFast.
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 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.
	`Router(config)# no spanning-tree etherchannel guard misconfig`	Disables 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.
	`Router(config-if)# no spanning-tree guard root`	Disables 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* = **ethernet**, **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 4/4 detail`	Verifies the configuration impact on a port.

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 4/4 detail`	Verifies the configuration impact on that port.

1. *type* = **ethernet**, **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#
```

CHAPTER 19

Configuring Layer 3 Interfaces

This chapter contains information about how to configure Layer 3 interfaces on the Catalyst 6500 series switches.

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

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

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 12, “Configuring VLANs” and Chapter 11, “Configuring VTP.”

- Catalyst 6500 series switches do 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 PISA.
- Catalyst 6500 series switches do 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:

- These features are supported 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.

- 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* = **ethernet**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**, or **ge-wan**

Configuring IPv4 Routing and Addresses

For complete information and procedures, refer to 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:

- For information about the **maximum paths** command, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, publication.
- The Policy Feature Card 3B (PFC3B) provides 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 PFC3B provides hardware support for PBR configured on a tunnel interface.
  - The PFC3B does not provide hardware support for PBR configured with the **set ip next-hop** keywords if the next hop is a tunnel interface.
  - If the PISA address falls within the range of a PBR ACL, traffic addressed to the PISA is policy routed in hardware instead of being forwarded to the PISA. To prevent policy routing of traffic addressed to the PISA, configure PBR ACLs to deny traffic addressed to the PISA.
  - Any options in Cisco IOS ACLs that provide filtering in a PBR route-map that would cause flows to be sent to the PISA 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)

  To configure PBR, refer to 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* = **ethernet**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**, or **ge-wan**

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 PISA supports IPX with fast switching.

For complete information and procedures, refer to 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* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**, or **ge-wan**

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, refer to 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* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**, or **ge-wan**

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

Refer to 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

CHAPTER 20

Configuring UDE and UDLR

This chapter describes how to configure unidirectional Ethernet (UDE) and unidirectional link routing (UDLR) on the Catalyst 6500 series switch.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

These sections describe UDE and UDLR:

Understanding UDE and UDLR

These sections describe UDE and UDLR:

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 20-2
- Understanding Hardware-Based UDE, page 20-2
- Understanding Software-Based UDE, page 20-3

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 Refer to the "Supported Hardware" section on page 20-2 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 20-3
- Configuring UDLR, page 20-6

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 20-4
- Configuring Hardware-Based UDE, page 20-4
- Configuring Software-Based UDE, page 20-5

UDE Configuration Guidelines

When configuring UDE, follow these guidelines:

- 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 20-6
- Configuring a Receive-Only Tunnel Interface for a UDE Send-Only Port, page 20-7
- Configuring a Send-Only Tunnel Interface for a UDE Receive-Only Port, page 20-7

UDLR Back-Channel Tunnel Configuration Guidelines

When configuring UDLR back-channel tunnels, follow these guidelines:

- The PFC3B does not provide hardware support for UDLR back-channel tunnels. The PISA 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
```

CHAPTER 21

Configuring Multiprotocol Label Switching

This chapter describes how to configure Multiprotocol Label Switching (MPLS) on a Catalyst 6500 series switch.

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

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

This chapter consists of these sections:

MPLS Label Switching

These sections describe MPLS label switching:

Understanding MPLS

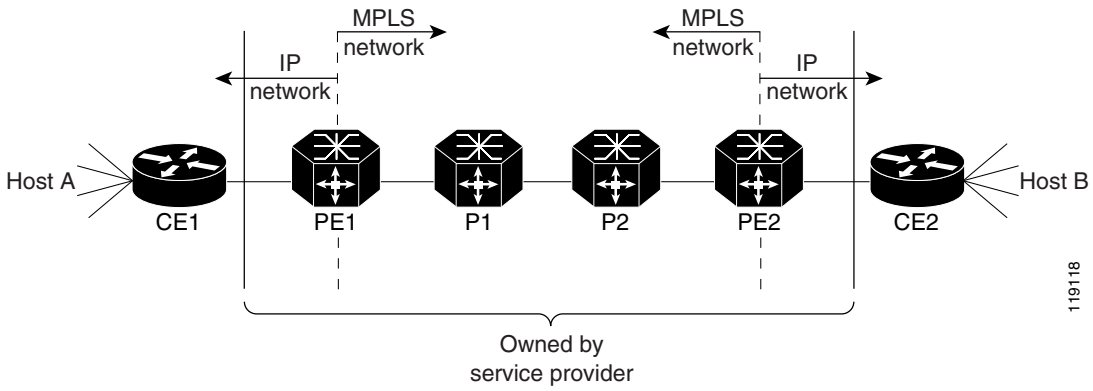
MPLS uses label switching to forward packets over various link-level technologies such as Packet-over-SONET (POS), Frame Relay, ATM, and Ethernet. Labels are assigned to packets based on groupings or forwarding equivalence classes (FECs). The label is added between the Layer 2 and the Layer 3 header.

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 21-1 shows an MPLS network of a service provider that connects two sites of a customer network.

Figure 21-1 MPLS Network



For additional information on MPLS, see this publication:

http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcftagov_ps1835_TSD_Products_Configuration_Guide_Chapter.html

Understanding MPLS Label Switching

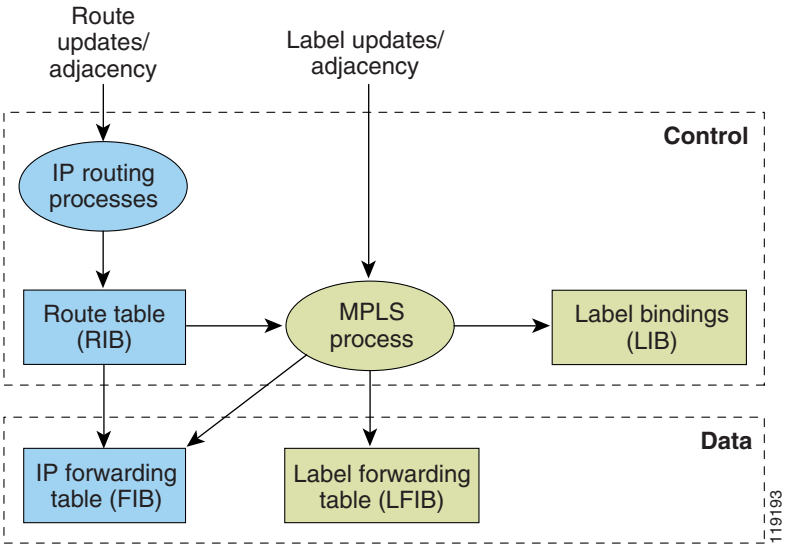
Supervisor Engine 32 PISA supports Multiprotocol Label Switching (MPLS) virtual private networks (VPNs), and Layer 2 Ethernet over MPLS (EoMPLS), with quality of service (QoS) and security.

The PISA on the supervisor engine performs Layer 3 control-plane functions, including address resolution and routing protocols. The PISA processes information from the Routing and Label Distribution Protocols and builds the IP forwarding (FIB) table and the label forwarding (LFIB) table. The PISA distributes the information in both tables to the PFC3B.

The PFC3B receives the information and creates its own copies of the FIB and LFIB tables. Together, these tables comprise the FIB TCAM. The PFC3B 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 21-2 shows the various functional blocks that support MPLS label switching. 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 21-2 MPLS Forwarding, Control and Data Planes



IP to MPLS

At the ingress to the MPLS network, the PFC3B 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 PFC3B 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 21-8.

MPLS to MPLS

At the core of an MPLS network, the PFC3B 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 PFC3B 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 PFC3B 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 PFC3B 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 PFC3B 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 PFC3B looks up the top label in the FIB TCAM and recirculates the packet. Then the PFC3B 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 PFC3B 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 PFC3B for additional processing.

Supported Hardware Features

The following hardware features are supported:

- 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 PFC3B provides the capability to recirculate the packets. See the "Recirculation" section on page 21-4.
- 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:

Note Multi-VPN Routing and Forwarding (VRF) for CE Routers (VRF Lite) is supported with the following features: IPv4 forwarding between VRFs interfaces, IPv4 ACLs, and IPv4 HSRP. IPv4 multicast is not supported.

- Multi-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:

  http://www.cisco.com/en/US/products/hw/routers/ps259/prod_bulletin09186a00800921d7.html.

- 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:

  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/12_2/switch/configuration/guide/xcftagc_ps1835_TSD_Products_Configuration_Guide_Chapter.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:

  http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsdserv3.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/12_2s/feature/guide/fstefa_3.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:

  http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsiarea3.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:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_l3_vpns/configuration/12-2sx/mp-l3-vpns-12-2sx-book.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:

  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:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_l3_vpns/configuration/12-2sx/mp-l3-vpns-12-2sx-book.html

- MPLS VPN Interautonomous System (InterAS) Support —This feature allows an MPLS VPN to span service providers and autonomous systems. See this publication:

  http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/fsias24.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:

  http://www.cisco.com/en/US/docs/ios-xml/ios/mp_ias_and_csc/configuration/12-2sx/mp-carrier-bgp.html

- MPLS VPN Hot Standby Router Protocol (HSRP) Support—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:

  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:

  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 21-13.

MPLS Guidelines and Restrictions

When configuring MPLS, follow these guidelines and restrictions:

- The PFC3B supports up to 8 load-shared paths. Cisco IOS releases for other platforms support only 4 load-shared paths.
- The PFC3B supports MTU checking and fragmentation.

Note Fragmentation is supported with software (for IP to MPLS path). See the **mtu** command in the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY.

Note For information on other limitations and restrictions, see the "MPLS VPN Guidelines and Restrictions" section on page 21-11 and the "EoMPLS Guidelines and Restrictions" section on page 21-14.

MPLS Supported Commands

The PFC3B MPLS supports these 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 these publications:

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

Configuring MPLS

For information about configuring MPLS, see the Cisco IOS software documentation at the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcftagov_ps1835_TSD_Products_Configuration_Guide_Chapter.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 PFC3B forwards MPLS labeled packets without explicit configuration. If the packet has three labels or less and the underlying packet is IPv4, then the PFC3B uses the source and destination IPv4 address. If the underlying packet is not IPv4 or more than three labels are present, the PFC3B 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 21-8.

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#sh ip ro 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#


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
```

VPN Switching

These sections describe VPN switching:

VPN Switching Operation

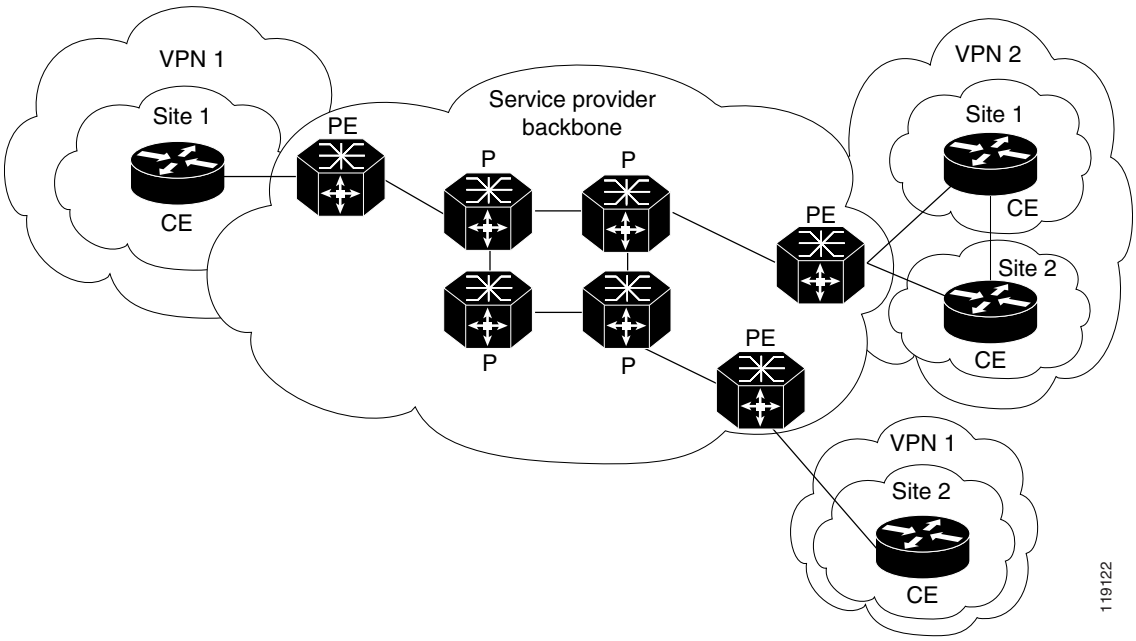
The IP VPN feature for MPLS allows a Cisco IOS network to deploy scalable IP Layer 3 VPN backbone services to multiple sites deployed on a shared infrastructure while also providing the same access or security policies as a private network. VPN based on MPLS technology provides the benefits of routing isolation and security, as well as simplified routing and better scalability.

Refer to 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 21-3.

Figure 21-3 VPNs with Service Provider Backbone



At the ingress PE, the PFC3B makes a forwarding decision based on the packet headers. The PFC3B contains a table that maps VLANs to VPNs. In the Catalyst 6500 series switch architecture, all physical ingress interfaces in the system are associated with a specific VPN. The PFC3B 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 line card 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 PFC3B 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 PFC3B 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, follow these guidelines and restrictions:

- The PFC3B supports a total of 1024 VRFs per chassis a with enhanced OSMs; using a nonenhanced OSM causes the system to default to 511 VRFs.
- 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 these publications:

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

Configuring MPLS VPN

For information on configuring MPLS VPN, refer to the “MPLS Virtual Private Networks” section at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcftagc_ps1835_TSD_Products_Configuration_Guide_Chapter.html

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, OSM, and FlexWAN CE-facing interfaces. MPLS switching configuration is identical to configuration on other platforms.

```
!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:

- 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.

The PFC3B supports both hardware-based EoMPLS as well as OSM-, FlexWAN, or FlexWAN2-based EoMPLS. For more information, see this URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SX_OSM_config/mpls.html

For information on requirements for Supervisor Engine 2-based EoMPLS, see this URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SX_OSM_config/mpls.html#Supervisor_Engine_2-Based_EoMPLS

For information on other AToM implementations (ATM AAL5 over MPLS, ATM Cell Relay over MPLS, Frame Relay over MPLS), see this URL:

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 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 OSM-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 an OSM. When local Layer 2 switching is not required, use PFC3B-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. The PFC3B 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.

Note The **no mls qos rewrite ip dscp** command is not compatible with the MPLS and MPLS VPN features. See Chapter 38, "Configuring PFC QoS."

Note Do not use the **no mls qos rewrite ip dscp** command if you have PFC3B-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 the supervisor engine 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 directed to the supervisor engine and not to the EoMPLS cloud.
  - The native VLAN of a trunk must not be configured as an EoMPLS VLAN.
- 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 PFC3B-based EoMPLS).
- EoMPLS 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 OSM and 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 21-16
- Configuring VLAN-Based EoMPLS, page 21-16
- Configuring Port-Based EoMPLS, page 21-19

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:

- 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 does not allow local switching of packets between interfaces unless you use loopback ports.
- A system can have both an OSM or FlexWAN configuration andEoMPLS configuration enabled at the same time. Cisco supports this configuration but does not recommend it. Unless the uplinks to the MPLS core are through OSM or FlexWAN-enabled interfaces, OSM or FlexWAN-based EoMPLS connections will not be active; this causes packets for OSM or FlexWAN-based EoMPLS arriving on non-WAN interfaces to be dropped. For information on WAN (FlexWAN and OSM) EoMPLS, see this URL:

  http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SX_OSM_config/mpls.html

LAN ports can receive Layer 2 traffic, impose labels, and switch the frames into the MPLS core without using an OSM or FlexWAN module.

You can configure an OSM or a FlexWAN module to face the core of MPLS network and use either the OSM configuration, the FlexWAN configuration, or the PFC3B configuration. For more information, see this URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/12.2SX_OSM_config/mpls.html

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, 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, 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
```

CHAPTER 22

Configuring IPv4 Multicast VPN Support

This chapter describes how to configure IPv4 Multicast Virtual Private Network (MVPN) support on Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter contains these sections:

Understanding How MVPN Works

These sections describe MVPN:

MVPN Overview

MVPN is a standards-based feature that transmits IPv4 multicast traffic across an MPLS VPN cloud. MVPN on Catalyst 6500 series switches uses the existing PFC3B hardware support for IPv4 multicast traffic 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.

1. If a particular multicast stream exceeds the defined threshold, the sending PE router dynamically creates a data MDT for that particular multicast traffic.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.

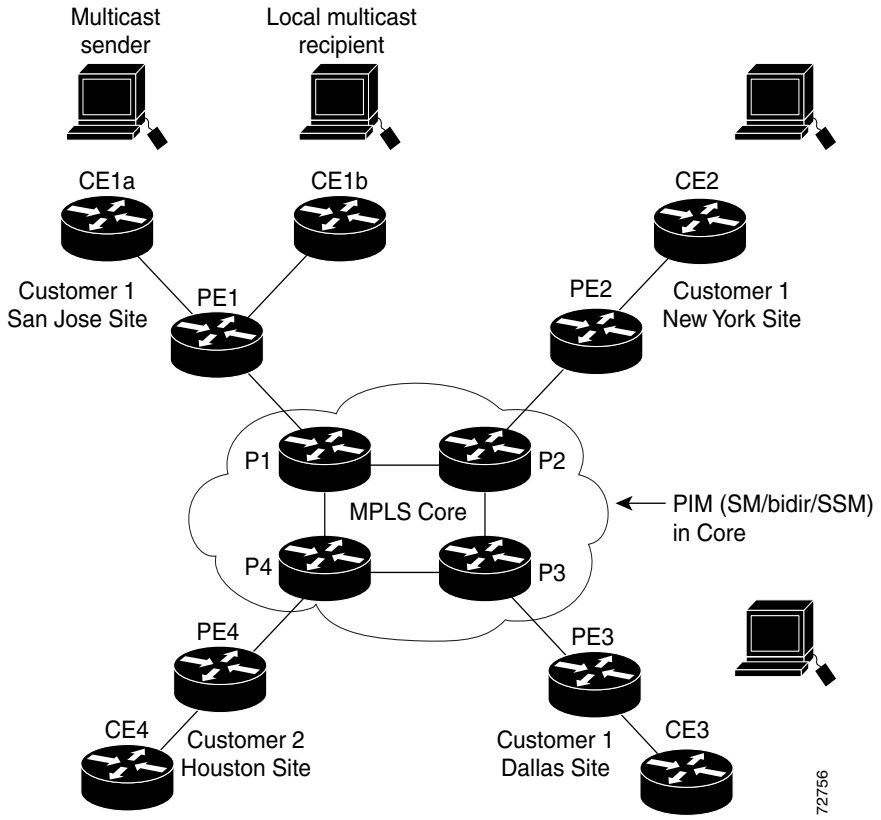
Note For technical information about the DATA-MDT JOIN message and other aspects of the data MDT creation and usage, see the Internet-Draft, *Multicast in MPLS/BGP IP VPNs*, by Eric C. Rosen et al.

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.

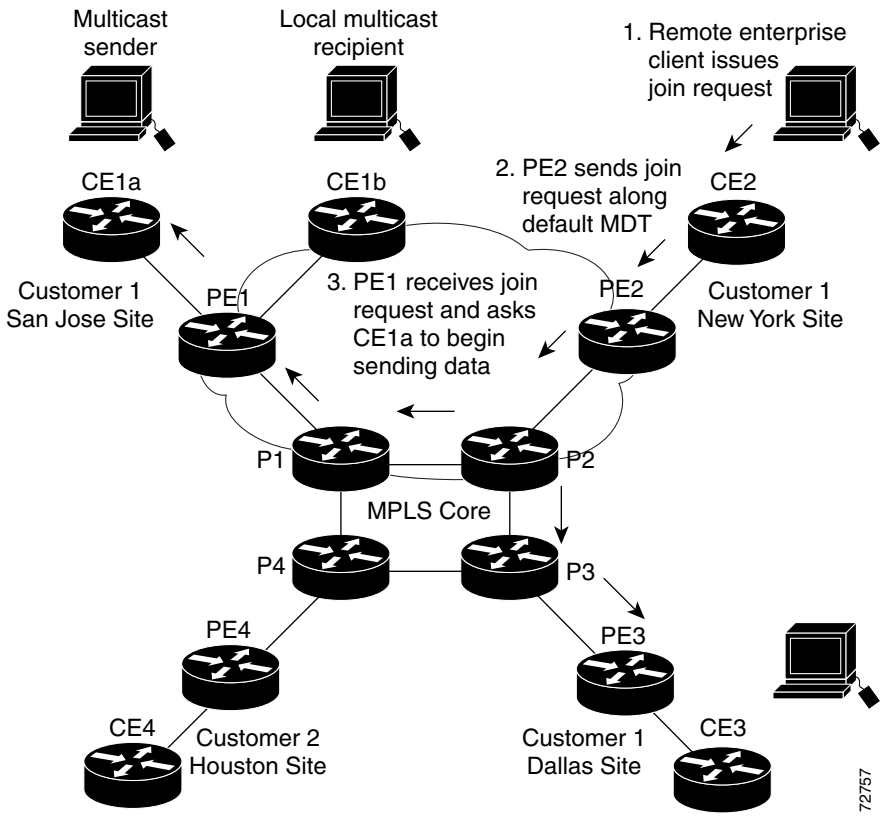
Figure 22-1 shows the situation in this network when no one outside of San Jose has joined the multicast broadcast, which means that no data is flowing along the default MDT. Each PE router maintains a PIM relationship with the other PE routers over the default MDT, as well as a PIM relationship with its directly attached PE routers.

Figure 22-1 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 22-2 shows how the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 22-2 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

Hardware acceleration for IPv4 multicast over VPN traffic forwards multicast traffic to the appropriate VPNs at wire speed without increased PISA CPU utilization.

In a customer VRF, hardware acceleration supports multicast traffic in PIM dense, PIM sparse, PIM bidirectional, and PIM Source Specific Multicast (SSM) modes.

In the service provider core, hardware acceleration supports multicast traffic in PIM sparse, PIM bidirectional, and PIM SSM modes. In the service provider core, hardware acceleration does not support multicast traffic in PIM dense mode.

MVPN Configuration Guidelines and Restrictions

When configuring MVPN, follow these guidelines and restrictions:

- 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 mode.
- 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 22-8
- Configuring a Multicast VPN Routing and Forwarding Instance, page 22-9
- Configuring Multicast VRF Routing, page 22-15
- Configuring Interfaces for Multicast Routing to Support MVPN, page 22-20

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).
	`Router(config)# no mls ip multicast replication-mode ingress`	Enables automatic detection of the replication mode.
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.
	`Router(config)# no ip vrf vrf_name`	Deletes the VRF entry.
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.
	`Router(config-vrf)# no rd route_distinguisher`	Deletes the route distinguisher.
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.
	`Router(config-vrf)# no route-target [[import | export | both] route_target_ext_community]`	Deletes the route-target extended community.
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.
`Router(config-vrf)# no mdt default`	Deletes 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.
`Router(config-vrf)# no mdt data`	Deletes the data MDT.

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.
`Router(config-vrf)# no log-reuse`	Disables data MDT logging.

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:

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.
	`Router(config)# no ip multicast-routing`	Disables 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.
`Router(config)# no ip multicast-routing`	Disables 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 refer to a VRF that has been previously created, as specified in the "Configuring a Multicast VPN Routing and Forwarding Instance" section on page 22-9.
- **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.
`Router(config)# no ip pim vrf vrf_name register-source`	Disables IPv4 multicast VRF routing.

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 Rendezvous Point (RP) Address

To specify the PIM VRF 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):
`Router(config)# no ip pim vrf vrf_name rp-address rp_address`	Clears the PIM RP IPv4 address.

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 a Multicast Source Discovery Protocol (MSDP) Peer

To configure an 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.
`Router(config)# no ip msdp vrf vrf_name peer {peer_name | peer_address} [connect-source interface_type interface_number] [remote-as ASN]`	Clears the PIM RP IPv4 address.

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.
`Router(config)# no ip multicast vrf vrf_name cache-headers [rtp]`	Disables IPv4 multicast header storage.

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.
`Router(config)# no ip multicast vrf vrf_name route-limit limit [threshold]`	Clears the configured maximum number of routes.

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.
`Router(config)# no ip multicast mrinfo-filter`	Clears the configured maximum number of routes.

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 22-20
- Configuring PIM on an Interface, page 22-20
- Configuring an Interface for IPv4 VRF Forwarding, page 22-21
- Sample Configuration, page 22-22

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.

	Command	Purpose
Step 3	`Router(config-if)# ip pim {dense-mode | sparse-mode | sparse-dense-mode}`	Enables PIM on the interface.
	`Router(config)# no ip pim [dense-mode | sparse-mode | sparse-dense-mode]`	Disables PIM.

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
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.
`Router(config-if)# no ip vrf forwarding [vrf_name]`	Disables IPv4 VRF forwarding.

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 22-22
- MVPN Configuration with Default and Data MDTs, page 22-24

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
...
```

CHAPTER 23

Configuring IP Unicast Layer 3 Switching

This chapter describes how to configure IP unicast Layer 3 switching on the Catalyst 6500 series switches.

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

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

This chapter consists of these sections:

Note

- IPX traffic is fast switched on the PISA. For more information, refer to 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 25, "Configuring IPv4 Multicast Layer 3 Switching."

Understanding How Layer 3 Switching Works

These sections describe Layer 3 switching:

Understanding Hardware Layer 3 Switching

Hardware Layer 3 switching allows the PFC3B , instead of the PISA, to forward IP unicast traffic between subnets. Hardware Layer 3 switching provides wire-speed forwarding on the PFC3B, instead of in software on the PISA. Hardware Layer 3 switching requires minimal support from the PISA. The PISA routes any traffic that cannot be hardware Layer 3 switched.

Hardware Layer 3 switching supports the routing protocols configured on the PISA. Hardware Layer 3 switching does not replace the routing protocols configured on the PISA.

Hardware Layer 3 switching runs on the PFC3B 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 NetFlow switching for TCP intercept, reflexive ACL forwarding decisions
- Hardware Cisco Express Forwarding (CEF) switching for all other IP unicast traffic

The PISA 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 Catalyst 6500 series 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 46, “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 Catalyst 6500 series switch performs a packet rewrite at the egress port based on information learned from the PISA so that the packets appear to have been routed by the PISA.

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 PISA to be routed to Destination B, the switch recognizes that the packet was sent to the Layer 2 (MAC) address of the PISA.

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 PISA. 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		
PISA 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:

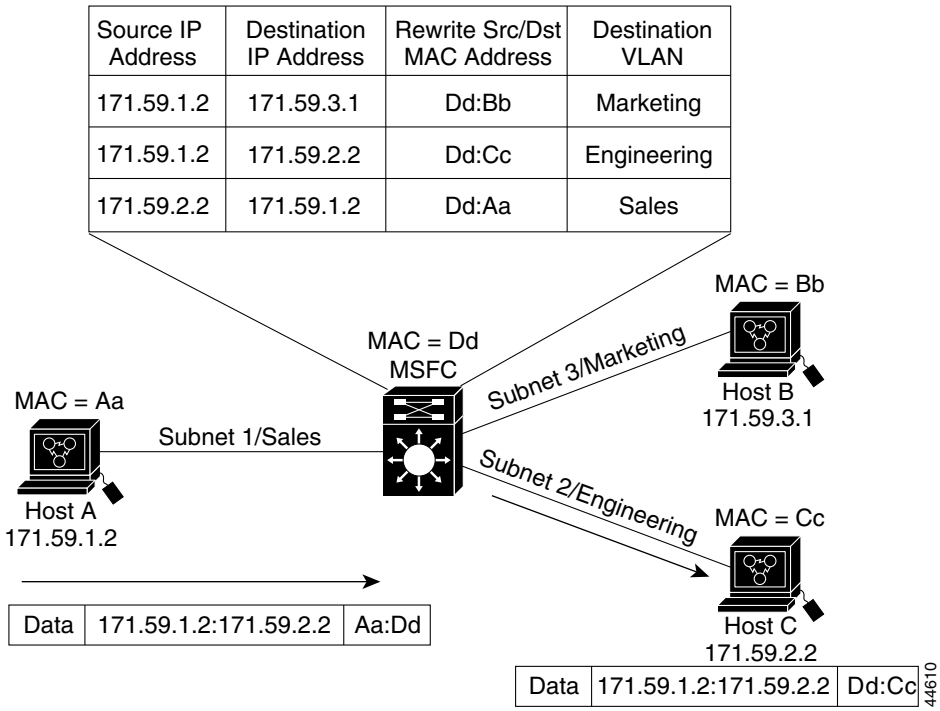
Layer 2 Frame Header		**Layer 3 IP Header**				**Data**	**FCS**
Destination	Source	Destination	Source	TTL	Checksum		
Destination B MAC	*PISA MAC*	*Destination B IP*	*Source A IP*	*n-1*	*calculation2*		

Hardware Layer 3 Switching Examples

Figure 23-1 on page 23-3 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 23-1 Hardware Layer 3 Switching Example Topology



Default Hardware Layer 3 Switching Configuration

Table 23-1 shows the default hardware Layer 3 switching configuration.

Table 23-1 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 PISA	Enabled (cannot be disabled)
Cisco IOS dCEF[1] enable state on PISA	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)
  - 802.3 with 802.2 and SNAP

Configuring Hardware Layer 3 Switching

Note For information on configuring unicast routing on the PISA, see Chapter 19, "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* = **ethernet**, **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.

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 load-balancing decisions are exactly the same, 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 PISA apply only to traffic that is CEF-switched in software on the PISA. The commands do not affect traffic that is hardware Layer 3 switched on the PFC3B.

For information about Cisco IOS CEF and dCEF on the PISA, refer to 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* = **ethernet**, **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* = **ethernet**, **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.

CHAPTER 24

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 on Catalyst 6500 series switches:

- The *Cisco IOS IPv6 Configuration Library*, "Implementing IPv6 Multicast":

  http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/configuration/12-2sx/ipv6-12-2sx-book.html
- The *Cisco IOS IPv6 Command Reference*:

  http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html

These sections provide additional information about IPv6 multicast support on 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

Features that Support IPv6 Multicast

These features support IPv6 multicast:

- RPR and RPR+ redundancy mode—See Chapter 6, "Configuring RPR Supervisor Engine Redundancy."
- Multicast Listener Discovery version 2 (MLDv2) snooping—See Chapter 26, "Configuring MLDv2 Snooping for IPv6 Multicast Traffic."

  Note MLDv1 snooping is not supported.

- IPv6 Multicast rate limiters—See Chapter 33, "Configuring Denial of Service Protection."
- IPv6 Multicast: Bootstrap Router (BSR)—See the BSR information in the *Cisco IOS IPv6 Configuration Library* and *Cisco IOS IPv6 Command Reference*.
- IPv6 Access Services—See DHCPv6 Prefix Delegation—See this publication:

  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:

  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 on Catalyst 6500 series switches:

- With Release 12.2(18)SXE and later releases, the PFC3 and DFC3 provide hardware support for the following:
  - Completely switched IPv6 multicast flows
  - IPv6 PIM-Sparse Mode (PIM-SM) (S,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 and RPR+ redundancy mode (see Chapter 6, "Configuring RPR Supervisor Engine Redundancy")
  - Ingress and egress PFC QoS (see Chapter 38, "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
  - PIM-SM (*,G) forwarding
  - 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

Refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY for information about these IPv6 multicast commands, which are new or changed in Release 12.2(18)SXE:

- **ipv6 mfib hardware-switching**
- **mls rate-limit multicast ipv6** (see Chapter 33, "Configuring Denial of Service Protection")
- **show ipv6 mfib**
- **show mls rate-limit** (see Chapter 33, "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 24-4

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 MSFC:

```
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 MSFC3.

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

CHAPTER 25

Configuring IPv4 Multicast Layer 3 Switching

This chapter describes how to configure IPv4 multicast Layer 3 switching on the Catalyst 6500 series switches.

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

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

This chapter consists of these sections:

Understanding How IPv4 Multicast Layer 3 Switching Works

These sections describe how IPv4 multicast Layer 3 switching works:

IPv4 Multicast Layer 3 Switching Overview

The Policy Feature Card 3B (PFC3B) 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 PFC3B.

The PFC3B supports hardware switching of (*,G) state flows. The PFC3B supports 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 PFC3B uses 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 27, "Configuring IGMP Snooping for IPv4 Multicast Traffic").

Multicast Layer 3 Switching Cache

This section describes how the PFC3B maintains Layer 3 switching information in hardware tables.

The PFC3B populates 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.

The PISA updates its multicast routing table and forwards the new information to the PFC3B whenever it receives traffic for a new flow. In addition, if an entry in the multicast routing table on the PISA ages out, the PISA deletes the entry and forwards the updated information to the PFC3B.

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 PFC3B maintains a list of outgoing interfaces for the IP multicast group. From this list, the PFC3B 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 PISA using the **no ip multicast-routing** command, all multicast Layer 3 switching cache entries on the PFC3B 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 PISA 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 PFC3B performs a packet rewrite that is based on information learned from the PISA 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 PFC3B 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 PFC3B 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 PFC3B 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 PISA (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 PFC3B replicates the rewritten packet onto the appropriate destination VLANs, where it is forwarded to members of IP multicast group G1.

After the PFC3B 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	*PISA 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 PISA 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 25-4
- Completely Switched Flows, page 25-4

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.

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 PFC3B prevents multicast traffic bridged on the source VLAN for that flow from reaching the PISA interface in that VLAN, freeing the PISA 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 PFC3B periodically sends multicast packet and byte count statistics for all completely switched flows to the PISA. The PISA 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:

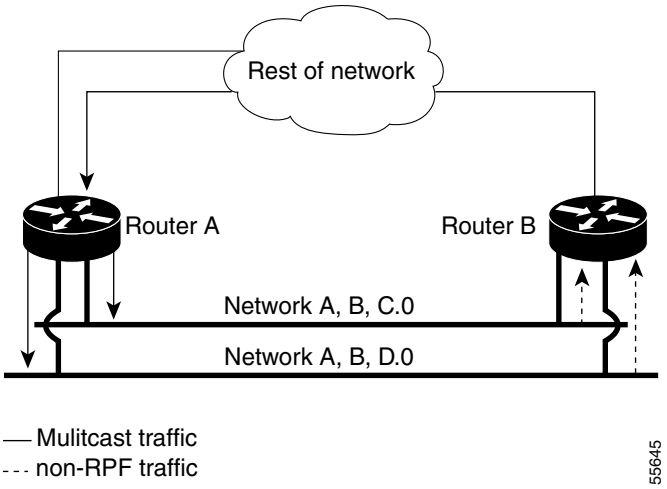
- Non-RPF Traffic Overview, page 25-5
- Filtering of RPF Failures for Stub Networks, page 25-5
- Rate Limiting of RPF Failure Traffic, page 25-6

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 25-1). 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 Catalyst 6500 series switch processes non-RPF traffic in hardware on the PFC3B by filtering (dropping) or rate limiting the non-RPF traffic.

Figure 25-1 Redundant Multicast Router Configuration in a Stub Network



Filtering of RPF Failures for Stub Networks

The PFC3B supports 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 PFC3B 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 25-13.

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 PFC3B bridges the packet to the PISA 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 PISA.

To support the PIM assert mechanism, the PFC3B periodically forwards a percentage of the non-RPF flow packets to the PISA.

The first packets for directly connected sources in PIM sparse mode are also rate-limited and are processed by the CPU.

Rate limiting of RPF failures is disabled by default.

Understanding How IPv4 Bidirectional PIM Works

The PFC3B supports hardware forwarding of IPv4 bidirectional PIM groups. To support IPv4 bidirectional PIM groups, the PFC3B 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 supervisor engine accepts packets from the RPF and from the DF interfaces.

When the supervisor engine 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 25-18.

Default IPv4 Multicast Layer 3 Switching Configuration

Table 25-1 shows the default IP multicast Layer 3 switching configuration.

Table 25-1 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 27, "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

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), refer to 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, refer to 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.
`Router(config)# no ip multicast-routing`	Disables 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.
	`Router(config-if)# no ip pim [dense-mode | sparse-mode | sparse-dense-mode]`	Disables IP PIM on a Layer 3 interface.

1. *type* = **ethernet**, **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
Router(config-if)#
```

This example shows how to enable PIM sparse mode on an interface:

```
Router(config-if)# ip pim sparse-mode
Router(config-if)#
```

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 25-9.

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)# no mls ip multicast`	Disables IP multicast Layer 3 switching on a Layer 3 interface.

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

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)#
```

Specifying the Maximum Number of Multicast Routes

By default, a PFC3B supports 32,000 multicast routes in sparse mode. The following default settings apply for maximum number of multicast routes:

- 32,000 for PIM-SM/DM/SSM for ingress- or egress-replication mode
- 32,000 for IPv4 bidirectional PIM ingress-replication mode
- 10,700 for IPv4 bidirectional PIM egress-replication mode

By entering the **mls ip multicast max-routes** command, you can increase the maximum number of multicast routes to 64,000 for PIM-SM/DM/SSM with either ingress- and egress-replication mode.

Note Rate limiting of directly connected sources is not available if you increase the maximum number of multicast routes above the default values.

To change the maximum number of multicast routes supported for PIM-SM/DM/SSM, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls ip multicast max-routes`	Specifies the maximum number of multicast routes.
Step 1	`Router# show mls ip multicast`	Displays the multicast route configuration.

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 PISA. 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.
`Router(config)# no mls ip multicast threshold`	Reverts to the default 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.
`Router(config)#` **`no mls ip multicast connected`**	Disables 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 supervisor engine forwards flow statistics messages to the PISA 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 PISA.

To specify how often flow statistics messages forwarded from the supervisor engine to the PISA, perform this task:

Command	Purpose
`Router(config)#` **`mls ip multicast flow-stat-timer`** *`num`*	Specifies how the supervisor engine forwards flow statistics messages to the PISA.
`Router(config)#` **`no mls ip multicast flow-stat-timer`** *`num`*	Restores the default.

This example shows how to configure the supervisor engine to forward flow statistics messages to the PISA 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 mintues.

To enable shortcut-consistency checking, perform this task:

Command	Purpose
`Router(config)# mls ip multicast consistency-check`	Enables shortcut-consistency checking.
`Router(config)# no mls ip multicast consistency-check num`	Restores the default.

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.
	`Router(config-if)# no mls ip multicast stub`	Disables ACL-based filtering of RPF failures on an interface.

1. *type* = **ethernet**, **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#
```

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 PISA 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* = **ethernet**, **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 indicatesthat 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* = **ethernet**, **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 25-18
- Configuring the Rendezvous Point for IPv4 Bidirectional PIM Groups, page 25-19
- Setting the IPv4 Bidirectional PIM Scan Interval, page 25-19
- Displaying IPv4 Bidirectional PIM Information, page 25-20

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.
`Router(config)# no ip pim bidir-enable`	Disables 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.
`Router(config)# no mls ip multicast bidir gm-scan-interval`	Restores the default.

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 25-2 describes IPv4 multicast Layer 3 switching debug commands that you can use to troubleshoot IP multicast Layer 3 switching problems.

Table 25-2 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 PFC3B. You can display entries based on any combination of the participating PISA, 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 25-17.

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.

CHAPTER 26

Configuring MLDv2 Snooping for IPv6 Multicast Traffic

This chapter describes how to configure Multicast Listener Discovery version 2 (MLDv2) snooping for IPv6 multicast traffic on the Catalyst 6500 series switches.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- To constrain IPv4 Multicast traffic, see Chapter 27, "Configuring IGMP Snooping for IPv4 Multicast Traffic."

This chapter consists of these sections:

Understanding How MLDv2 Snooping Works

These sections describe MLDv2 snooping:

MLDv2 Snooping Overview

MLDv2 snooping allows Catalyst 6500 series switches to examine MLDv2 packets and make forwarding decisions based on their content.

You can configure the switch to use MLDv2 snooping in subnets that receive MLDv2 queries from either MLDv2 or the MLDv2 snooping querier. MLDv2 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.

MLDv2, which runs at Layer 3 on a multicast router, generates Layer 3 MLDv2 queries in subnets where the multicast traffic needs to be routed. You can configure the MLDv2 snooping querier on the switch to support MLDv2 snooping in subnets that do not have any multicast router interfaces. For more information about the MLDv2 snooping querier, see the "Enabling the MLDv2 Snooping Querier" section on page 26-8.

MLDv2 (on a multicast router) or the MLDv2 snooping querier (on the supervisor engine) sends out periodic general MLDv2 queries that the switch forwards through all ports in the VLAN, and to which hosts respond. MLDv2 snooping monitors the Layer 3 MLDv2 traffic.

Note 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, MLDv2 snooping constrains the multicast traffic to only the multicast router ports.

MLDv2 Messages

These are the MLDv2 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

MLDv2 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 MLDv2 messages:

- Source lists
- INCLUDE or EXCLUDE mode

Because the Layer 2 table is (MAC-group, VLAN) based, with MLDv2 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

MLDv2 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 MLDv2 snooping software processes the MLDv2 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.

MLDv2 Snooping Proxy Reporting

Because MLDv2 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, MLDv2 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 MLDv2 report or by sending an MLDv2 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.

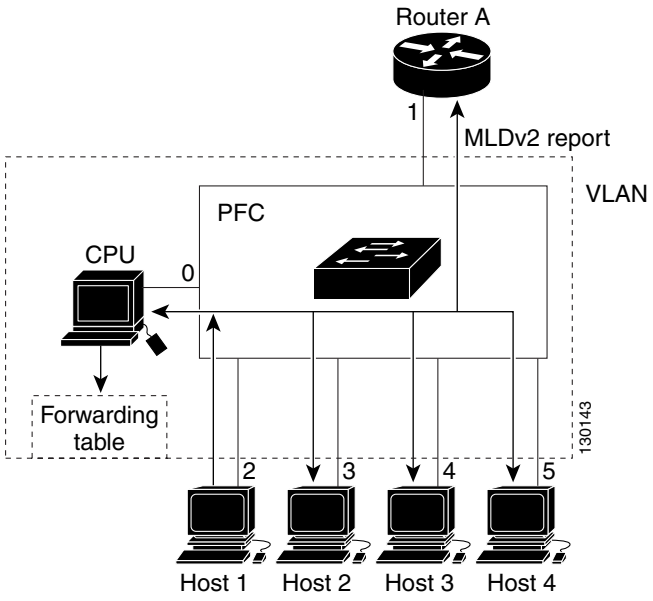
In response to a snooped MLDv2 report, the switch creates an entry in its Layer 2 forwarding table for the VLAN on which the report was received. When other hosts that are interested in this multicast traffic send MLDv2 reports, the switch snoops their reports and adds them to the existing Layer 2 forwarding 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 MLDv2 report.

MLDv2 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 26-1).

Layer 2 multicast groups learned through MLDv2 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 MLDv2 snooping learning. Multicast group membership lists can consist of both static and MLDv2 snooping-learned settings.

Figure 26-1 Initial MLDv2 Listener Report



Multicast router A sends an MLDv2 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 MLDv2 report to the group with the equivalent MAC destination address of 0x0100.5E01.0203.

When the switch snoops the MLDv2 report multicast by Host 1, the switch uses the information in the MLDv2 report to create a forwarding-table entry, as shown in Table 26-1, that includes the port numbers of Host 1, the multicast router, and the switch.

Table 26-1 MLDv2 Snooping Forwarding Table

Destination MAC Address	Type of Packet	Ports
0100.5exx.xxxx	MLDv2	0
0100.5e01.0203	!MLDv2	1, 2

The switch hardware can distinguish MLDv2 information packets from other packets for the multicast group. The first entry in the table indicates that only MLDv2 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 MLDv2 packets (!MLDv2) 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 MLDv2 report for the same group (Figure 26-2), the switch snoops that message and adds the port number of Host 4 to the forwarding table as shown in Table 26-2. Because the forwarding table directs MLDv2 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 26-2 Second Host Joining a Multicast Group

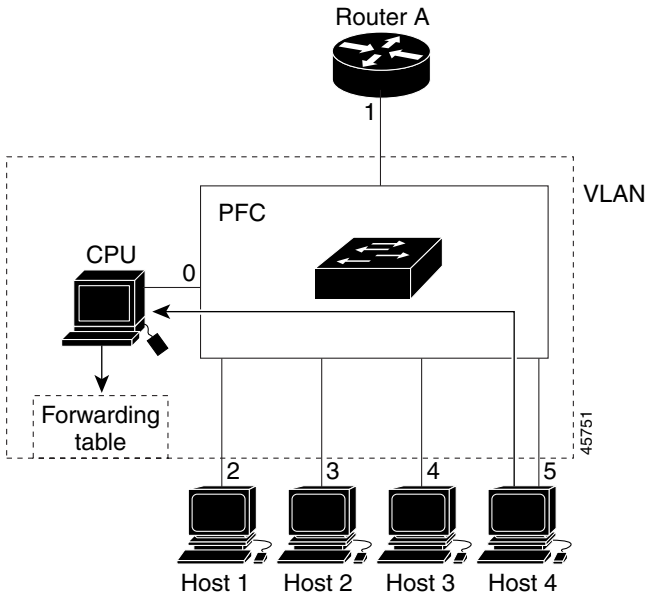


Table 26-2 Updated MLDv2 Snooping Forwarding Table

Destination MAC Address	Type of Packet	Ports
0100.5exx.xxxx	MLDv2	0
0100.5e01.0203	!MLDv2	1, 2, 5

Leaving a Multicast Group

These sections describe leaving a multicast group:

- Normal Leave Processing, page 26-6
- Fast-Leave Processing, page 26-6

Normal Leave Processing

Interested hosts must continue to respond to the periodic MLDv2 general queries. As long as at least one host in the VLAN responds to the periodic MLDv2 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 MLDv2 general queries (called a "silent leave"), or they can send an MLDv2 filter mode change record.

When MLDv2 snooping receives a filter mode change record from a host that configures the EXCLUDE mode for a group, MLDv2 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 MLDv2 snooping does not receive an MLDv2 report in response to the general query, MLDv2 snooping assumes that no other hosts connected to the interface are interested in receiving traffic for the specified multicast group, and MLDv2 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 MLDv2 snooping does not receive an MLDv2 report in response to the general query, MLDv2 snooping removes the group entry and relays the MLDv2 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 MLDv2 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 MLDv2 Snooping Querier

Use the MLDv2 snooping querier to support MLDv2 snooping in a VLAN where PIM and MLDv2 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 MLDv2 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 MLDv2 querier so that it can send queries.

When enabled, the MLDv2 snooping querier sends out periodic MLDv2 queries that trigger MLDv2 report messages from the switch that wants to receive IP multicast traffic. MLDv2 snooping listens to these MLDv2 reports to establish appropriate forwarding.

You can enable the MLDv2 snooping querier on all the Catalyst 6500 series switches in the VLAN, but for each VLAN that is connected to switches that use MLDv2 to report interest in IP multicast traffic, you must configure at least one switch as the MLDv2 snooping querier.

You can configure a switch to generate MLDv2 queries on a VLAN regardless of whether or not IP multicast routing is enabled.

Default MLDv2 Snooping Configuration

Table 26-3 shows the default MLDv2 snooping configuration.

Table 26-3 MLDv2 Snooping Default Configuration

Feature	Default Values
MLDv2 snooping querier	Disabled
MLDv2 snooping	Enabled
Multicast routers	None configured
MLDv2 report suppression	Enabled
MLDv2 snooping router learning method	Learned automatically through PIM or MLDv2 packets
Fast-Leave Processing	Enabled
MLDv2 Explicit Host Tracking	Enabled

MLDv2 Snooping Configuration Guidelines and Restrictions

When configuring MLDv2 snooping, follow these guidelines and restrictions:

- MLDv2 is derived from Internet Group Management Protocol version 3 (IGMPv3). MLDv2 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-v2.02.txt for detailed information on MLDv2 protocol.
- MLDv2 protocol messages are Internet Control Message Protocol version 6 (ICMPv6) messages.
- MLDv2 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.
- MLDv2 snooping supports private VLANs. Private VLANs do not impose any restrictions on MLDv2 snooping.
- MLDv2 snooping constrains traffic in MAC multicast groups 0100.5e00.0001 to 0100.5eff.ffff.
- MLDv2 snooping does not constrain Layer 2 multicasts generated by routing protocols.

MLDv2 Snooping Querier Configuration Guidelines and Restrictions

When configuring the MLDv2 snooping querier, follow these guidelines and restrictions:

- Configure the VLAN in global configuration mode (see Chapter 12, "Configuring VLANs").
- Configure an IPv6 address on the VLAN interface (see Chapter 19, "Configuring Layer 3 Interfaces"). When enabled, the MLDv2 snooping querier uses the IPv6 address as the query source address.
- If there is no IPv6 address configured on the VLAN interface, the MLDv2 snooping querier does not start. The MLDv2 snooping querier disables itself if the IPv6 address is cleared. When enabled, the MLDv2 snooping querier restarts if you configure an IPv6 address.
- When enabled, the MLDv2 snooping querier does not start if it detects MLDv2 traffic from an IPv6 multicast router.
- When enabled, the MLDv2 snooping querier starts after 60 seconds with no MLDv2 traffic detected from an IPv6 multicast router.
- When enabled, the MLDv2 snooping querier disables itself if it detects MLDv2 traffic from an IPv6 multicast router.
- QoS does not support MLDv2 packets when MLDv2 snooping is enabled.
- You can enable the MLDv2 snooping querier on all the Catalyst 6500 series switches in the VLAN that support it. One switch is elected as the querier.

Enabling the MLDv2 Snooping Querier

Use the MLDv2 snooping querier to support MLDv2 snooping in a VLAN where PIM and MLDv2 are not configured because the multicast traffic does not need to be routed.

To enable the MLDv2 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 MLDv2 snooping querier.
	`Router(config-if)# no ipv6 mld snooping querier`	Disables the MLDv2 snooping querier.

	Command	Purpose
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 MLDv2 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
Router#
```

Configuring MLDv2 Snooping

Note To use MLDv2 snooping, configure a Layer 3 interface in the subnet for IPv6 multicast routing or enable the MLDv2 snooping querier in the subnet (see the "Enabling the MLDv2 Snooping Querier" section on page 26-8).

These sections describe how to configure MLDv2 snooping:

Note Except for the global enable command, all MLDv2 snooping commands are supported only on VLAN interfaces.

Enabling MLDv2 Snooping

To enable MLDv2 snooping globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# ipv6 mld snooping`	Enables MLDv2 snooping.
	`Router(config)# no ipv6 mld snooping`	Disables MLDv2 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 MLDv2 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 MLDv2 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 MLDv2 snooping.
	`Router(config-if)# no ipv6 mld snooping`	Disables MLDv2 snooping.
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 MLDv2 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.
	`Router(config)# no mac-address-table static mac_addr vlan vlan_id`	Clears 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* = **ethernet**, **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* = **ethernet**, **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.
	`Router(config-if)# no ipv6 mld snooping last-member-query-interval`	Reverts to the default value.
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` `Router(config-if)# no ipv6 mld snooping fast-leave`	Enables fast-leave processing in the VLAN. Disables 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` `Router(config-if)# no ipv6 mld snooping ssm-safe-reporting`	Enables 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 MLDv6 Snooping Information

These sections describe displaying MLDv6 snooping information:

- Displaying Multicast Router Interfaces, page 26-14
- Displaying MAC Address Multicast Entries, page 26-14
- Displaying MLDv2 Snooping Information for a VLAN Interface, page 26-15

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 MLDv2 Snooping Information for a VLAN Interface

To display MLDv2 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 MLDv2 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      -
```

CHAPTER 27

Configuring IGMP Snooping for IPv4 Multicast Traffic

This chapter describes how to configure Internet Group Management Protocol (IGMP) snooping for IPv4 multicast traffic on the Catalyst 6500 series switches.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- To constrain IPv6 Multicast traffic, see Chapter 26, "Configuring MLDv2 Snooping for IPv6 Multicast Traffic."

This chapter consists of these sections:

Understanding How IGMP Snooping Works

These sections describe IGMP snooping:

IGMP Snooping Overview

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.

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 25, "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 "Enabling the IGMP Snooping Querier" section on page 27-8.

IGMP (on a multicast router) or the IGMP snooping querier (on the supervisor engine) 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).

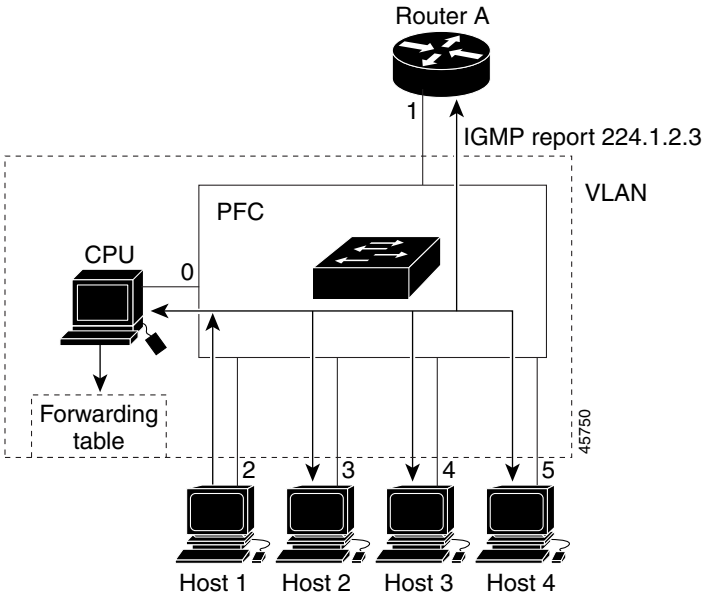
In response to an IGMP join request, the switch creates an entry in its Layer 2 forwarding table for the VLAN on which the join request was received. When other hosts that are interested in this multicast traffic send IGMP join requests, the switch adds them to the existing Layer 2 forwarding table entry. 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 27-1).

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 27-1 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 27-1, that includes the port numbers of Host 1, the multicast router, and the switch internal CPU.

Table 27-1 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 27-2), the CPU receives that message and adds the port number of Host 4 to the forwarding table as shown in Table 27-2. 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 27-2 Second Host Joining a Multicast Group

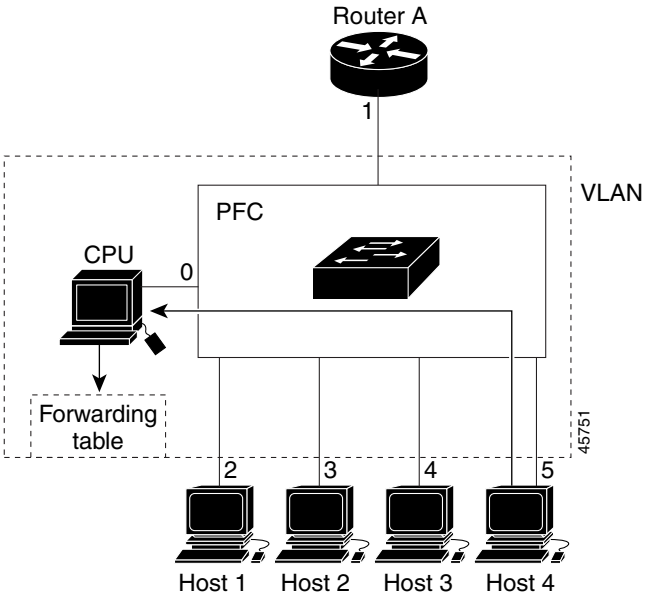


Table 27-2 Updated IGMP Snooping Forwarding Table

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 27-4
- Fast-Leave Processing, page 27-5

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." To configure the interval, enter the **ip igmp snooping last-member-query-interval** *interval* command.

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 IGMP 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.

You can enable the IGMP snooping querier on all the Catalyst 6500 series switches in the VLAN, but for each VLAN that is connected to switches that use IGMP to report interest in IP multicast traffic, you must configure at least one switch as the IGMP snooping querier.

You can configure a switch to generate IGMP queries on a VLAN regardless of whether or not IP multicast routing is enabled.

Understanding IGMP Version 3 Support

IGMP snooping supports IGMP version 3. IGMP version 3 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 IGMP version 3 snooping on a Catalyst 6500 series switch, the system maintains IGMP version 3 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 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.

IGMPv3 Fast-Leave Processing

IGMP version 3 fast-leave processing is enabled by default. To disable IGMP version 3 fast-leave processing you must turn off explicit-host tracking.

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 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.

Proxy Reporting

Because IGMPv3 does not have report suppression, all the hosts send their complete membership information to the router in response to queries. The switch receives these responses, updates the database and forwards the reports to the router. To prevent the router from becoming overloaded with reports, you can configure the switch for proxy-reporting mode. In proxy reporting mode, the switch forwards only the first report for a channel to the router and suppresses all other reports for the same channel.

With IGMPv3 proxy reporting, the switch does proxy reporting for unsolicited reports, as well as for reports received in the general query interval. Proxy reporting is turned on by default. When you disable proxy reporting, the switch works in transparent mode and updates the IGMP snooping database as it receives reports and forwards this information to the upstream router, which can then explicitly track all reporting hosts.

To support a mix of IGMPv2 and IGMPv3 hosts, the switch converts the IGMPv2 report into a EXCLUDE mode report. You must configure the switch to support both IGMPv2 and IGMPv3 hosts.

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 27-3 shows the default IGMP snooping configuration.

Table 27-3 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
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 PISA as a CGMP server. Refer to the *Cisco IOS IP Configuration Guide, Release 12.2*, “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, refer to 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:

- Configure the VLAN in global configuration mode (see Chapter 12, "Configuring VLANs").
- Configure an IP address on the VLAN interface (see Chapter 19, "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 supports IGMP version 2.
- When enabled, the IGMP snooping querier does not start if it detects IGMP traffic from a multicast router.
- When enabled, the IGMP snooping querier starts after 60 seconds with no IGMP traffic detected from a multicast router.
- When enabled, the IGMP snooping querier disables itself if it detects IGMP traffic from a multicast router.
- QoS does not support IGMP packets when IGMP snooping is enabled.
- You can enable the IGMP snooping querier on all the Catalyst 6500 series switches in the VLAN. One switch is elected as the querier.

Note When you are in configuration mode you can enter EXEC mode commands by entering the **do** keyword before the EXEC mode command.

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.
	`Router(config-if)# no ip igmp snooping querier`	Disables 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 172.20.52.106 255.255.255.248
Router(config-if)# igmp snooping querier
Router(config-if)# end
Router# show ip igmp interface vlan 200 | include querier
IGMP snooping querier is enabled on this interface
Router#
```

Configuring IGMP Snooping

Note To use IGMP snooping, configure a Layer 3 interface in the subnet for multicast routing (see Chapter 25, "Configuring IPv4 Multicast Layer 3 Switching") or enable the IGMP snooping querier in the subnet (see the "Enabling the IGMP Snooping Querier" section on page 27-8).

IGMP snooping allows Catalyst 6500 series switches to examine IGMP packets and make forwarding decisions based on their content.

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.
	`Router(config)# no ip igmp snooping`	Disables 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.
	`Router(config-if)# no ip igmp snooping`	Disables 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 vl25 | include snooping
  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#
```

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.
	`Router(config)# no mac-address-table static mac_addr vlan vlan_id`	Clears 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* = **ethernet**, **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-if)# ip igmp snooping mrouter interface type[1] slot/port`	Configures a static connection to a multicast router.
Step 2	`Router(config-if)# end`	Exits configuration mode.
Step 3	`Router# show ip igmp snooping mrouter`	Verifies the configuration.

1. *type* = **ethernet**, **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)# ip igmp snooping mrouter interface fastethernet 5/6
Router(config-if)#
```

Configuring the IGMP 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 IGMP fast-leave processing and the IGMP query interval are configured, fast-leave processing takes precedence.

To configure the interval for the IGMP 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)# ip igmp snooping last-member-query-interval interval`	Configures the interval for the IGMP snooping queries sent by the switch. Default is 1 second. Valid range is 100 to 999 milliseconds.
	`Router(config-if)# no ip igmp snooping last`	Reverts to the default value.

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

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 fast-leave`	Enables IGMP fast-leave processing in the VLAN.
	`Router(config-if)# no ip igmp snooping fast-leave`	Disables IGMP fast-leave processing in the VLAN.

This example shows how to enable IGMP fast-leave processing on the VLAN 200 interface and verify the configuration:

```
Router# interface vlan 200
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
Router(config-if)#
```

Configuring Source Specific Multicast (SSM) Mapping

Note Do not configure SSM mapping in a VLAN that supports IGMPv3 multicast receivers.

To configure SSM mapping, refer to this publication:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipmulti_igmp/configuration/12-2sx/imc_ssm_mapping.html

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 snooping explicit-tracking`	Enables explicit host tracking.
	`Router(config-if)# no ip igmp snooping explicit-tracking`	Clears the explicit host tracking configuration.
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 snooping explicit-tracking
Router(config-if)# end
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
```

Displaying IGMP Snooping Information

These sections describe displaying IGMP snooping information:

- Displaying Multicast Router Interfaces, page 27-14
- Displaying MAC Address Multicast Entries, page 27-14
- Displaying IGMP Snooping Information for a VLAN Interface, page 27-15
- Displaying IGMP Snooping Statistics, page 27-15

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 ip igmp snooping mrouter** *vlan_ID*	Displays multicast router interfaces.

This example shows how to display the multicast router interfaces in VLAN 1:

```
Router# show ip igmp 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 IGMP Snooping Information for a VLAN Interface

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   :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      -
Router#
```

CHAPTER 28

Configuring PIM Snooping

This chapter describes how to configure protocol independent multicast (PIM) snooping on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding How PIM Snooping Works

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.

Note To use PIM snooping, you must enable IGMP snooping on the Catalyst 6500 series 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 28-1 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 28-1 PIM Join Message Flow without PIM Snooping

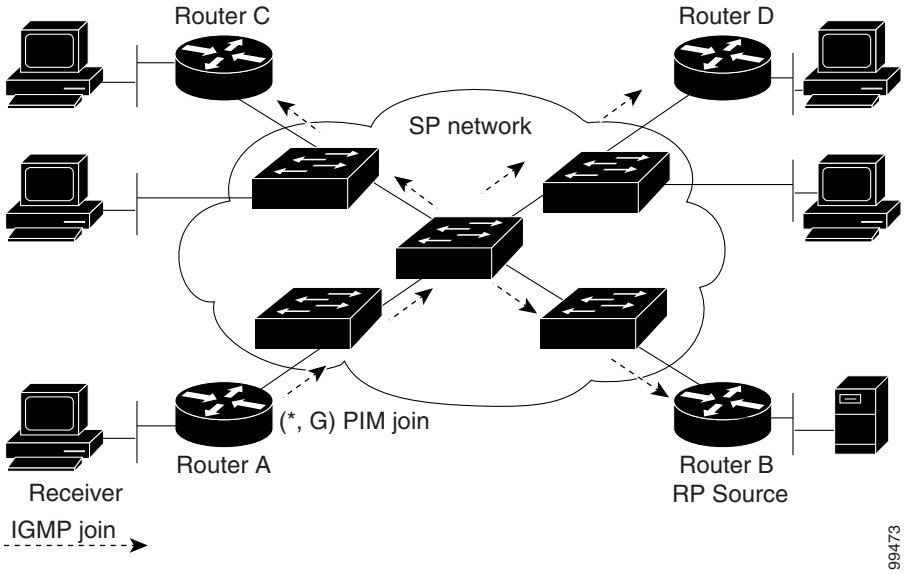


Figure 28-2 shows the flow of a PIM join message with PIM snooping enabled. In the figure, the switches restrict the PIM join message and forward it only to the router that needs to receive it (Router B).

Figure 28-2 PIM Join Message Flow with PIM Snooping

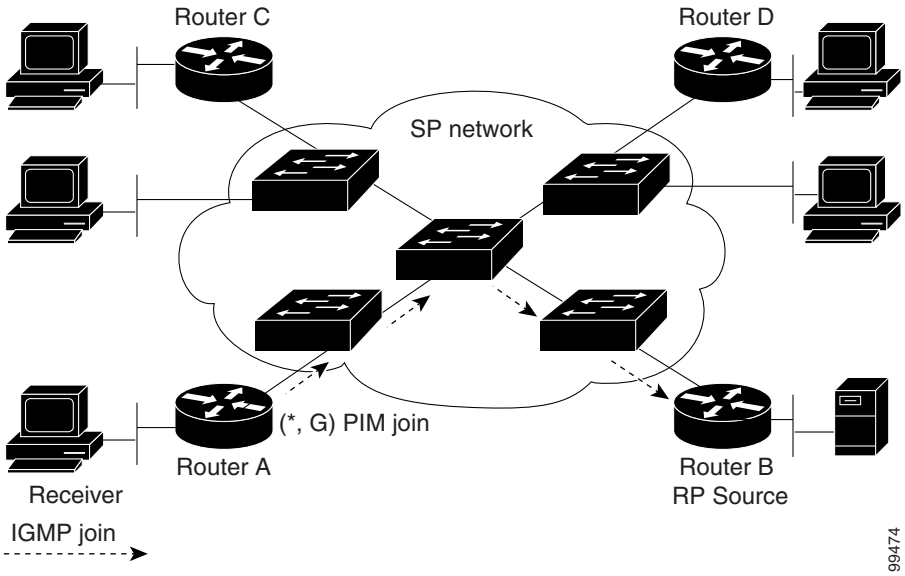


Figure 28-3 shows the flow of data traffic without PIM snooping enabled. In the figure, the switches flood the data traffic intended for Router A to all connected routers.

Figure 28-3 Data Traffic Flow without PIM Snooping

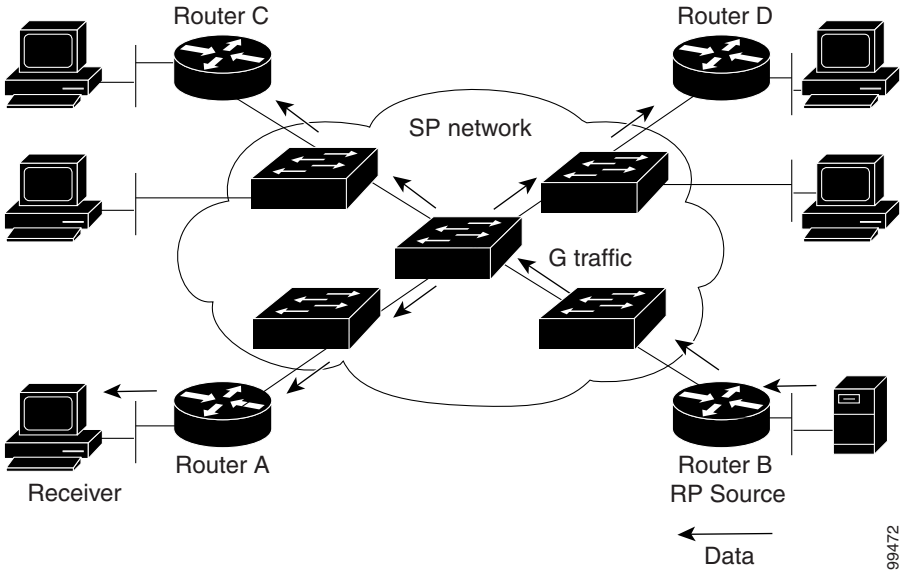
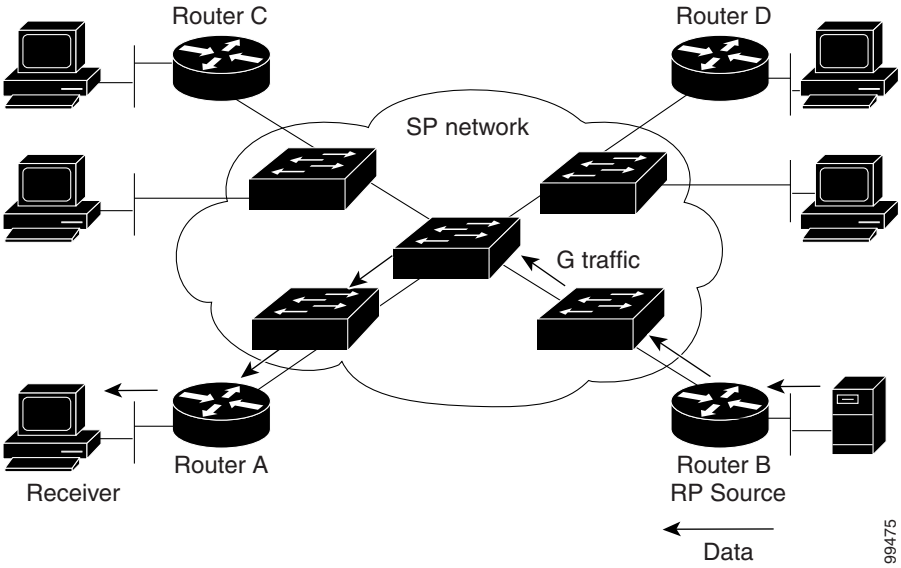


Figure 28-4 shows the flow of data traffic with PIM snooping enabled. In the figure, the switches forward the data traffic only to the router that needs to receive it (Router A).

Figure 28-4 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 28-5
- Enabling PIM Snooping in a VLAN, page 28-5
- Disabling PIM Snooping Designated-Router Flooding, page 28-6

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

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
```

CHAPTER 29

Configuring RGMP

This chapter supplements the information and procedures about Router-Port Group Management Protocol (RGMP) in the Release 12.2 publication at this URL:

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

This chapter consists of these sections:

Understanding How RGMP Works

RGMP constrains multicast traffic that exits the Catalyst 6500 series switch through ports to which only disinterested multicast routers are connected. RGMP reduces network congestion by forwarding multicast traffic to only those routers that are configured to receive it.

Note To use RGMP, you must enable IGMP snooping on the Catalyst 6500 series switch. IGMP snooping constrains multicast traffic that exits through LAN ports to which hosts are connected. IGMP snooping does not constrain traffic that exits through LAN ports to which one or more multicast routers are connected.

Note You must enable Protocol Independent Multicast (PIM) on all routers and switches for RGMP to work. Only PIM sparse mode is currently supported.

All routers on the network must be RGMP-capable. RGMP-capable routers send RGMP hello messages periodically. The RGMP hello message tells the Catalyst 6500 series switch not to send multicast data to the router unless an RGMP join message has also been sent to the Catalyst 6500 series switch from that router. When an RGMP join message is sent, the router is able to receive multicast data.

To stop receiving multicast data, a router must send an RGMP leave message to the Catalyst 6500 series switch. To disable RGMP on a router, the router must send an RGMP bye message to the Catalyst 6500 series switch.

Table 29-1 provides a summary of the RGMP packet types.

Table 29-1 RGMP Packet Types

Description	Action
Hello	When RGMP is enabled on the router, no multicast data traffic is sent to the router by the Catalyst 6500 series switch unless an RGMP join is specifically sent for a group.
Bye	When RGMP is disabled on the router, all multicast data traffic is sent to the router by the Catalyst 6500 series switch.
Join	Multicast data traffic for a multicast MAC address from the Layer 3 group address G is sent to the router. These packets have group G in the Group Address field of the RGMP packet.
Leave	Multicast data traffic for the group G is not sent to the router. These packets have group G in the group address field of the RGMP packet.

Default RGMP Configuration

RGMP is permanently enabled on Layer 2 LAN ports. RGMP is disabled by default on Layer 3 interfaces.

RGMP Configuration Guidelines and Restrictions

When configuring RGMP, follow these guidelines and restrictions:

- Either RGMP or PIM snooping can be enabled in a VLAN but not both.
- RGMP supports PIM sparse mode. RGMP does not support PIM dense mode. RGMP explicitly supports the two AutoRP groups in dense mode by not restricting traffic to those groups but by flooding it to all router ports. For this reason, you should configure PIM sparse-dense mode. If you configure groups other than the AutoRP groups for dense mode, their traffic will not be correctly forwarded through router ports that have been enabled for RGMP.
- To effectively constrain multicast traffic with RGMP, connect RGMP-enabled routers to separate ports on RGMP-enabled Catalyst 6500 series switches. (VLAN interfaces satisfy this restriction.)
- RGMP only constrains traffic that exits through LAN ports on which it detects an RGMP-enabled router. If a non-RGMP enabled router is detected on a LAN port, that port receives all multicast traffic.
- RGMP does not support directly connected multicast sources in the network. A directly connected multicast source will send multicast traffic into the network without signaling through RGMP or PIM. This multicast traffic will not be received by an RGMP-enabled router unless the router already requested receipt of that multicast group through RGMP. This restriction applies to hosts and to functions in routers that source multicast traffic, such as the **ping** and **mtrace** commands and multicast applications that source multicast traffic, such as UDPTN.
- RGMP supports directly connected receivers in the network. Traffic to these receivers will be constrained by IGMP snooping, or if the receiver is a router itself, by PIM and RGMP.
- CGMP is not supported in networks where RGMP is enabled on routers. You cannot enable both RGMP and CGMP on a Layer 3 interface. If RGMP is enabled on a Layer 3 interface, CGMP is silently disabled and vice versa.

- The following properties of RGMP are the same as for IGMP snooping:
  - RGMP constrains traffic based on the multicast group, not on the sender's IP address.
  - If spanning tree topology changes occur in the network, the state is not flushed as it is with Cisco Group Management Protocol (CGMP).
  - RGMP does not constrain traffic for multicast groups 224.0.0.x (x = 0...255), which allows use of PIMv2 bootstrap router (BSR) in an RGMP-controlled network.
  - RGMP in Cisco network devices operates on MAC addresses, not on IP multicast addresses. Because multiple IP multicast addresses can map to one MAC address (see RFC 1112), RGMP cannot differentiate between the IP multicast groups that might map to a MAC address.
  - The capability of the Catalyst 6500 series switch to constrain traffic is limited by its content-addressable memory (CAM) table capacity.

Enabling RGMP on Layer 3 Interfaces

To enable RGMP on a Layer 3 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)# ip rgmp`	Enables RGMP on the Layer 3 interface.
	`Router(config-if)# no ip rgmp`	Disables RGMP on the Layer 3 interface.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# debug ip rgmp [name_or_group_address]`	(Optional) Monitors RGMP.

1. *type* = **ethernet**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**, or **ge-wan**

This example shows how to configure RGMP on FastEthernet port 3/3:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/3
Router(config-if)# ip rgmp
Router(config-if)# end
Router#
```

CHAPTER 30

Configuring Network Security

This chapter contains network security information unique to the Catalyst 6500 series switches, 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, refer to these publications:

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

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, refer to 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 Cisco IOS Unicast Reverse Path Forwarding check (Unicast RPF check):

- Understanding PFC3B Unicast RPF Check Support, page 30-2
- Unicast RPF Check Guidelines and Restrictions, page 30-3
- Configuring Unicast RPF Check, page 30-3

Understanding PFC3B Unicast RPF Check Support

For a complete explanation of how Unicast RPF check works, refer to the *Cisco IOS Security Configuration Guide*, Release 12.2, "Other Security Features," "Configuring Unicast Reverse Path Forwarding" at this URL:

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

The PFC3B provides hardware support for RPF check of traffic from multiple interfaces.

With strict-method Unicast RPF check, the PFC3B supports two parallel paths for all prefixes in the routing table, and up to four parallel paths for prefixes reached through any of four user-configurable RPF interface groups (each interface group can contain four interfaces).

With loose-method Unicast RPF check (also known as exist-only method), the PFC3B supports up to eight reverse-path interfaces (the Cisco IOS software is limited to eight reverse paths in the routing table).

There are four methods of performing Unicast RPF check in Cisco IOS:

- Strict Unicast RPF check

- Strict Unicast RPF check with allow-default
- Loose Unicast RPF check
- Loose Unicast RPF check with allow-default

You configure Unicast RPF check on a per-interface basis, but the PFC3B supports only one Unicast RPF method for all interfaces that have Unicast RPF check enabled. When you configure an interface to use a Unicast RPF method that is different from the currently configured method, all other interfaces in the system that have Unicast RPF check enabled use the new method.

Unicast RPF Check Guidelines and Restrictions

When configuring Unicast RPF check, follow these guidelines and restrictions:

- If you configure Unicast RPF check to filter with an ACL, the PFC3B determines whether or not traffic matches the ACL. The PFC3B sends the traffic denied by the RPF ACL to the PISA for the Unicast RPF check. Packets permitted by the ACL are forwarded in hardware without a Unicast RPF check (CSCdz35099).
- Because the packets in a denial-of-service attack typically match the deny ACE and are sent to the PISA for the Unicast RPF check, they can overload the PISA.
- The PFC3B provides hardware support for traffic that does not match the Unicast RPF check ACL, but that does match an input security ACL.
- The PFC3B does not provide hardware support for the Unicast RPF check for policy-based routing (PBR) traffic. (CSCea53554)

Configuring Unicast RPF Check

These sections describe how to configure Unicast RPF check:

- Configuring the Unicast RPF Check Mode, page 30-3
- Configuring the Multiple-Path Unicast RPF Check Mode on a PFC3B, page 30-5
- Enabling Self-Pinging, page 30-6

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.
	`Router(config-if)# no ip verify unicast`	Reverts to the default 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* = **ethernet**, **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 on a PFC3B

To configure the multiple-path Unicast RPF check mode on a PFC3B, 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 on a PFC3B.
	`Router(config)# no mls ip cef rpf mpath {punt | interface-group}`	Returns to the default (**mls ip cef rpf mpath punt**).
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls cef ip rpf`	Verifies the configuration.

When configuring the multiple path RPF check mode, note the following information:

- **punt** mode (default)—The PFC3B 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 PISA for Unicast RPF check in software.
- **pass** mode—The PFC3B 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 PFC3B performs the Unicast RPF check in hardware for single-path and two-path prefixes. The PFC3B 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 on a PFC3B

To configure multiple-path Unicast RPF interface groups on a PFC3B, 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 on a PFC3B.
Step 2	`Router(config)# mls ip cef rpf interface-group group_number`	Removes an interface group.
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.
	`Router(config-if)# no ip verify unicast source reachable-via any allow-self-ping`	Disables self-pinging.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.

1. *type* = **ethernet**, **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
```

CHAPTER 31

Understanding Cisco IOS ACL Support

This chapter describes Cisco IOS ACL support on the Catalyst 6500 series switches:

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

Cisco IOS ACL Configuration Guidelines and Restrictions

The following guidelines and restrictions apply to Cisco IOS ACL configurations:

- You can apply Cisco IOS ACLs directly to Layer 3 ports and to VLAN interfaces.
- You can apply VLAN ACLs (VACLs) to VLANs (refer to Chapter 32, "Configuring 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 PFC3B does not provide hardware support for Cisco IOS IPX ACLs. Cisco IOS IPX ACLs are supported in software on the PISA.
- By default, the PISA 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 supervisor engine drops most of the denied packets in hardware and sends only a small number of packets to the PISA to be dropped (10 packets per second, maximum), which generates ICMP-unreachable messages.

  To eliminate the load imposed on the PISA 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.

- We strongly recommend that you use named ACLs (rather than numbered ACLs) as this conserves 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.

Hardware and Software ACL Support

Access control lists (ACLs) can be processed in hardware by the PFC3B or in software by the PISA. The following behavior describes software and hardware handling of ACLs:

- The PFC3B provides more efficient hardware support for named ACLs than it can for numbered ACLs.
- 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) flows are processed in hardware. If a field specified in a VACL is not supported by hardware processing 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. Catalyst 6500 series switches do 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 PISA for software processing without impacting other flows.
- The PFC3B does not provide hardware support for Cisco IOS IPX ACLs. Cisco IOS IPX ACLs are supported in software on the PISA.
- 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 with a PFC3B" section on page 31-3).
- 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.
- 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.

Optimized ACL Logging with a PFC3B

These sections describe optimized ACL logging (OAL):

Understanding OAL

OAL provides hardware support for ACL logging. Unless you configure OAL, packets that require logging are processed completely in software on the PISA. OAL permits or drops packets in hardware on the PFC3B and uses an optimized routine to send information to the PISA 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 PFC3B.
- 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)
  - 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 PISA 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 31-4
- Configuring OAL on an Interface, page 31-5
- Displaying OAL Information, page 31-5
- Clearing Cached OAL Entries, page 31-5

Note

- For complete syntax and usage information for the commands used in this section, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY.
- 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.
`Router(config)# no logging ip access-list cache {entries | interval | rate-limit | threshold}`	Reverts OAL global parameters to defaults.

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.
	`Router(config-if)# no logging ip access-list cache`	Disables OAL on the interface.
Step 3	`Router(config-if)# logging ip access-list cache out`	Enables OAL for egress traffic on the interface.
	`Router(config-if)# no logging ip access-list cache`	Disables OAL 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 31-6
- Determining Logical Operation Unit Usage, page 31-6

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 31-6 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)

CHAPTER 32

Configuring VLAN ACLs

This chapter describes how to configure VLAN ACLs (VACLs) on Catalyst 6500 series switches.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- OAL and VACL capture are incompatible. Do not configure both features on the switch. With OAL configured (see the "Optimized ACL Logging with a PFC3B" section on page 31-3), use SPAN to capture traffic.

This chapter consists of these sections:

Understanding VACLs

These sections describe VACLs:

VACL Overview

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 regular Cisco IOS standard or extended ACLs that are configured on router interfaces only and 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 hardware. VACLs use Cisco IOS ACLs. VACLs ignore any Cisco IOS ACL fields that are not supported in hardware.

You can configure VACLs for IP, IPX, and MAC-Layer traffic. VACLs applied to WAN interfaces support only IP traffic for VACL capture.

When you configure a VACL and apply it to a VLAN, all packets entering the VLAN are checked against this VACL. If you apply a VACL to the VLAN and an ACL to a routed interface in the VLAN, a packet coming in to the VLAN is first checked against the VACL and, if permitted, is then checked against the input ACL before it is handled by the routed interface. When the packet is routed to another VLAN, it is first checked against the output ACL applied to the routed interface and, if permitted, the VACL configured for the destination VLAN is applied. If a VACL is configured for a packet type and a packet of that type does not match the VACL, the default action is deny.

Note

- TCP Intercepts and Reflexive ACLs take precedence over a VACL action if these are configured on the same interface.
- VACLs and CBAC cannot be configured on the same interface.
- IGMP packets are not checked against VACLs.

Bridged Packets

Figure 32-1 shows a VACL applied on bridged packets.

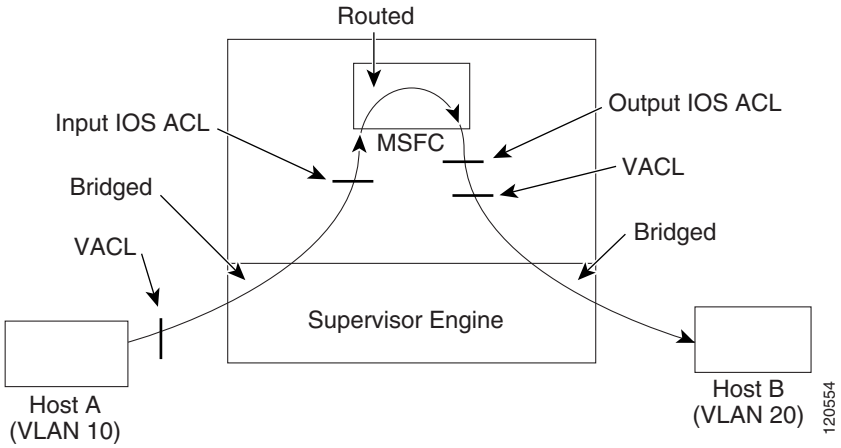
Figure 32-1 Applying VACLs on Bridged Packets

Routed Packets

Figure 32-2 shows how ACLs are applied on routed and Layer 3-switched packets. For routed or Layer 3-switched packets, the ACLs are applied in the following order:

1. VACL for input VLAN
2. Input Cisco IOS ACL
3. Output Cisco IOS ACL
4. VACL for output VLAN

Figure 32-2 Applying VACLs on Routed Packets

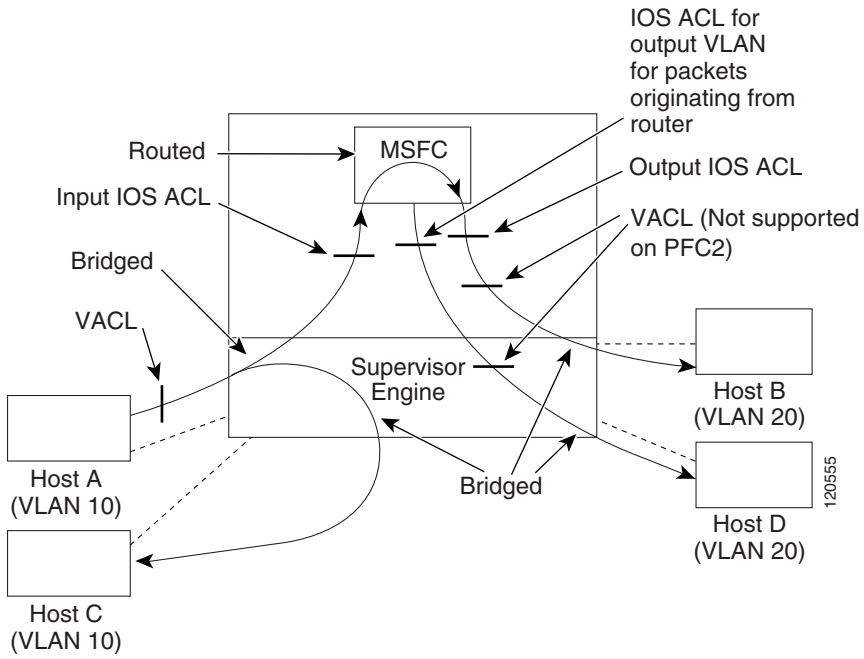


Multicast Packets

Figure 32-3 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. VACL for input VLAN
   b. Input Cisco IOS ACL
2. Packets after multicast expansion:
   a. Output Cisco IOS ACL
   b. VACL for output VLAN
3. Packets originating from router—VACL for output VLAN

Figure 32-3 Applying VACLs on Multicast Packets



Configuring VACLs

These sections describe how to configure VACLs:

VACL Configuration Overview

VACLs use standard and extended Cisco IOS IP and IPX ACLs, and MAC Layer-named ACLs (see the "Configuring MAC ACLs" section on page 38-54) 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 with a match clause and an action clause. The match clause specifies IP, IPX, 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 use access control for 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 use access control for both the input and output routed traffic. You can define a VACL to use access control for 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.

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.

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.

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.

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 32-9.

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} | ipx address {800-999 | 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} | ipx address {800-999 | 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, refer to the "Configuring MAC ACLs" section on page 38-54.
- For information about Cisco IOS ACLs, refer to 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 the "VLAN Access Map Configuration and Verification Examples" section on page 32-9.

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]} | {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]} | {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 32-9.
- 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 32-9.

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 32-9.

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 8-8 and the "Configuring the Layer 2 Trunk Not to Use DTP" section on page 8-9).

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**.
	`Router(config-if)# no switchport capture allowed vlan`	Clears the configured destination VLAN list and returns to the default value (**all**).
Step 3	`Router(config-if)# switchport capture`	Configures the port to capture VACL-filtered traffic.
	`Router(config-if)# no switchport capture`	Disables the capture function on the interface.

1. *type* = **ethernet**, **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 8-8) 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 8-14) 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 mordred
Vlan access-map "mordred"  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 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 VACLs" section on page 32-4 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
```

CHAPTER 33

Configuring Denial of Service Protection

This chapter contains information on how to protect your Catalyst 6500 series switch against Denial of Service (DoS) attacks. The information covered in this chapter is unique to the Catalyst 6500 series switches, 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, refer to these publications:

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

This chapter consists of these sections:

Understanding How DoS Protection Works

This section contains information about the available methods to counteract DoS attacks with a PFC3B and includes configuration examples. The PFC3B 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 the "Understanding How Control Plane Policing Works" section on page 33-18.

These sections describe DoS protection with a PFC3B:

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 Catalyst 6500 series 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 on the Catalyst 6500 series switches.

QoS Rate Limiting

QoS ACLs limit the amount of a particular type of traffic that is processed by the PISA. If a DoS attack is initiated against the PISA, QoS ACLs can prevent the DoS traffic from reaching the PISA data path and congesting it. The PFC3B 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 PISA.

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 PISA 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 FIB tables on the supervisor engine.

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 PFC3B 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 with a PFC3B 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 PISA 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 Catalyst 6500 series 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 PISA 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 PFC3B.

ARP Policing

During an attack, malicious users may try to overwhelm the PISA CPU with control packets such as routing protocol or ARP packets. These special control packets can be hardware rate limited using a specific routing protocol and an ARP policing mechanism configurable with the **mls qos protocol** command. The routing protocols supported include RIP, BGP, LDP, OSPF, IS-IS, IGRP, and EIGRP. For example, the command **mls qos protocol arp police 32000** rate limits ARP packets in hardware at 32,000 bps. Although this policing mechanism effectively protects the PISA CPU against attacks such as line-rate ARP attacks, it does not only police routing protocols and ARP packets to the switch but also polices traffic through the box 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* **pass-through** command.

This example shows how to display the available protocols to use with ARP 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 arp** command:

```
Router(config)# mls qos protocol arp ?
  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 because a platform that supports hardware forwarding, such as the Catalyst 6500 series switch, reduces the need for redirects.
- Disable unreachables because a platform that supports hardware unreachables, such as the Catalyst 6500 series switch, reduces the need for 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 PFC3B

The PFC3B supports additional hardware-based rate limiters. The PFC3B 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 (PFC3B) 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 PFC3B, 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.

The hardware-based rate limiters available on the PFC3B are as follows:

- Ingress and egress ACL bridged packets
- uRPF check failures
- FIB receive cases
- FIB glean cases
- Layer 3 security features
- ICMP redirects
- ICMP unreachable (ACL drop)
- No-route (FIB miss)
- VACL log
- TTL failure
- MTU failure
- Multicast IPv4
- Multicast IPv6

Ingress-Egress ACL Bridged Packets (Unicast Only)

This rate limiter rate limits packets sent to the PISA 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 PISA. 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 PISA 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 PISA. The uRPF check rate limiters allow you to rate limit the packets per second that are bridged to the PISA CPU when a uRPF check failure occurs.

This example shows how to rate limit the uRPF check failure packets sent to the PISA 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 PISA 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
```

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 PISA). The ICMP unreachable rate limiter allows you to rate limit the packets that are sent to the PISA containing unreachable addresses.

This example shows how to rate limit the packets that are sent to the PISA 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 PISA 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 PISA. This situation occurs when traffic enters a port and contains the destination of a host on a subnet that is locally connected to the

PISA, 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 PISA 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 PISA 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 PISA. For these security features, you need to rate limit the number of these packets being sent to the PISA 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 PISA may be overwhelmed. Rate limiting would be advantageous in this situation.

IPsec and inspection are also done by the PISA 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 PISA 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 PISA sends ICMP-redirect messages to the host to correct its sending path. If this traffic occurs continuously, and is not rate limited, the PISA 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 PISA 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 PISA 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 PISA CPU. This might cause the PISA to be overwhelmed.

This example shows how to rate limit packets failing the MTU failures from being sent to the PISA 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 supervisor engine. IGMP snooping listens to IGMP messages between the hosts and the supervisor engine. You cannot enable the Layer 2 PDU rate limiter if the Catalyst 6500 series 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 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 supervisor engine and not the PISA CPU. You cannot enable the Layer 2 PDU rate limiter if the Catalyst 6500 series 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 supervisor engine. 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 Catalyst 6500 series 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 PFC3B with an IP checksum error or a length inconsistency error, it must be sent to the PISA 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 PISA 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 PISA 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 PISA. For this reason, it may be desirable to rate limit the flow destined to the PISA 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
```

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 33-1 lists the IPv6 rate limiters and the class of traffic that each rate limiter serves.

Table 33-1 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 33-2 shows the DoS protection default configuration for the PFC3B hardware-based rate limiters.

Table 33-2 PFC3B 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 on systems configured with a PFC3B, 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 33-19.

- These rate limiters are supported:
  - 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

The PFC3B supports 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#
```

Understanding How Control Plane Policing Works

The control plane policing (CoPP) feature increases security on the Catalyst 6500 series switch by protecting the PISA from unnecessary or DoS traffic and giving priority to important control plane and management traffic. The PFC3B provides hardware support for CoPP. CoPP works with the PFC3B rate limiters.

Note The Supervisor Engine 2 does not support CoPP.

The PFC3B 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 traffic managed by the PISA is divided into three functional components or *planes*:

- Data plane
- Management plane
- Control plane

The majority of traffic managed by the PISA 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:

- Classes that match multicast are not applied in hardware but are applied in software.
- CPP is not supported in hardware for broadcast packets. The combination of ACLs, traffic storm control, and CPP software protection provides protection against broadcast DoS attacks.
- CoPP does not support ARP policies. ARP policing mechanisms provide protection against ARP storms.
- 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.
- Do not use the **log** keyword in CoPP policy ACLs.
- If you have a large QoS configuration, the system may run out of TCAM space. If this is the case, CoPP may be performed in software.
- 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.
- 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 PFC3B 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.
- CoPP is not enabled in hardware unless MMLS QoS is enabled globally with the **mls qos** command. If the **mls qos** command is not entered, CoPP will only work in software and will not provide any benefit to the hardware.
- 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.
- CoPP is performed on a per-forwarding-engine basis and software CoPP is performed on an aggregate basis.

- CoPP is not supported in hardware for multicast packets. The combination of ACLs, multicast CPU rate limiters, and CoPP software protection provides protection against multicast DoS attacks.
- CoPP does not support ACEs with the **log** keyword.
- CoPP uses hardware QoS TCAM resources. Enter the **show tcam utilization** command to verify the TCAM utilization.
- CoPP does not support MAC ACLs.

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, refer to the "Defining Traffic Classification" section on page 33-22.

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.
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.

Step 4	`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 5	`Router(config)# control-plane` `Router(config-cp)#`	Enters the control plane configuration mode.
Step 6	`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. QoS ACLs supported 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 33-22
- Traffic Classification Guidelines, page 33-23
- Sample Basic ACLs for CoPP Traffic Classification, page 33-24

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 PISA 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 PISA.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 PISA 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 PISA. 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 33-24.

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 PISA) 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
```

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, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS System Message Guide*, Release 12.2ZY at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/system/messages/sysmsg.html

To configure sticky ARP on a Layer 3 interface, perform the following 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* = **ethernet**, **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#
```

CHAPTER 34

Configuring DHCP Snooping

This chapter describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping on Catalyst 6500 series switches.

This chapter consists of the following major sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

Overview of DHCP Snooping

DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding database (also referred to as a DHCP snooping binding table).

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. You can use DHCP snooping to differentiate between untrusted interfaces connected to the end user and trusted interfaces connected to the DHCP server or another switch.

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.

An untrusted message is a message that is received from outside the network or firewall. When you use DHCP snooping in a service-provider environment, an untrusted message is sent from a device that is not in the service-provider network, such as a customer's switch. Messages from unknown devices are untrusted because they can be sources of traffic attacks.

The DHCP snooping binding database contains the MAC address, the IP address, the lease time, the binding type, the VLAN number, and the interface information that corresponds to the local untrusted interfaces of a switch. The database does not contain information regarding hosts interconnected with a trusted interface.

In a service-provider network, a trusted interface is connected to a port on a device in the same network. An untrusted interface is connected to an untrusted interface in the network or to an interface on a device that is not in the network.

When a switch receives a packet on an untrusted interface and the interface belongs to a VLAN in which DHCP snooping is enabled, the switch compares the source MAC address and the DHCP client hardware address. If addresses match (the default), the switch forwards the packet. If the addresses do not match, the switch drops the packet.

The switch drops DHCP packets when any of these situations occur:

- The switch receives a packet from a DHCP server, such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet, from 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.
- The switch receives a DHCPRELEASE or DHCPDECLINE message that contains a MAC address in the DHCP snooping binding table, but 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.

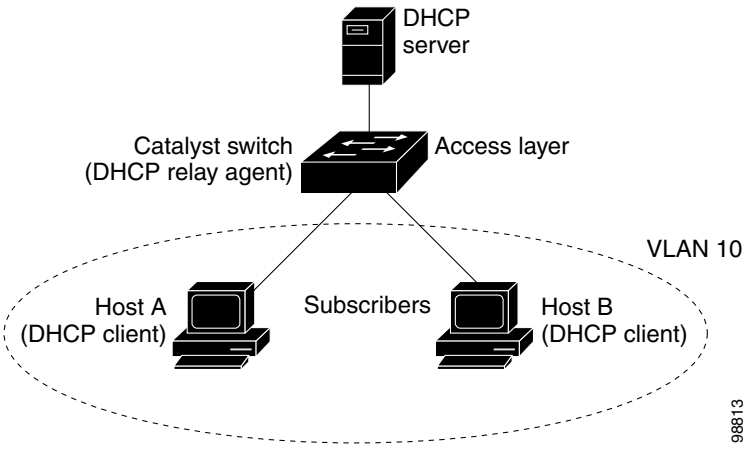
Note With the DHCP option 82 on untrusted port feature enabled, use dynamic ARP inspection on the aggregation switch to protect untrusted input interfaces.

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 34-1 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 34-1 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:

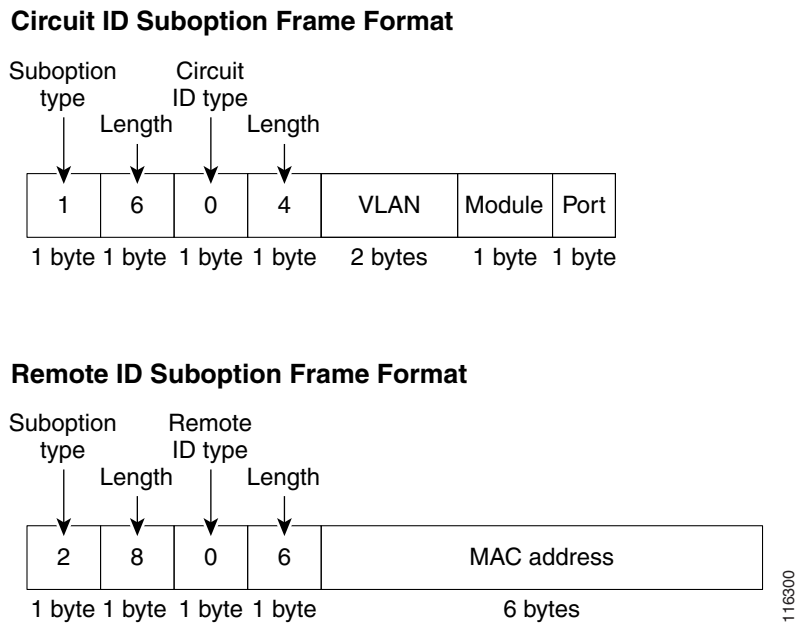
- The host (DHCP client) generates a DHCP request and broadcasts it on the network.
- When the switch receives the DHCP request, it adds the option-82 information in the packet. The option-82 information contains the switch MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption).
- If 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 34-2 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 34-2 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 34-2 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.

Default Configuration for DHCP Snooping

Table 34-1 shows all the default configuration values for each DHCP snooping option.

Table 34-1 Default Configuration Values for DHCP Snooping

Option	Default Value/State
DHCP snooping	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 Configuration Guidelines and Restrictions

When configuring DHCP snooping, follow these guidelines and restrictions:

- 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, refer to "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
- A maximum of 512 bindings are allowed in the DHCP snooping database. If you configure more than 512 DHCP bindings, all bindings will be removed.
- 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.
- To use any DHCP snooping features, you must globally enable DHCP snooping on the switch.
- DHCP snooping is not active until DHCP snooping is enabled on a VLAN.
- 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.

Configuring DHCP Snooping

These sections describe how to configure DHCP snooping:

Enabling DHCP Snooping Globally

Note Enable this feature during a maintenance window, 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.
	`Router(config)# no ip dhcp snooping`	Disables DHCP snooping.
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.
	`Router(config)# no ip dhcp snooping information option`	Disables DHCP option-82 data insertion.
Step 2	`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`	(Optional) Enables untrusted ports to accept incoming DHCP packets with option-82 information. The default setting is disabled.
	`Router(config)# no ip dhcp snooping information option allow-untrusted`	Disables the DHCP option 82 on untrusted port feature.
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 MAC Address Verification

With DHCP snooping MAC address verification enabled, DHCP snooping verifies that the source MAC address in DHCP packets that are received on untrusted ports match the client hardware address in the 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.
	`Router(config)# no ip dhcp snooping verify mac-address`	Disables 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

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.
	`Router(config)# no ip dhcp snooping`	Disables DHCP snooping.
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` `Router(config-if)# no ip dhcp snooping trust`	Configures the interface as trusted. Reverts to the default (untrusted) state.
Step 3	`Router(config-if)# do show ip dhcp snooping | begin pps`	Verifies the configuration.

1. *type* = **ethernet**, **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 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)# **no ip dhcp snooping limit rate**	Disables DHCP packet rate limiting.
Step 4	Router(config-if)# **do show ip dhcp snooping | begin pps**	Verifies the configuration.

1. *type* = **ethernet**, **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* }	(Required) Configures a URL for the database agent (or file) and the related timeout values.
Router(config)# **no ip dhcp snooping database** [**write-delay** | **timeout**]	Clears the configuration.
Router# **show ip dhcp snooping database** [**detail**]	(Optional) Displays the current operating state of the database agent and statistics associated with the transfers.
Router# **clear ip dhcp snooping database statistics**	(Optional) Clears the statistics associated with the database agent.

Command	Purpose
Router# **renew ip dhcp snooping database** [**validation none**] [*url*]	(Optional) 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*	(Optional) Adds bindings to the snooping database.
Router# **no ip dhcp snooping binding** *mac_address* **vlan** *vlan_ID ip_address* **interface** *ifname*	(Optional) Deletes bindings to the snooping database.

When configuring the DHCP snooping database agent, note the following information:

- 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 34-13
- Example 2: Reading Binding Entries from a TFTP File, page 34-14
- Example 3: Adding Information to the DHCP Snooping Database, page 34-16

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 Succeded 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 failed upon 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 set of counters are cleared by the **clear** command. Therefore, 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 Succeded 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 Succeded 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 the following 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 34-2 describes the fields in the **show ip dhcp snooping binding** command output.

Table 34-2 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

CHAPTER 35

Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) on the Catalyst 6500 series switch.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding DAI

These sections describe how DAI helps prevent ARP spoofing attacks:

Understanding ARP

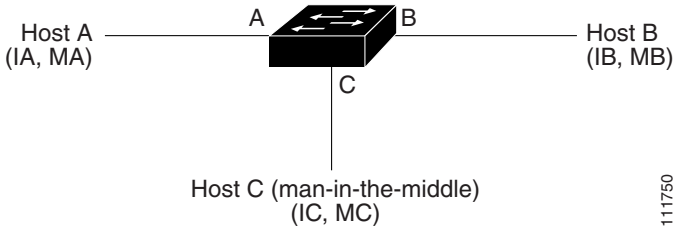
ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, Host B wants to send information to Host A but does not have the MAC address of Host A in its ARP cache. Host B generates a broadcast message for all hosts within the broadcast domain 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 35-1 shows an example of ARP cache poisoning.

Figure 35-1 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 35-8). The switch logs dropped packets (see the "Logging of Dropped Packets" section on page 35-4).

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 35-11).

Interface Trust States and Network Security

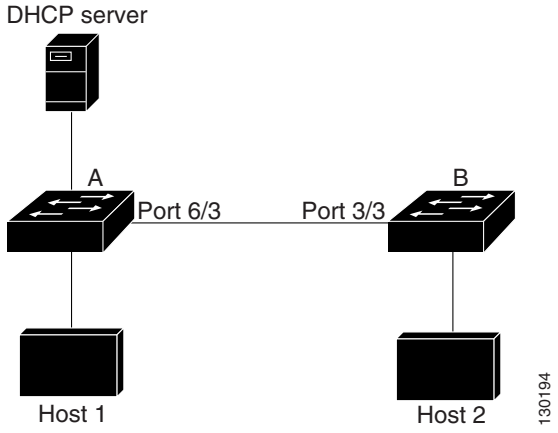
DAI associates a trust state with each interface on the switch. Packets arriving on trusted interfaces bypass all DAI validation checks, and those arriving on untrusted interfaces undergo the DAI validation process.

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 35-2, 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 35-2 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 35-20.

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 35-9.

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 35-12.

Default DAI Configuration

Table 35-1 shows the default DAI configuration.

Table 35-1 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 34, "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. Otherwise, the physical port remains suspended in the port channel. A port channel inherits its trust state from the first physical port that joins the channel. Consequently, the trust state of the first physical port need not match the trust state of the channel.

  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 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 (disabled by default).
	`Router(config)# no ip arp inspection vlan {vlan_ID | vlan_range}`	Disables 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 does not check ARP packets that it receives from the other switch on the trusted interface. It simply forwards the packets.

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 35-12.

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 (default: untrusted).
	`Router(config)# no ip arp inspection trust`	Configures the connection between switches as untrusted.
Step 4	`Router(config-if)# do show ip arp inspection interfaces`	Verify the DAI configuration.

1. *type* = **ethernet**, **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 *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, 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.
	`Router(config-if)# no ip arp inspection limit`	Clears the ARP packet rate-limiting configuration.
Step 4	`Router(config-if)# do show ip arp inspection interfaces`	Verifies the configuration.

1. *type* = **ethernet**, **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 35-5.

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 (disabled by default).
	`Router(config-if)# no errdisable recovery cause arp-inspection`	Disables 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 (default is none).
	`Router(config)# no ip arp inspection validate {[dst-mac] [ip] [src-mac]}`	Disables 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).
	`Router(config)# no ip arp inspection log-buffer entries`	Reverts to the default buffer size (32).
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.
	`Router(config)# no ip arp inspection log-buffer logs`	Reverts to the default system message configuration.
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 35-2.

Table 35-2 Commands for Displaying DAI Information

Command	Description
show arp access-list [*acl_name*]	Displays detailed information about ARP ACLs.
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 35-3.

Table 35-3 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 35-4:

Table 35-4 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 35-16
- Sample Two: One Switch Supports DAI, page 35-20

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 35-2 on page 35-3. 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 34, "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 35-2 on page 35-3 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#
```

CHAPTER 36

Configuring Traffic Storm Control

This chapter describes how to configure the traffic storm control feature on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

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, 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. 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.

Figure 36-1 shows the broadcast traffic patterns on a LAN interface over a given 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 36-1 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 on the Catalyst 6500 series switches 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 a threshold is reached, 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 WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, WS-X6148A-GE-TX, and WS-X6148V-GE-TX switching modules do not support traffic storm control, nor do versions of these modules equipped with inline power (Power over Ethernet, or PoE) daughtercards.
- 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.

Enabling Traffic Storm Control

To enable traffic storm control, 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.
	`Router(config-if)# no storm-control broadcast level`	Disables broadcast traffic storm control on the interface.

	Command	Purpose
Step 3	`Router(config-if)# storm-control multicast level level[.level]` **Note** The **storm-control multicast** command is supported only on 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.
	`Router(config-if)# no storm-control multicast level`	Disables multicast traffic storm control 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 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.
	`Router(config-if)# no storm-control unicast level`	Disables unicast traffic storm control on the interface.
Step 5	`Router(config-if)# end`	Exits configuration mode.
Step 6	`Router# show running-config interface`	Verifies the configuration.

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

When configuring the traffic storm control level, note the following information:

- You can configure traffic storm control on an EtherChannel (a port channel interface).
- 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.

Note On these modules, a level value of 0.33 percent or less suppresses all traffic:

—WS-X6704-10GE
—WS-X6748-SFP
—WS-X6724-SFP
—WS-X6748-GE-TX

Note On module WS-X6716-10G-3C / 3CXL Oversubscription Mode, a level value of 0.29 percent or less suppresses all traffic.

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.

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 70.5 percent:

```
Router# configure terminal
```

```
Router(config)# interface gigabitethernet 3/16
Router(config-if)# storm-control multicast level 70.5
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 70.00
Router# storm-control multicast level 70.00
Router# spanning-tree portfast edge
Router# end

Router# configure terminal
Router(config)# interface gigabitethernet 4/10
Router(config-if)# storm-control unicast level 20
Router(config-if)# end

Router# show interfaces gig4/10 counters storm-control

Port      UcastSupp %     McastSupp %     BcastSupp %  TotalSuppDiscards
Gi4/10      20.00           20.00           20.00              0

Router#
```

Displaying Traffic Storm Control Settings

To display traffic storm control information, use the commands described in Table 36-1.

Table 36-1 Commands for Displaying Traffic Storm Control Status and Configuration

Command	Purpose
Router# **show interfaces** [{*type*[1] *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** [{*type*[1] *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* = **ethernet**, **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 use the **storm-control** keyword to display the discard count.

CHAPTER 37

Configuring Unknown Unicast and Multicast Flood Blocking

This chapter describes how to configure the unknown unicast flood blocking (UUFB) and unknown multicast flood blocking (UMFB) features on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter contains these sections:

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 and UMFB 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.

Note Entering the **switchport block multicast** command on nonreceiver (router) ports of the VLAN could disrupt routing protocols. This command could also disrupt ARP functionality and other protocols, such as Network Time Protocol (NTP), that make use of local subnetwork multicast control groups in the 224.0.0.0/24 range.

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
```

CHAPTER 38

Configuring PFC QoS

This chapter describes how to configure quality of service (QoS) as implemented on the Policy Feature Card 3B (PFC3B) on the Supervisor Engine 32 PISA.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- For information about QoS and MPLS, see Chapter 39, "Configuring MPLS QoS."
- QoS on the Catalyst 6500 series switches (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.
- To configure NBAR on the PISA, refer to this publication:

  http://www.cisco.com/en/US/docs/ios/12_4t/qos/configuration/guide/qsnbar1.html
- QoS features implemented on port ASICs are supported on interfaces where you configure PISA-accelerated features.
- QoS features implemented on the PFC are not supported on interfaces where you configure PISA-accelerated features.
- To avoid unexpected application of QoS to the PISA EtherChannel, do not configure QoS on the WS-S32-10GE-PISA or WS-S32-GE-PISA ports (see the "Supervisor Engine 32 PISA Ports" section on page 4-2).

This chapter contains these sections:

Understanding How PFC QoS Works

The term "PFC QoS" refers to QoS on the Catalyst 6500 series switch. PFC QoS is implemented on various switch components in addition to the PFC3B. These sections describe how PFC QoS works:

- Overview, page 38-2
- Component Overview, page 38-5
- Understanding Classification and Marking, page 38-14
- Understanding Port-Based Queue Types, page 38-19

Note The PFC3B does not provide QoS for FlexWAN module ports. Refer to 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

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.

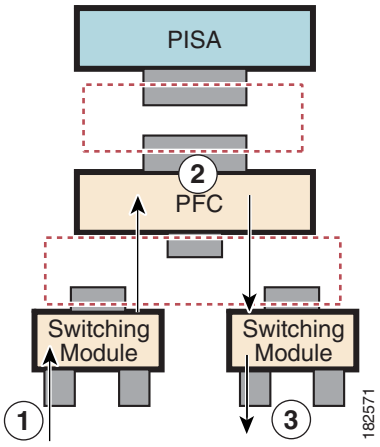
QoS makes network performance more predictable and bandwidth utilization more effective. QoS selects (classifies) network traffic, uses or assigns QoS labels to indicate priority, makes the packets comply with the configured resource usage limits (polices the traffic and marks the traffic), and provides congestion avoidance where resource contention exists.

PFC QoS classification, policing, marking, and congestion avoidance is implemented in hardware on the PFC3B and in LAN switching module port Application Specific Integrated Circuits (ASICs).

Note Catalyst 6500 series switches do 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, Catalyst 6500 series switches do not support MQC-configured queuing.

Figure 38-1 shows an overview of QoS processing in a Catalyst 6500 series switch.

Figure 38-1 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 received 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, QoS classifies the traffic on the basis of its Layer 2 CoS value and assigns it to an ingress queue to provide congestion avoidance.
2. PFC QoS features (not supported with PISA-accelerated features):
   - Internal DSCP—On the PFC3B, 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 congestion avoidance.

These figures provide more detail about the relationship between QoS and the switch components:

- Figure 38-2, Traffic Flow and PFC QoS Features with PFC3B
- Figure 38-3, PFC QoS Features and Component Overview

Figure 38-2 shows traffic flow and PFC QoS features with a PFC3B.

Figure 38-2 Traffic Flow and PFC QoS Features with PFC3B

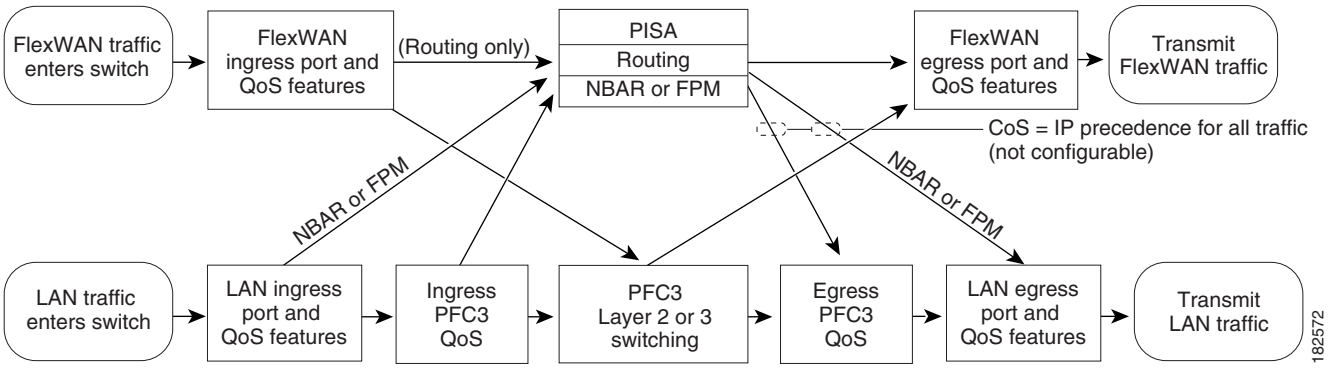
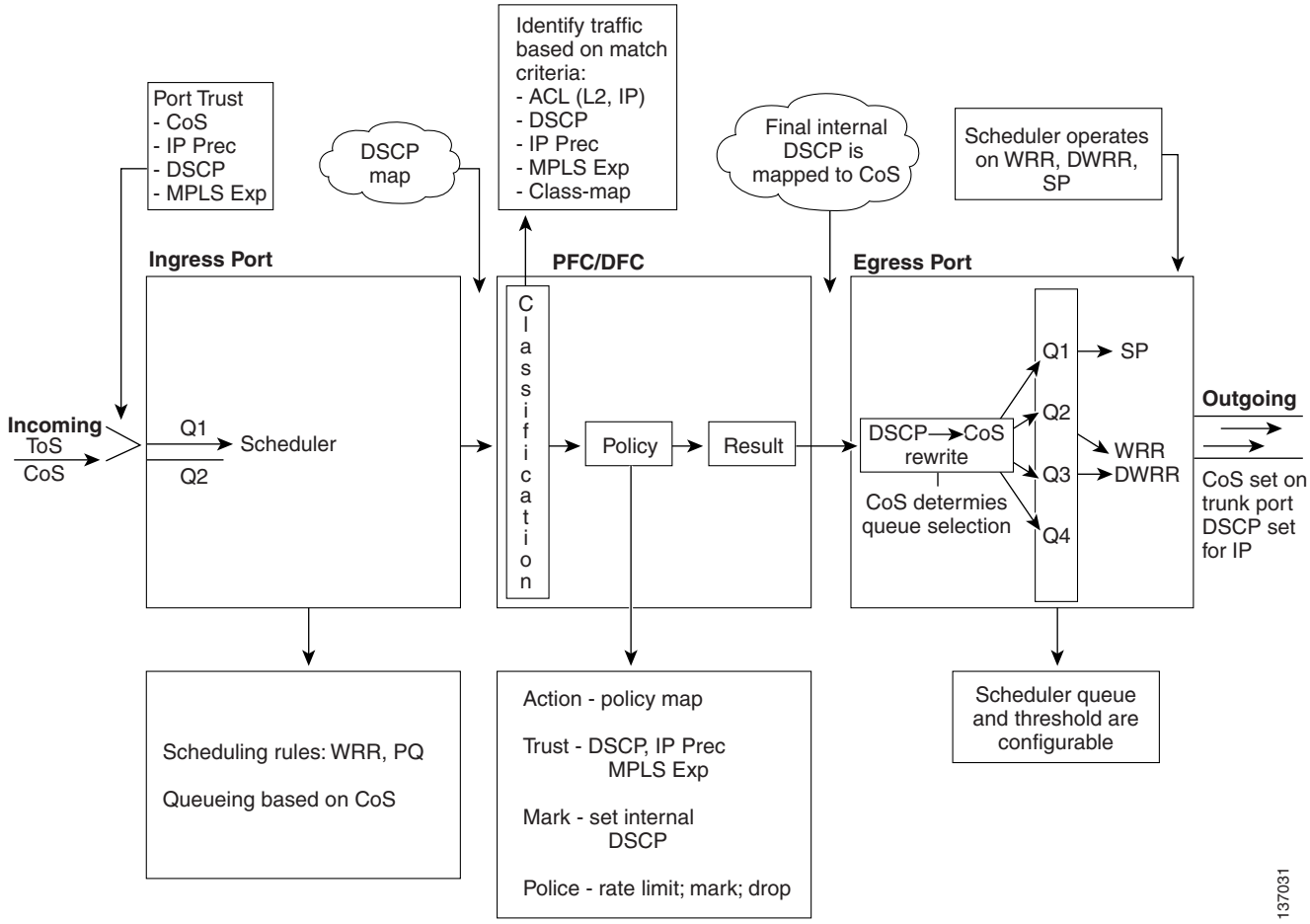


Figure 38-2 shows how traffic flows through the PFC QoS features with PFC3B:

- Traffic can enter on any type of port and exit on any type of port.
- 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 PFC3B or routed in software by the PISA.
  - 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 (not supported with PISA-accelerated features).
  - Ingress LAN-port traffic can be Layer-2 or Layer-3 switched by the PFC3B or routed in software by the PISA.
  - Egress PFC QoS and egress LAN-port QoS can be applied to LAN-port egress traffic (not supported with PISA-accelerated features).

Figure 38-3 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 38-5
- PFC QoS Features, page 38-7
- PFC QoS Egress Port Features, page 38-11

Ingress LAN Port PFC QoS Features

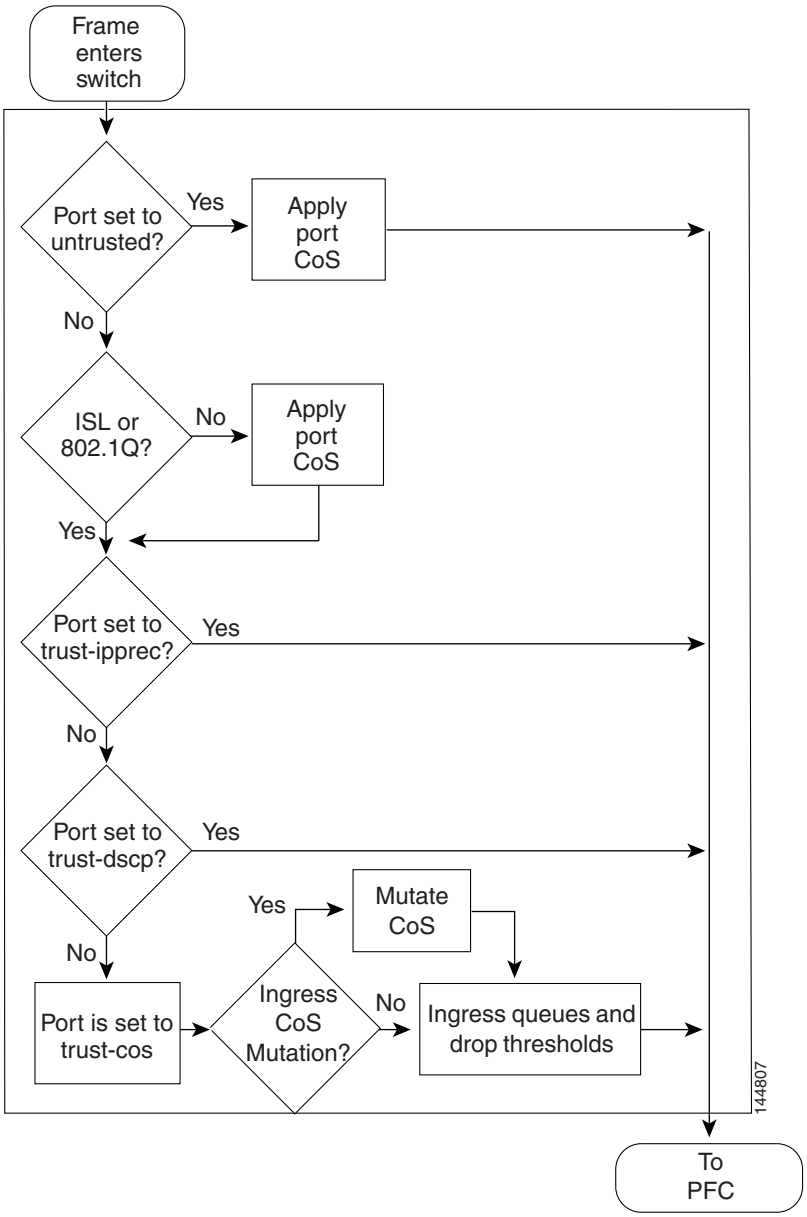
These sections provide an overview of the ingress port QoS features:

- Flowchart of Ingress LAN Port PFC QoS Features, page 38-6
- Port Trust, page 38-7
- Ingress Congestion Avoidance, page 38-7

Flowchart of Ingress LAN Port PFC QoS Features

Figure 38-4 shows how traffic flows through the ingress LAN port PFC QoS features.

Figure 38-4 Ingress LAN Port PFC QoS Features



Note Ingress CoS mutation is supported only on 802.1Q tunnel 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. See the "Ingress Classification and Marking at Trust CoS LAN Ports" section on page 38-15 for more information about ingress congestion avoidance.

PFC QoS Features

These sections describe PFC3Bs as they relate to QoS:

Supported Policy Feature Cards

The policy feature card (PFC3B) is a daughter card that resides on the supervisor engine. The PFC3B provides QoS in addition to other functionality.

PFC QoS Feature List and Flowchart

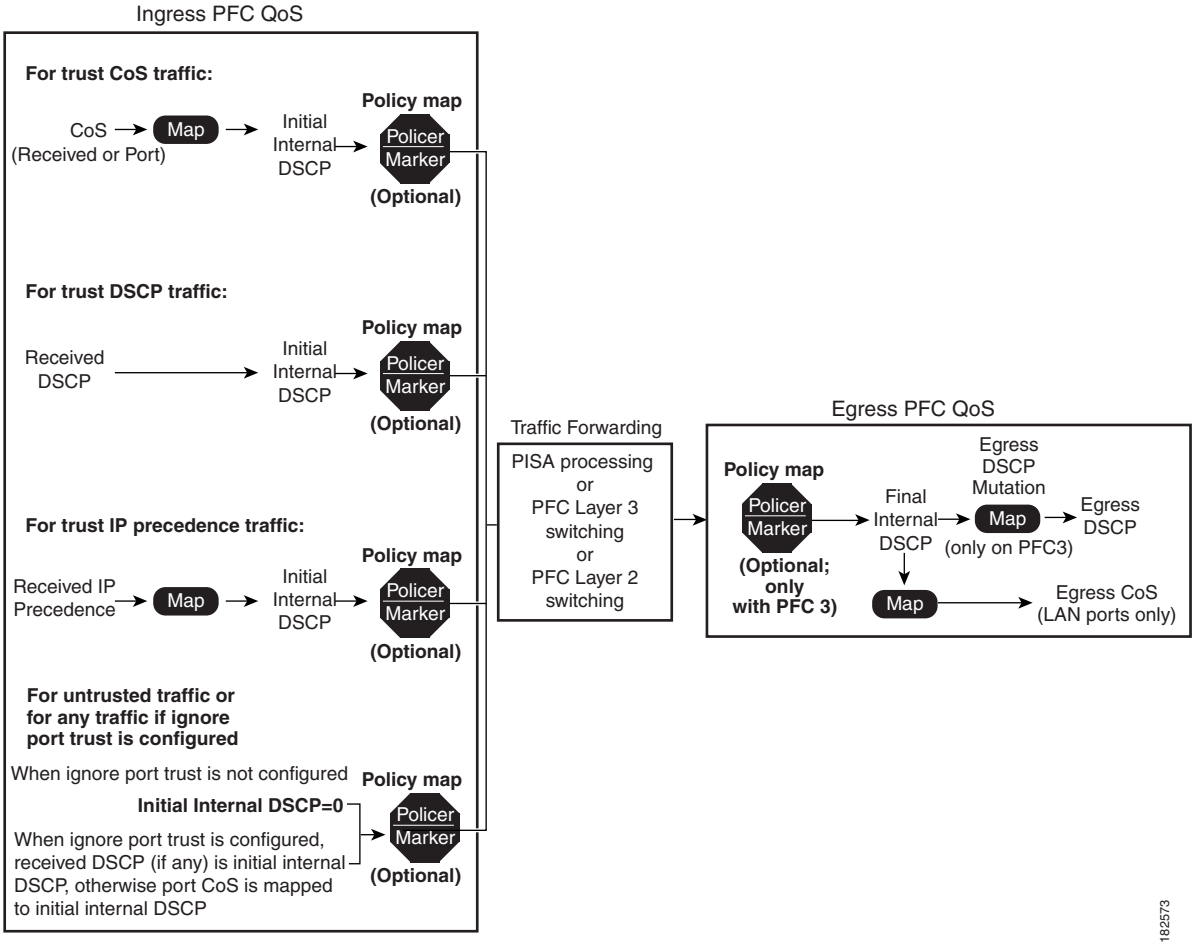
Table 38-1 lists the QoS features supported on the PFC3B. These PFC QoS features are not supported on interfaces where you configure PISA-accelerated features.

Table 38-1 QoS Features Supported on the PFC3B

Feature	PFC3B
Flow granularity	Source Destination
QoS ACLs	IP, MAC
DSCP transparency **Note** Enabling DSCP transparency disables egress ToS rewrite.	Optional
Egress ToS rewrite	Optional
Policing:	
Ingress aggregate policers	Yes
Egress aggregate policers	Yes
Number of aggregate policers	1022
Microflow policers	64 rates
Number of flows per Microflow policer	110,000
Unit of measure for policer statistics	Bytes
Basis of policer operation	Layer 2 length

Figure 38-5 shows how traffic flows through the QoS features on the PFC3B.

Figure 38-5 QoS Features on the PFC3B



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 PFC3B, 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 PFC3B 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.

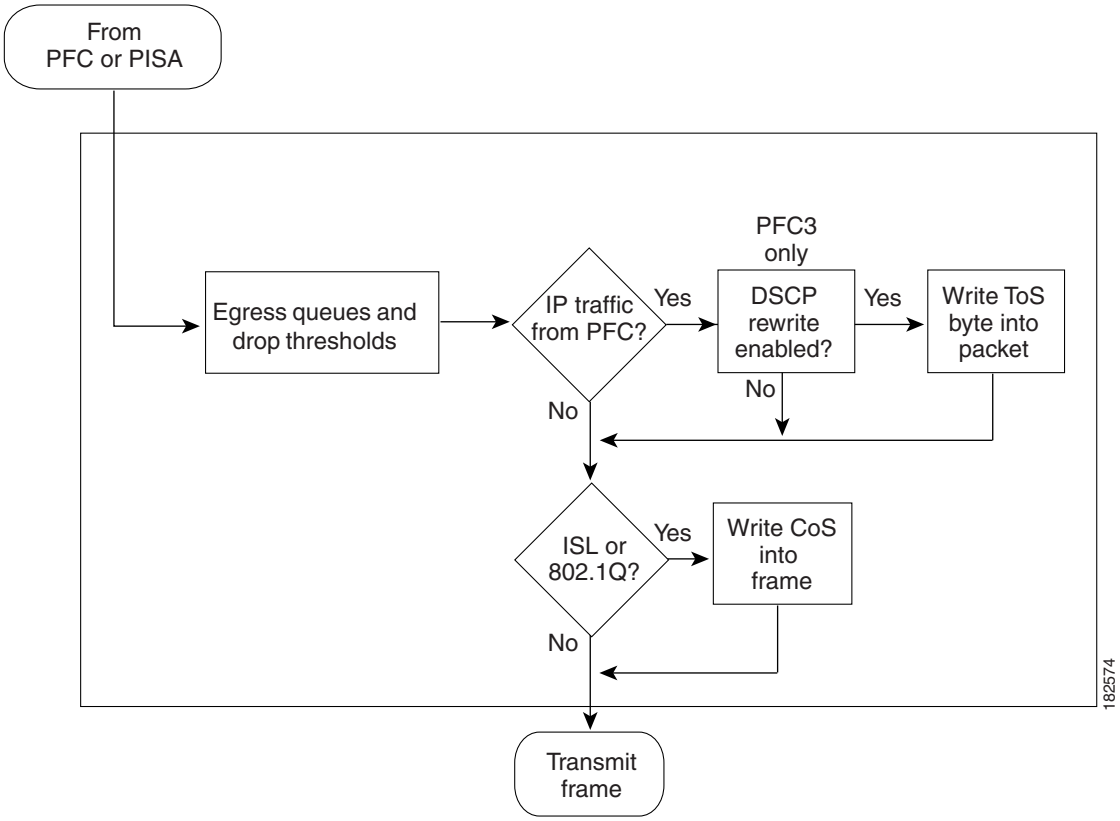
PFC QoS Egress Port Features

These sections describe PFC QoS egress port features:

Flowchart of PFC QoS Egress LAN Port Features

Figure 38-6 shows how traffic flows through the QoS features on egress LAN ports.

Figure 38-6 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.

Egress DSCP Mutation with a PFC3B

With a PFC3B, 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 that PFC QoS supports.

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 38-44).

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.
- With a PFC3B, you cannot apply microflow policing to ARP traffic.
- You cannot set a trust state in an output policy.
- Egress PFC QoS is not supported on interfaces where you configure PISA-accelerated features.

Egress ACL Support for Remarked DSCP

Note

- Egress ACL support for remarked DSCP is also known as packet recirculation.
- Egress ACL support for remarked DSCP is not supported on interfaces where you configure PISA-accelerated features.

Egress ACL support for remarked DSCP 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 PFC3B 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 PFC3B processes each QoS-filtered IP packet twice: once to apply ingress PFC QoS and once to apply egress PFC QoS.

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 on the Catalyst 6500 series switches:

- Classification and Marking at Trusted and Untrusted Ingress Ports, page 38-14
- Classification and Marking on the PFC3B Using Service Policies and Policy Maps, page 38-15
- Classification and Marking on the PISA, page 38-16

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

In all releases, 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.

Ingress LAN port classification, marking, and congestion avoidance 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 38-14
- Ingress Classification and Marking at Trusted Ports, page 38-14

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 38-15
- Ingress Classification and Marking at Trust IP Precedence Ports, page 38-15
- Ingress Classification and Marking at Trust DSCP Ports, page 38-15

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 38-19 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. The PFC3B 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.

Ingress port queues and thresholds use only Layer 2 CoS, and PFC QoS does not implement ingress port congestion avoidance on ports configured to trust DSCP.

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 PFC3B Using Service Policies and Policy Maps

Note PFC QoS classification and marking with service policies is not supported on interfaces where you configure PISA-accelerated features.

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

You can attach one policy map to each Layer 3 interface 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 38-44 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.

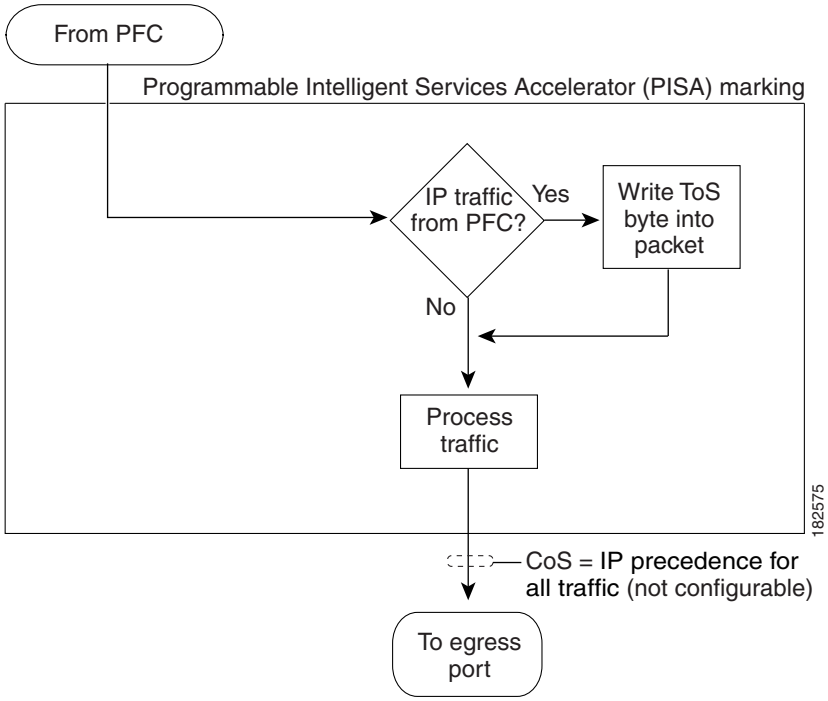
Note A trust CoS policy map cannot restore received CoS in traffic from untrusted ports. Traffic from untrusted ports always has the port CoS value.

- Aggregate and microflow policers—PFC QoS can use policers to either mark or drop both conforming and nonconforming traffic.

Classification and Marking on the PISA

Some traffic is routed in software on the PISA. PFC QoS sends IP traffic to the PISA with the final internal DSCP values. CoS is equal to IP precedence in all traffic sent from the PISA to egress ports.

Figure 38-7 Marking with PFC3B and PISA



Note Traffic that is Layer 3 switched on the PFC3B does not go through the PISA and retains the CoS value assigned by the PFC3B.

Policers

These sections describe policers:

- Overview of Policers, page 38-17
- Aggregate Policers, page 38-18
- Microflow Policers, page 38-18

Note Interfaces on which you configure PISA-accelerated features do not support policing in hardware on the PFC.

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 PFC3B supports both ingress and egress PFC QoS, which includes ingress and egress policing. Traffic shaping is supported on some WAN modules. For more information about traffic shaping on the FlexWAN module, refer to the FlexWAN QoS documentation at this URL:

http://www.cisco.com/en/US/docs/routers/7600/install_config/flexwan_config/flexqos.html

Note Policers can act on ingress traffic per-port or per-VLAN. The policers can act on egress traffic 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.

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. With a PFC3B, you can configure MAC ACLs to filter IPX traffic.
- By default, microflow policers only affect traffic routed by the PISA. 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.

With a PFC3B, 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 Catalyst 6500 series switch LAN modules:

- Ingress and Egress Buffers and Layer 2 CoS-Based Queues, page 38-20
- Ingress Queue Types, page 38-21
- Egress Queue Types, page 38-22
- Module to Queue Type Mappings, page 38-23

Ingress and Egress Buffers and Layer 2 CoS-Based Queues

The Ethernet LAN module 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 Catalyst 6500 series switch LAN modules support the following types of queues:

- Standard queues
- Strict-priority queues

The Catalyst 6500 series switch LAN modules 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 Catalyst 6500 series switch LAN modules provides 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.

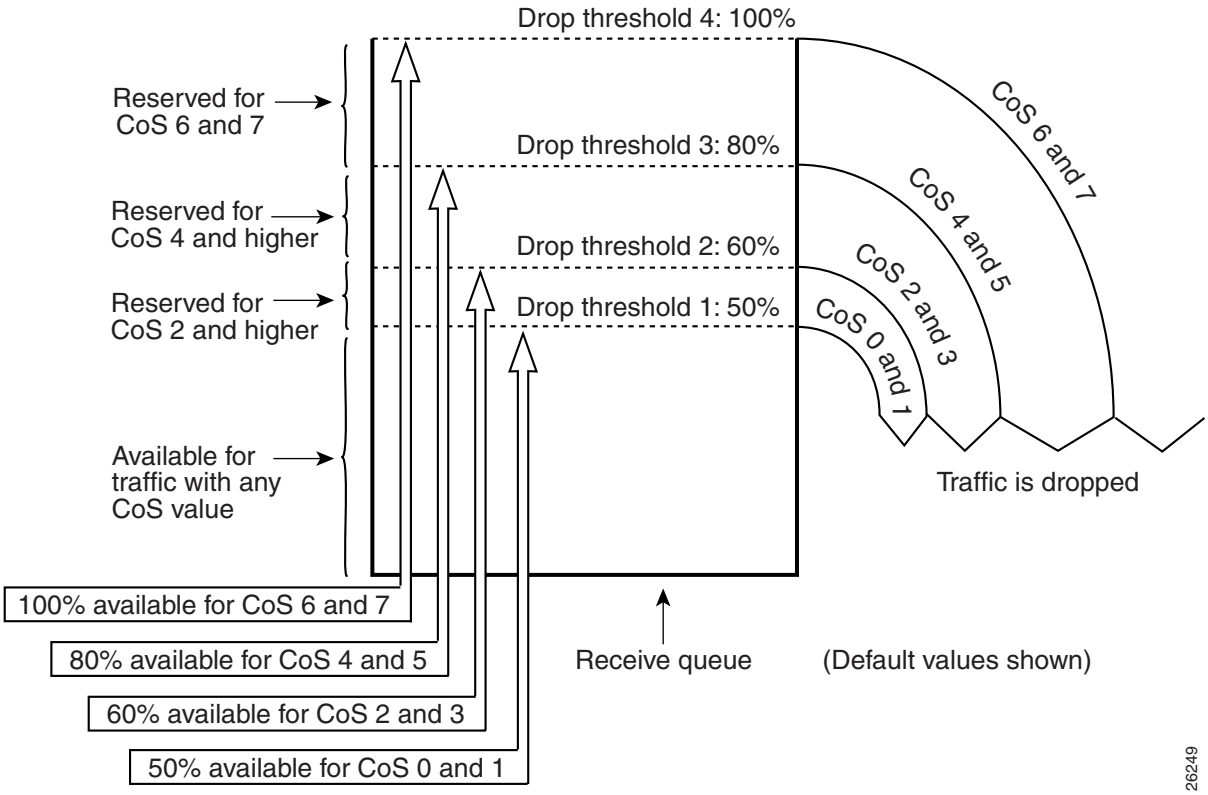
  For example, if CoS 2 is assigned to queue 1, threshold 2, and the threshold 2 levels are 40 percent (low) and 80 percent (high), then frames with CoS 2 will not be dropped until queue 1 is at least 40 percent full. As the queue depth approaches 80 percent, frames with CoS 2 have an increasingly higher 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.

The combination of multiple queues and the scheduling algorithms associated with each queue allows the switch to provide congestion avoidance.

Figure 38-8 illustrates the drop thresholds for a **1q4t** ingress LAN port. Drop thresholds in other configurations function similarly.

Figure 38-8 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.
- **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 non-configurable (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

- **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 38-2—Supervisor Engine Module QoS Queue Structures
- Table 38-3—Ethernet and Fast Ethernet Module Queue Structures
- Table 38-4—Gigabit and 10/100/1000 Ethernet Modules
- Table 38-5—10 Gigabit Ethernet Modules

Table 38-2 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
WS-S32-10GE-PISA	2q8t	WRR	1p3q8t	DWRR SRR			
10 Gigabit Ethernet ports					193 MB	105 MB	88 MB
Gigabit Ethernet port					17.7 MB	9.6 MB	8.1 MB
WS-S32-GE-PISA					17.7 MB	9.6 MB	8.1 MB

Table 38-3 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 38-3 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-X6148A-RJ45	1p1q4t	—	1p3q8t	DWRR	5.3 MB	60KB	5.3 MB
WS-X6148A-45AF							
WS-X6148X2-RJ-45	1p1q0t	—	1p3q1t	DWRR	1,116 KB	28 KB	1,088 KB
WS-X6148X2-45AF							
WS-X6196-RJ-21							
WS-X6196-21AF							
WS-X6024-10FL-MT	1q4t	—	2q2t	WRR	64 KB	8 KB	56 KB

Table 38-4 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-X6548-GE-TX	1q2t	—	1p2q2t	WRR	1.4 MB	185 KB	1.2 MB
WS-X6548V-GE-TX							
WS-X6548-GE-45AF							

Table 38-4 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-X6516-GBIC	1p1q4t	—	1p2q2t	WRR	512 KB	73 KB	439 KB
WS-X6516A-GBIC					1 MB	135 KB	946 KB
WS-X6516-GE-TX					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							
WS-X6148A-GE-TX			1p3q8t	DWRR	5.5 MB	120 KB	5.4 MB
WS-X6148A-GE-45AF							

Table 38-5 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-X6502-10GE	1p1q8t	—	1p2q1t	DWRR	64.2 MB	256 KB	64 MB
WS-X6501-10GEX4							

PFC QoS Default Configuration

These sections describe the PFC QoS default configuration:

- PFC QoS Global Settings, page 38-26
- Default Values with PFC QoS Enabled, page 38-27
- Default Values with PFC QoS Disabled, page 38-38

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
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 38-27
- Transmit-Queue Limit s, page 38-27
- Bandwidth Allocation Ratios, page 38-28
- Default Drop-Threshold Percentages and CoS Value Mappings, page 38-28

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%
8q8t	Lowest priority: 80%
	Intermediate queues: 0%
	Highest priority: 20%

Transmit-Queue Limit s

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%

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
8q8t	90:0:0:0:0:0:0:10
1p3q8t	22:33:45
1p7q8t	22:33:45:0:0:0:0
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 38-29
- 1q4t Receive Queues, page 38-29
- 1p1q4t Receive Queues, page 38-30
- 1p1q0t Receive Queues, page 38-30
- 1p1q8t Receive Queues, page 38-31
- 1q8t Receive Queues, page 38-32
- 2q8t Receive Queues, page 38-33
- 8q8t Receive Queues, page 38-34
- 2q2t Transmit Queues, page 38-34
- 1p2q2t Transmit Queues, page 38-35
- 1p3q8t Transmit Queues, page 38-36
- 1p7q8t Transmit Queues, page 38-37
- 1p3q1t Transmit Queues, page 38-38
- 1p2q1t Transmit Queues, page 38-38

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

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
		Tail-drop	Not supported
		WRED-drop	40% low, 70% high
	Threshold 2	CoS	6 and 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)

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
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

- QoS features implemented on port ASICs are supported on interfaces where you configure PISA-accelerated features.
- QoS features implemented on the PFC are not supported on interfaces where you configure PISA-accelerated features.
- 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 random-detect**
  - **rcv-queue queue-limit**
  - **wrr-queue queue-limit**
  - **wrr-queue bandwidth** (except Gigabit Ethernet LAN ports)
  - **priority-queue cos-map**
  - **rcv-queue cos-map**
  - **wrr-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**

PFC3B Guidelines

- The PFC3B supports 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 PFC3B supports IPv6 named extended ACLs and named standard ACLs.
- The PFC3B 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 PFC3B does not provide hardware-assistance for these features:
  - Cisco IOS reflexive ACLs
  - TCP intercept
  - Context-Based Access Control (CBAC)
  - Network Address Translation (NAT)
- You cannot apply microflow policing to ARP traffic.
- The PFC3B does not apply egress policing to traffic that is being bridged to the PISA.
- The PFC3B does not apply egress policing or egress DSCP mutation to multicast traffic from the PISA.
- With a PFC3B, PFC QoS does not rewrite the ToS byte in bridged multicast traffic.
- The PFC3B supports up to 1023 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 the **match any** class map command.
- PFC QoS supports class maps that contain a *single* **match** command.
- 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)

CIR and PIR Rate Value Range	Granularity
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 Gps)	16777216 (16 Mb)
1073741825 to 2147483648 (2 Gps)	33554432 (32 Mb)
2147483649 to 4294967296 (4 Gps)	67108864 (64 Mb)
4294967296 to 8589934592 (8 Gps)	134217728 (128 Mb)
8589934592 to 10000000000 (10 Gps)	268435456 (256 Mb)

Within each range, PFC QoS programs the PFC3B 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 PFC3B 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 on the Catalyst 6500 series switches:

- Enabling PFC QoS Globally, page 38-45
- Enabling Ignore Port Trust, page 38-46
- Configuring DSCP Transparency, page 38-46
- Enabling Queueing-Only Mode, page 38-47

Note

- PFC QoS processes both unicast and multicast traffic.
- QoS features implemented on port ASICs are supported on interfaces where you configure PISA-accelerated features.
- QoS features implemented on the PFC are not supported on interfaces where you configure PISA-accelerated features.

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.
	`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 [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.
	`Router(config)# no mls qos marking ignore port-trust`	Disables ignore port trust globally on the switch (default).
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, the PFC3B supports 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.
	`Router(config)# mls qos rewrite ip dscp`	Enables 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.
	`Router(config)# no mls qos queueing-only`	Disables PFC QoS globally on the switch. **Note** You cannot disable queueing-only mode separately.
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.
	`Router(config-if)# no mls qos bridged`	Disables microflow policing of bridged traffic.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos`	Verifies the configuration.

1. *type* = **ethernet**, **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.

	Command	Purpose
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.
	`Router(config-if)# no mls qos vlan-based`	Disables VLAN-based PFC QoS.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos`	Verifies the configuration.

1. *type* = **ethernet**, **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.
	`Router(config-if)# no platform ip features sequential [access-group IP_acl_name_or_number]`	Disables egress ACL support for remarked DSCP on the ingress interface.
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* = **ethernet**, **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.
`Router(config)# no mls qos aggregate-policer` *policer_name*	Deletes 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:

- You can apply aggregate policers to IPv6 traffic.
- With a PFC3B, policing uses the Layer 2 frame size.
- See the "PFC QoS Configuration Guidelines and Restrictions" section on page 38-39 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—10 gigabits per second, entered as 10000000000
- 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:
  - The minimum token bucket size is 1 kilobyte, entered as 1000 (the *maximum_burst_bytes* parameter must be set larger than the *normal_burst_bytes* parameter).
  - The maximum token bucket size is 32 megabytes, entered as 32000000.
  - To sustain a specific rate, set the token bucket size to be at least the rate value divided by 4000 because tokens are removed from the bucket every 1/4000th of a second (0.25 ms).
  - Because the token bucket must be large enough to hold at least one frame, set the parameter 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 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—10 gigabits per second, entered as 10000000000
- (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:

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
MAC Layer	No	No	Yes
ARP	No	No	Yes

  - The PFC3B supports IPv6 named extended ACLs and named standard ACLs.
  - The PFC3B supports ARP ACLs.

    Note —The PFC3B does not apply IP ACLs to ARP traffic.

    —With a PFC3B, you cannot apply microflow policing to ARP traffic.

  - The PFC3B does not support IPX ACLs. With the PFC3B, you can configure MAC ACLs to filter IPX traffic.
  - PFC QoS supports time-based Cisco IOS ACLs.
  - Except for MAC ACLs and ARP ACLs, refer to 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 30, "Configuring Network Security," for additional information about ACLs on the Catalyst 6500 series switches.
- **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:

- Configuring Protocol-Independent MAC ACL Filtering, page 38-54
- Enabling VLAN-Based MAC QoS Filtering, page 38-55
- Configuring MAC ACLs, page 38-56

Note You can use MAC ACLs with VLAN ACLs (VACLs). For more information, see Chapter 32, "Configuring VLAN ACLs."

Configuring 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.
	`Router(config-if)# no mac packet-classify`	Disables protocol-independent MAC ACL filtering on the interface.

1. *type* = **ethernet**, **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 PFC3B.
- Protocol-independent MAC ACL filtering supports microflow policing when the permitted traffic is routed in software by the PISA.

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 (IPX filtering with a MAC ACL is supported only with a PFC3B).

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 access-list extended list_name`	Configures a MAC ACL.
	`Router(config)# no mac access-list extended list_name`	Deletes a MAC ACL.
Step 2	`Router(config-ext-macl)# {permit | deny} {src_mac_mask | any} {dest_mac_mask | any} [{protocol_keyword | {ethertype_number ethertype_mask}} [vlan vlan_ID] [cos cos_value]]`	Configures an access control entry (ACE) in a MAC ACL.
	`Router(config-ext-macl)# no {permit | deny} {src_mac_mask | any} {dest_mac_mask | any} [{protocol_keyword | {ethertype_number ethertype_mask}} [vlan vlan_ID] [cos cos_value]]`	Deletes an ACE from a MAC ACL.

When configuring an entry in a MAC-Layer ACL, note the following information:

- The PFC3B supports the **ipx-arpa** and **ipx-non-arpa** 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 4-byte values in dotted hexadecimal format. For example, 0030.9629.9f84.
- You can enter MAC address masks as three 4-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 PFC3B does not apply IP ACLs to ARP traffic.
- With a PFC3B, 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.
	`Router(config)# no arp access-list list_name`	Deletes an ARP ACL.
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.
	`Router(config-arp-nacl)# no {permit | deny} {ip {any | host sender_ip | sender_ip sender_ip_wildcardmask} mac any`	Deletes an ACE from an ARP ACL.

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 PFC3B. 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 38-58
- Class Map Filtering Guidelines and Restrictions, page 38-58
- Configuring Filtering in a Class Map, page 38-59

Creating a Class Map

To create a class map, perform this task:

Command	Purpose
`Router(config)# class-map class_name`	Creates a class map.
`Router(config)# no class-map class_name`	Deletes a class map.

Class Map Filtering Guidelines and Restrictions

When configuring class map filtering, follow these guidelines and restrictions:

- PFC QoS supports multiple match criteria in class maps configured with the **match-any** keywords.
- When multiple **match access-group** ACLs are included in a **match-any** class map, and one ACL contains a **deny** entry, all match criteria after the **deny** 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** 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** entry into different class maps.

- PFC QoS supports class maps that contain a single **match** command.
- The PFC3B 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.
- Catalyst 6500 series switches do 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 38-54
- Configuring ARP ACLs for QoS Filtering, page 38-57

Other ACLs are not documented in this publication. See the references under **access-list** in the "PFC QoS Policy Configuration Overview" section on page 38-53.

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)# no match access-group name` *acl_index_or_name*	Clears the ACL configuration from the class map.

Command	Purpose
`Router (config-cmap)# match protocol ipv6`	(Optional—for IPv6 traffic) Configures the class map to filter IPv6 traffic.
`Router (config-cmap)# no match protocol ipv6`	Clears IPv6 filtering.
`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)# no match precedence ipp_value1 [ipp_value2 [ipp_valueN]]`	Clears configured IP precedence values from the class map.
`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)# no match dscp dscp_value1 [dscp_value2 [dscp_valueN]]`	Clears configured DSCP values from the class map.
`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)# no match ip precedence ipp_value1 [ipp_value2 [ipp_valueN]]`	Clears configured IP precedence values from the class map.
`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.
`Router (config-cmap)# no match ip dscp dscp_value1 [dscp_value2 [dscp_valueN]]`	Clears configured DSCP values from the class map.

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 38-61
- Policy Map Class Configuration Guidelines and Restrictions, page 38-61
- Creating a Policy Map Class and Configuring Filtering, page 38-62
- Configuring Policy Map Class Actions, page 38-62

Creating a Policy Map

To create a policy map, perform this task:

Command	Purpose
`Router(config)# policy-map policy_name`	Creates a policy map.
`Router(config)# no policy-map policy_name`	Deletes the 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 class maps that contain a single **match** command.
`Router(config-pmap)# no class class_name`	Clears use of the class map.

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 38-63
- Configuring the Policy Map Class Trust State, page 38-63
- Configuring Policy Map Class Policing, page 38-63

Configuring Policy Map Class Marking

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.
`Router(config-pmap-c)# no set {dscp dscp_value | precedence ip_precedence_value}`	Clears the marking configuration.

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.
`Router(config-pmap-c)# no trust`	Reverts to the default policy-map class trust state (untrusted).

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 38-64
- Configuring a Per-Interface Policer, page 38-64

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.
`Router(config-pmap-c)# no police aggregate aggregate_name`	Clears use of the 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.
`Router(config-pmap-c)# no 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}]`	Deletes the per-interface policer from the policy-map class.

When configuring a per-interface policer, note the following information:

- With a PFC3B, 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.
- With a PFC3B, policing uses the Layer 2 frame size.
- See the "PFC QoS Configuration Guidelines and Restrictions" section on page 38-39 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:

  - With a PFC3B, 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.
  - With a PFC3B, 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—10 gigabits per second, entered as 10000000000
- 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:
  - The minimum token bucket size is 1 kilobyte, entered as 1000 (the *maximum_burst_bytes* parameter must be set larger than the *normal_burst_bytes* parameter)
  - The maximum token bucket size is 32 megabytes, entered as 32000000
  - To sustain a specific rate, set the token bucket size to be at least the rate value divided by 4000, because tokens are removed from the bucket every 1/4000th of a second (0.25 ms).
  - Because the token bucket must be large enough to hold at least one frame, set the parameter 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.

- (Not supported with the **flow** keyword.) 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—10 gigabits per second, entered as 10000000000
- (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` `Router(config-if)# no service-policy [input | output] policy_map_name`	Attaches a policy map to the interface. Removes the policy map from the interface.

	Command	Purpose
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* = **ethernet**, **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 with a PFC3B and only on Layer 3 interfaces (either LAN ports configured as Layer 3 interfaces or VLAN interfaces). With a PFC3B, 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.
- With a PFC3B, 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 Egress DSCP Mutation on a PFC3B

These sections describe how to configure egress DSCP mutation on a PFC3B:

- Configuring Named DSCP Mutation Maps, page 38-69
- Attaching an Egress DSCP Mutation Map to an Interface, page 38-70

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.
	`Router(config)# no mls qos map dscp-mutation map_name`	Reverts to the default 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* Router(config-if)# **no mls qos dscp-mutation** *mutation_map_name*	Attaches an egress DSCP mutation map to the interface. Removes the egress DSCP mutation map from 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* = **ethernet**, **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

PFC QoS supports ingress CoS mutation on IEEE 802.1Q tunnel ports configured to trust received CoS (see the "Applying Ingress CoS Mutation Maps to IEEE 802.1Q Tunnel Ports" section on page 38-72 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.
- PFC QoS supports ingress CoS mutation on WS-X6704-10GE, WS-X6748-SFP, WS-X6724-SFP, and WS-X6748-GE-TX switching modules.
- 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.
	Router(config)# **no mls qos map cos-mutation** *map_name*	Deletes the named map.
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.
	Router(config-if)# **no mls qos cos-mutation** *mutation_map_name*	Removes the ingress CoS mutation map from 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 38-73
- Mapping Received IP Precedence Values to Internal DSCP Values, page 38-74
- Configuring DSCP Markdown Values, page 38-74
- Mapping Internal DSCP Values to Egress CoS Values, page 38-76

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 PFC3B, 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.
	`Router(config)# no mls qos map cos-dscp`	Reverts to the default map.
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 PFC3B, 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.
	`Router(config)# no mls qos map ip-prec-dscp`	Reverts to the default map.
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
Router#
```

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.
	`Router(config)# no mls qos map policed-dscp {normal-burst | max-burst}`	Reverts to the default map.

	Command	Purpose
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 PFC3B 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.
	Router(config)# **no mls qos map dscp-cos**	Reverts to the default 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 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.
	Router(config-if)# **no mls qos trust**	Reverts to the default trust state (untrusted).
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* = **ethernet**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**, **ge-wan**, **pos**, or **atm**.
2. Not supported for **serial**, **pos** or **atm** interface types.

When configuring the trust state of a port, note the following information:

- With no other keywords, the **mls qos trust** command is equivalent to **mls qos trust dscp**.
- 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 38-70).
- 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 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` `Router(config-if)# no mls qos cos port_cos`	Configures the ingress LAN port CoS value. Reverts to the default 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* = **ethernet**, **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 38-80
- Configuring a WRED-Drop Transmit Queue, page 38-81
- Configuring a WRED-Drop and Tail-Drop Receive Queue, page 38-81
- Configuring a WRED-Drop and Tail-Drop Transmit Queue, page 38-82
- Configuring 1q4t/2q2t Tail-Drop Threshold Percentages, page 38-83

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), 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.

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.
	`Router(config-if)# no rcv-queue threshold [queue_id]`	Reverts to the default receive-queue tail-drop threshold percentages.
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.
	Router(config-if)# **no wrr-queue random-detect min-threshold** [*queue_id*]	Reverts to the default 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.
	Router(config-if)# **no wrr-queue random-detect max-threshold** [*queue_id*]	Reverts to the default 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:

- **8q8t** (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.
	Router(config-if)# **no rcv-queue threshold** [*queue_id*]	Reverts to the default 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.
	Router(config-if)# **no rcv-queue random-detect min-threshold** [*queue_id*]	Reverts to the default 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.
	Router(config-if)# **no rcv-queue random-detect max-threshold** [*queue_id*]	Reverts to the default high WRED-drop thresholds.
Step 5	Router(config-if)# **rcv-queue random-detect** *queue_id*	Enables WRED-drop thresholds.
	Router(config-if)# **no rcv-queue random-detect** [*queue_id*]	Enables tail-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**

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.
	`Router(config-if)# no wrr-queue threshold [queue_id]`	Reverts to the default 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.
	`Router(config-if)# no wrr-queue random-detect min-threshold [queue_id]`	Reverts to the default 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.
	`Router(config-if)# no wrr-queue random-detect max-threshold [queue_id]`	Reverts to the default high WRED-drop thresholds.
Step 5	`Router(config-if)# wrr-queue random-detect queue_id`	Enables WRED-drop thresholds.
	`Router(config-if)# no wrr-queue random-detect [queue_id]`	Enables tail-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**

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%`
	
	`Router(config-if)# no wrr-queue threshold [queue_id]`
		Configures the receive- and transmit-queue tail-drop thresholds.
		
		Reverts to the default 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 38-84
- Configuring CoS-Based Queue Mapping, page 38-85

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 CoS-Based Queue Mapping

These sections describe how to configure CoS-based queue mapping:

- Mapping CoS Values to Standard Receive-Queue Thresholds, page 38-85
- Mapping CoS Values to Standard Transmit-Queue Thresholds, page 38-86
- Mapping CoS Values to Strict-Priority Queues, page 38-87
- Mapping CoS Values to Tail-Drop Thresholds on 1q4t/2q2t LAN Ports, page 38-88

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]]]]]]]` `Router(config-if)# no rcv-queue cos-map`	Maps CoS values to the standard receive queue thresholds. Reverts to the default mapping.
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.
	`Router(config-if)# no wrr-queue cos-map`	Reverts to the default mapping.
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.
	`Router(config-if)# no priority-queue cos-map`	Reverts to the default mapping.
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)# no wrr-queue cos-map`	Reverts to the default mapping.
Step 4	`Router(config-if)# end`	Exits configuration mode.
Step 5	`Router# show queueing interface type[1] slot/port`	Verifies the configuration.

1. *type* = **ethernet**, **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 each queue as it is serviced in a round-robin fashion:

- Shaped round robin (SRR)—SRR allows a queue to use only the allocated bandwidth. Supported as an option on Supervisor Engine 32 SFP **1p3q8t** 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** 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.

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 38-84).
		• 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.

	`Router(config-if)# no wrr-queue [bandwidth | shape]`	Reverts to the default bandwidth allocation.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show queueing interface type[1] slot/port`	Verifies the configuration.

1. *type* = **ethernet**, **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** 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.

	`Router(config-if)# no rcv-queue queue-limit`	Reverts to the default size ratio.
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* = **ethernet**, **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 (**1p7q8t**)
    —WS-SUP32-10GE-3B (**1p3q8t**)
    —WS-SUP32-GE-3B (**1p3q8t**)
  - PFC QoS sets the egress strict-priority queue size equal to the high-priority queue size.
- 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 38-93. 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, refer to 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 38-94 and the "Accepting the Traffic Priority Value on Interswitch Links" section on page 38-97 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 Catalyst 6500 series switch Ethernet modules 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)

Catalyst 6500 series switch Ethernet modules 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 38-94) or to trust the packet's incoming DSCP value (as in the "Accepting the Traffic Priority Value on Interswitch Links" section on page 38-97). 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 DCSP-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 38-86.

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.

On the Catalyst 6500 series switch, the scheduling algorithms used on the LAN switching modules 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 22:33:45 (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 38-89).

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 Catalyst 6500 series switch Ethernet ports, the supported rate-limiting method is called policing. Policing is implemented in the PFC3B 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 38-94 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 50Mbps 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 supports 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
```

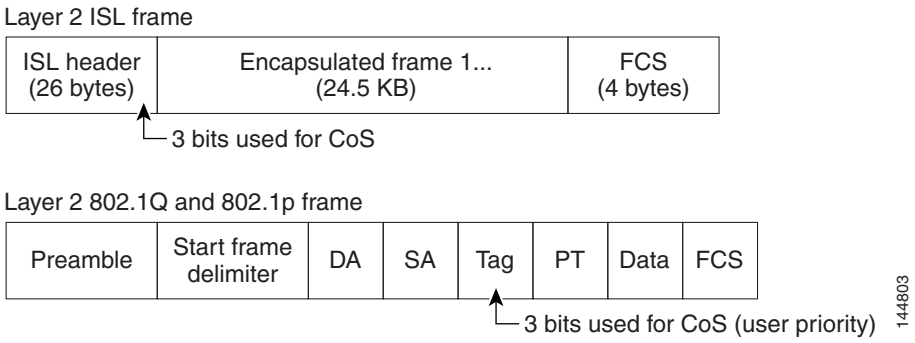
To monitor the policing operation, use these commands:

```
show policy-map interface fastethernet 5/1
show class-map
show mls qos ip fastethernet 5/1
```

PFC QoS Glossary

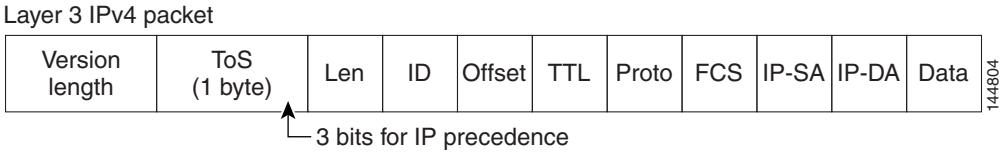
This section defines some of the QoS terminology used in this chapter:

- *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.

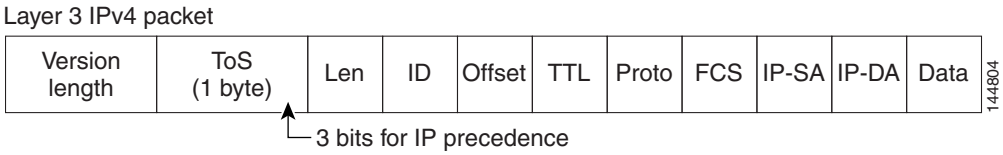


- *Classification* is the process used for selecting traffic to be marked for QoS.

- *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 PFC3B. 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.

CHAPTER 39

Configuring MPLS QoS

This chapter describes how to configure Multiprotocol Label Switching (MPLS) quality of service (QoS) on the Catalyst 6500 series switches.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- MPLS QoS extends to MPLS traffic the PFC QoS features described in Chapter 38, "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

This chapter contains these sections:

- Configuring Short Pipe Mode, page 39-34
- Configuring Uniform Mode, page 39-39

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

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 a MPLS QoS policy.

Trust

For received Layer 3 MPLS packets, the PFC3B 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 PFC3B 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 PFC3B 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 39-20 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 39-23 for information.

Preserving IP ToS

The PFC3B 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 up to eight egress EXP mutation maps to mutate the internal EXP value before it is written as the egress EXP value. You can attach egress EXP mutation maps to these interface types:

- Optical service module (OSM) ports
- LAN or OSM port subinterfaces
- Layer 3 VLAN interfaces
- Layer 3 LAN ports

You cannot attach EXP mutation maps to these interface types:

- Layer 2 LAN ports (switchports)
- FlexWAN ports or subinterfaces

For configuration information, see the"Configuring MPLS QoS Egress EXP Mutation" section on page 39-28.

MPLS DiffServ Tunneling Modes

The PFC3B 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 39-31 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.

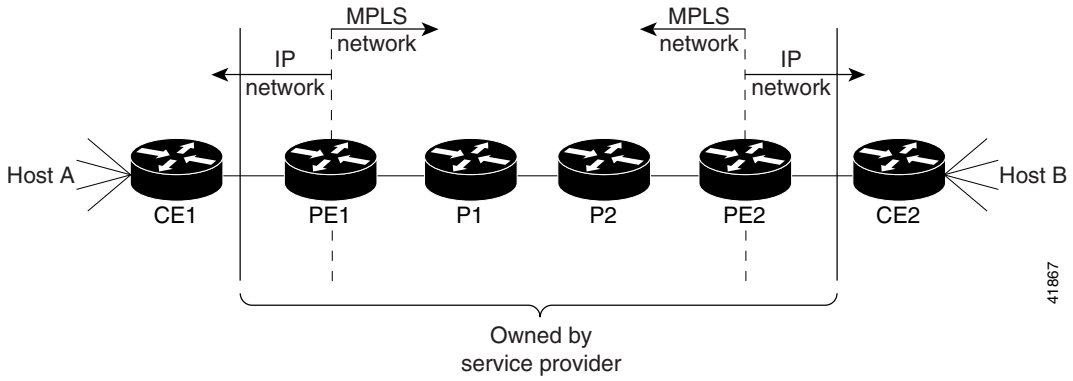
For more information, see the "MPLS DiffServ Tunneling Modes" section on page 39-31.

Mode MPLS QoS

This section describes how MPLS QoS works.

Figure 39-1 shows an MPLS network of a service provider that connects two sites of a customer network.

Figure 39-1 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 39-1, 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 39-6
- LSRs in the Core of an MPLS Network, page 39-6
- LERs at the Output Edge of an MPLS Network, page 39-7

Note The QoS capabilities at the input interface differ depending on whether the input interface is a LAN port, a WAN port on an OSM, or a port adapter on a FlexWAN or Enhanced FlexWAN module. This section is for LAN ports. For information on OSMs, see the *OSM Configuration Note, 12.2SX*. 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 PFC3B 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 PFC3B 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 PFC3B forwards the labeled packets to the appropriate output interface for processing.
6. The PFC3B 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 PFC3B receives the traffic from the input interface and uses the EXP bits to perform classification, marking, and policing.

3. The PFC3B 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 PFC3B forwards the labeled packets to the appropriate output interface for processing.
6. The PFC3B 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 PFC3B 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 PFC3B performs a lookup on the IP address to determine the packet's destination; the PFC3B 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 PFC3B 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 PFC3B 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 PFC3B. 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 39-8
- LERs at the IP Edge (MPLS, MPLS VPN), page 39-9
- LSRs at the MPLS Core, page 39-13

Note The following sections refer to QoS features for LAN ports, OSM ports, and FlexWAN ports. For details about how the different features work, refer to 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 39-9
- Classification for IP-to-MPLS MPLS QoS, page 39-10
- Classification at IP-to-MPLS Ingress Port, page 39-10
- Classification at IP-to-MPLS Egress Port, page 39-10

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 39-11
- Classification for MPLS-to-IP MPLS QoS, page 39-11
- Classification at MPLS-to-IP Ingress Port, page 39-11
- Classification at MPLS-to-IP Egress Port, page 39-12.

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 39-12 for additional MPLS VPN-specific QoS information.

IP to MPLS

The PFC3B 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 PFC3B 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 PFC3B 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 PFC3B applies the port trust and QoS policies, it assigns the internal DSCP. The PFC3B 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 PFC3B preserves the original IP ToS when the MPLS labels are imposed.

The PFC3B 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 PFC3B 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 PFC3B maps the internal DSCP to the imposed EXP value.
- A PFC3B 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 PFC3B maps the internal DSCP to the imposed EXP value.

Classification for IP-to-MPLS 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 **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. OSM and 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.

OSM and 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 and the OSM GE-WAN ports copy the egress CoS into the egress 802.1Q field.

MPLS to IP

PFC3B mode 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 PFC3B assigns the internal DSCP (internal priority that the PFC3B 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 PFC3B classifies the packet based on MPLS EXP ingress or egress policy. The PFC3B 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 PFC3B 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 PFC3B 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 21-4.

For incoming MPLS packets on the PE-to-CE ingress, the PFC3B 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. OSM and 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 OSM and WAN interfaces classify traffic on information in the transmitted IP header.

Note You can use PFC3 QoS features or OSM QoS features in an output policy; however, you cannot use both in the same output policy.

If the egress port is a trunk, the LAN ports and OSM GE-WAN 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 PFC3B.

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 21-4.

The PFC3B 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 PFC3B 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 PFC3B mode 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 PFC3B ignores the port trust state, the ingress CoS, and any policy-map **trust** commands. Instead, the PFC3B 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 PFC3B does next depends on whether it is swapping labels, imposing a new label, or popping a label:

- Swapping labels—When swapping labels, the PFC3B preserves the EXP value in the received topmost label and copies it to the EXP value in the outgoing topmost label. The PFC3B 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 PFC3B 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 PFC3B is performing a markdown, it uses the internal DSCP as an index into the internal DSCP markdown map. The PFC3B maps the result of the internal DSCP markdown to an EXP value using the internal DSCP-to-EXP global map. The PFC3B 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 PFC3B 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 PFC3B 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. PFC3B 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 PFC3B can mark in-profile and mark down out-of-profile traffic. After it marks the internal DSCP, the PFC3B uses the internal DSCP-to-EXP global map to map the internal DSCP to the EXP value in the newly imposed label. The PFC3B then copies the EXP in the imposed label to the underlying swapped label. The PFC3B 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 PFC3B preserves the EXP value in the exposed label. The PFC3B 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 PFC3B maps the internal DSCP to the EXP value in the exposed label using the DSCP-to-EXP global map. The PFC3B 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 QoS

MPLS QoS at the ingress to P1 or P2 supports the following:

- Matching with the **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 **mpls experimental topmost** command.

Classification at MPLS-to-MPLS Ingress Port

LAN port classification is based on the egress CoS from the PFC3B. OSM and 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 PFC3B. OSM and 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 and OSM GE-WAN 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. For OSM WAN ports, (default is trust DSCP) the DSCP is mapped to the imposed EXP.
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 on the Catalyst 6500 series switches 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 38-44.

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 PFC3B 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 PFC3B, 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 PFC3B 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` `Router(config)# no mls qos queueing-only`	Enables queueing-only mode. Disables PFC QoS globally. **Note** You cannot disable queueing-only mode separately.
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**), 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.
	`Router(config-if)# mls qos mpls trust exp`	Reverts to the default trust state where only the EXP value in the incoming packet is trusted.
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 PFC3B, 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 PFC3B hardware that provides the ability to rate limit a particular traffic class to a specific rate. The PFC3B 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 38-17.

To configure traffic policing, use the **police** command. For information on this command, see the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY.

	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 38-61.
- 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 PFC3B, 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 PFC3B does not mark in-profile traffic; when marking down out-of-profile traffic, the PFC3B marks the outgoing topmost label. The PFC3B 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 38-61.
- 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 PFC3B, 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 PFC3B 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 a MPLS QoS Policy Map Class Summary

To display a MPLS QoS policy map class summary, perform this task:

Command	Purpose
Router# **show mls qos mpls** [{**interface** *interface_type interface_number*} | {**module** *slot*}	Displays a MPLS QoS policy map class summary.

This example shows how to display a 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

These sections describe how to configure MPLS QoS egress EXP mutation:

- Configuring Named EXP Mutation Maps, page 39-29
- Attaching an Egress EXP Mutation Map to an Interface, page 39-29

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 name mutated_exp1 mutated_exp2 mutated_exp3 mutated_exp4 mutated_exp5 mutated_exp6 mutated_exp7 mutated_exp8`	Configures a named EXP mutation map.
	`Router(config)# no mls qos map exp-mutation name`	Reverts to the default 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 enter up to eight input 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 configure 15 ingress EXP mutation maps to mutate the internal EXP value before it is written as the ingress EXP value. You can attach ingress EXP mutation maps to any interface that PFC QoS supports.
- PFC QoS derives the egress EXP value from the internal DSCP value. If you configure ingress EXP mutation, PFC QoS does not derive the ingress EXP value from the mutated EXP value.

Attaching an Egress EXP Mutation Map to an Interface

To attach an egress EXP 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 exp-mutation exp-mutation-table-name`	Attaches an egress EXP mutation map to the interface.
	`Router(config-if)# no mls qos exp-mutation`	Removes the egress DSCP mutation map from 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* = **ethernet**, **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 an Ingress-EXP to Internal-DSCP Map, page 39-30
- Configuring a Named Egress-DSCP to Egress-EXP Map, page 39-30

Configuring an Ingress-EXP to Internal-DSCP Map

To configure an ingress-EXP to internal-DSCP map, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos map exp-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.
	`Router(config)# no mls qos map exp-dscp`	Reverts to the default map.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos maps`	Verifies the configuration.

This example shows how to configure an 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 a Named Egress-DSCP to Egress-EXP Map

To configure a named 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 a named 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.
	`Router(config)# no mls qos map dscp-exp`	Reverts to the default map.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos maps`	Verifies the configuration.

This example shows how to configure a named 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 PFC3B, 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 39-31.

  For the configuration information, see the "Configuring Short Pipe Mode" section on page 39-34.

- 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 39-32.

  For the configuration procedure, see the "Configuring Uniform Mode" section on page 39-39.

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.

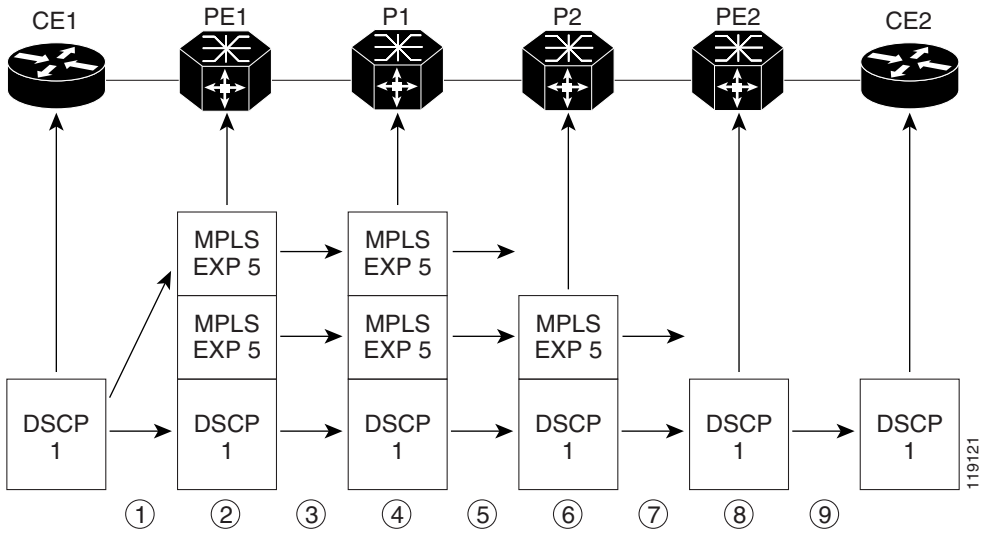
Short Pipe Mode

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 39-2 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:

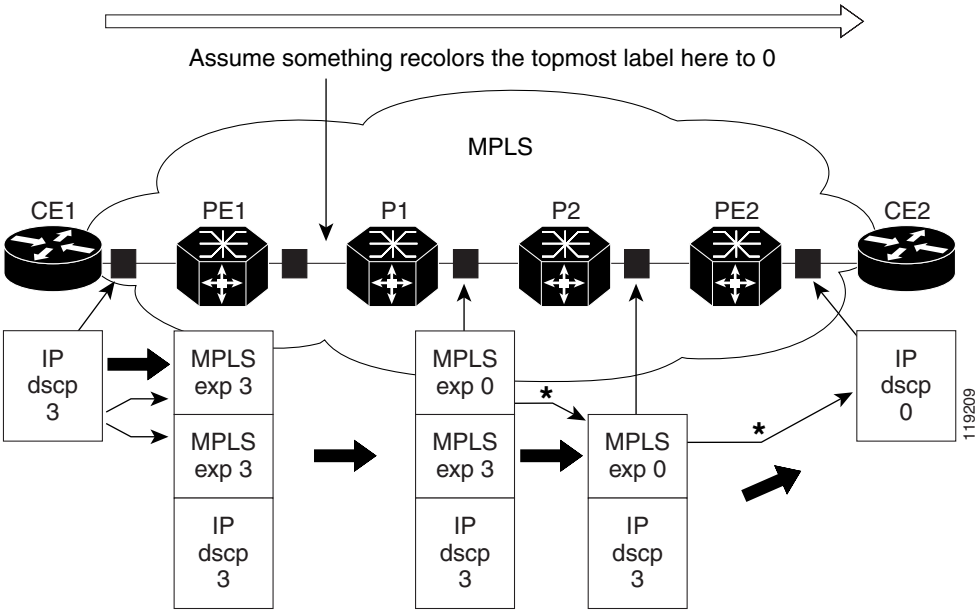
- Short Pipe mode is not supported if the MPLS-to-IP egress interface is EoMPLS (the adjacency has the end of marker (EOM) bit set).

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 39-3 Uniform Mode Operation



*In both the MPLS-to-MPLS and the MPLS-to-IP cases, the PHBs of the topmost popped label is copied into the new top label or the IP DSCP if no label remains

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 39-35
- Configuring Ingress PE Router—P Facing Interface, page 39-36
- Configuring the P Router—Output Interface, page 39-37
- Configuring the Egress PE Router—Customer Facing Interface, page 39-38

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 or OSM 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 PFC3B IP policies. For information about the classification for PFC3B IP policies based on IP ACL/DSCP/precedence, see Chapter 38, "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 OSM and 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 OSM and 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 OSM and 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:

- Configuring the Ingress PE Router—Customer Facing Interface, page 39-39
- Configuring the Ingress PE Router—P Facing Interface, page 39-40
- Configuring the Egress PE Router—Customer Facing Interface, page 39-41

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 PFC3B to copy the IP PHB into the MPLS PHB.

Note This description applies to PFC QoS for LAN or OSM 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
```

CHAPTER 40

Configuring PFC QoS Statistics Data Export

This chapter describes how to configure PFC QoS statistics data export on Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter contains these sections:

Understanding PFC QoS Statistics Data Export

Note

- The PFC QoS statistics data export feature is completely separate from NetFlow Data Export and does not interact with it.
- The PFC QoS statistics data export feature supports the PFC QoS features described in Chapter 38, "Configuring PFC QoS," that are implemented in the port ASICs and the PFC3B.

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 aggregate policers configured on the Catalyst 6500 series switch.

PFC QoS Statistics Data Export Default Configuration

Table 40-1 shows the PFC QoS statistics data export default configuration.

Table 40-1 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, page 40-2
- Enabling PFC QoS Statistics Data Export for a Port, page 40-3
- Enabling PFC QoS Statistics Data Export for a Named Aggregate Policer, page 40-4
- Enabling PFC QoS Statistics Data Export for a Class Map, page 40-5
- Setting the PFC QoS Statistics Data Export Time Interval, page 40-6
- Configuring PFC QoS Statistics Data Export Destination Host and UDP Port, page 40-7
- Setting the PFC QoS Statistics Data Export Field Delimiter, page 40-9

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.
	`Router(config)# no mls qos statistics-export`	Disables 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.
	`Router(config-if)# no mls qos statistics-export`	Disables 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* = **ethernet**, **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.
	`Router(config)# no mls qos statistics-export aggregate-policer aggregate_policer_name`	Disables 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”)
- PFC3B 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.
	`Router(config)# no mls qos statistics-export class-map classmap_name`	Disables 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”)
  - PFC3B 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”)
  - PFC3B 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.
	`Router(config)# no mls qos statistics-export interval interval_in_seconds`	Reverts to the default time interval for the PFC QoS statistics data export.
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 40-2 lists the supported PFC QoS data export facility and severity parameter values.

Table 40-2 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

Table 40-2 Supported PFC QoS Data Export Facility Parameter Values (continued)

Name	Definition	Name	Definition
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 40-3 lists the supported PFC QoS data export severity parameter values.

Table 40-3 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.
	`Router(config)# no mls qos statistics-export delimiter`	Reverts to the default 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
```

CHAPTER 41

Configuring Network Admission Control

This chapter describes how to configure Network Admission Control (NAC) on Catalyst 6500 series switches.

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

- *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

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.

Catalyst 6500 series switches support NAC Layer 2 IP validation. NAC Layer 2 IP validation operates on edge switches but 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.

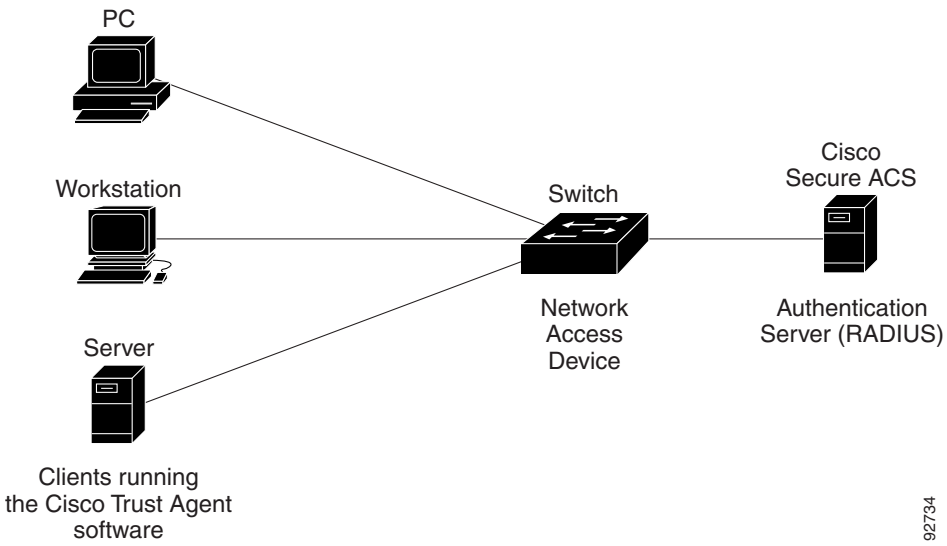
NAC provides *posture validation* for routed traffic on Catalyst 6500 series switches. Posture validation reduces the exposure of a virus to the network. This feature allows network access based on the antivirus credentials of the network device that is requesting network access. These credentials may 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 41-1.

Figure 41-1 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 that is connected to a switch access port through a direct connection, an IP phone, or a wireless access point. 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.

  The CTA software is also referred to as the *posture agent* or the *antivirus client*.

- Switch (edge switches)—This is the network access device that provides 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. The switch relays Extensible Authentication Protocol (EAP) messages between the endpoints and the authentication server.

  For Catalyst 6500 series switches, 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.

- 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.

  In this release, 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.

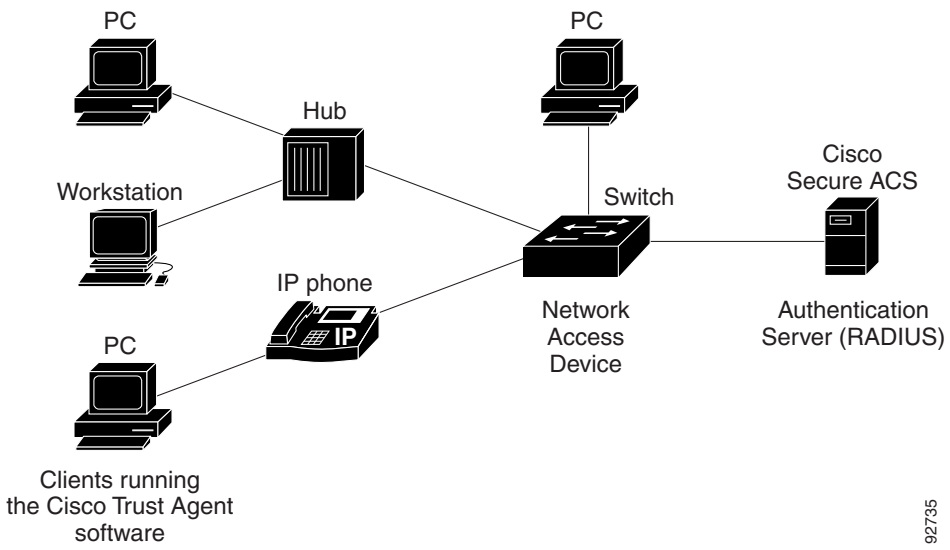
Note When the AAA server is down, the AAA down policy is applied only if there is no existing policy associated with the host. Typically, during revalidation when the AAA server goes down, the policies being used for the host are retained.

NAC Layer 2 IP Validation

You can use NAC Layer 2 IP on an access port on 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 switch access port through a direct connection, an IP phone, or a wireless access point, as shown in Figure 41-2.

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 41-2 Network Using NAC Layer 2 IP



These sections describe NAC Layer 2 IP validation:

- Posture Validation, page 41-4
- Cisco Secure ACS and AV Pairs, page 41-6
- Audit Servers, page 41-7
- ACLs, page 41-8
- NAC Timers, page 41-8
- NAC Layer 2 IP Validation and Redundant Supervisor Engines, page 41-10

Posture Validation

NAC Layer 2 IP supports the posture validation of multiple hosts on the same switch port, as shown in Figure 41-2.

When you enable NAC Layer 2 IP validation on a switch 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.

When DHCP snooping occurs initiating posture validation, it takes precedence over initiating posture validation when ARP snooping occurs. If only dynamic ARP inspection is enabled on the access VLAN assigned to a switch 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.

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 to ports 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 switch port 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.

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.

Cisco Secure ACS and AV Pairs

When NAC Layer 2 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 = switch 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 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.

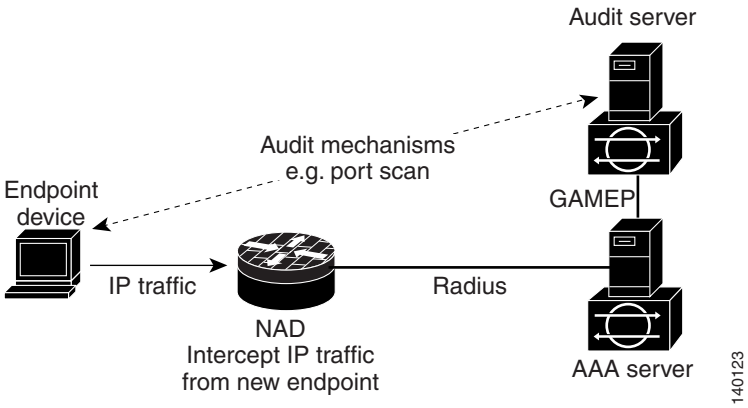
For more information about AV pairs that are supported by Cisco IOS software, see the ACS configuration and command reference documentation about the software releases running on the AAA clients.

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 41-3 shows how audit servers fit into the typical topology.

Figure 41-3 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 happens 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 sever 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 Layer 2 IP validation on a switch port, you must also configure a default port ACL on a switch port. You should also apply the default ACL 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 switch 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. However, if the switch gets an host access policy from the Cisco Secure ACS but the default ACL is not configured, the NAC Layer 2 IP configuration does not take effect.

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 switch 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 41-8
- Idle Timer, page 41-8
- Retransmission Timer, page 41-9
- Revalidation Timer, page 41-10
- Status-Query Timer, page 41-10

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 Layer 2 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 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.

For the Catalyst 6500 series switch, 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 Layer 2 IP Validation and Redundant Supervisor Engines

On Catalyst 6500 series switches with redundant supervisor engines, when RPR mode redundancy is configured, a switchover causes the loss of all information about currently postured hosts. When SSO mode redundancy is configured, a switchover triggers a reposturing of all currently postured hosts.

NAC Layer 2 IP Validation and RPR Redundancy

When RPR mode redundancy is configured, a switchover will lose all information regarding currently postured hosts. When SSO mode redundancy is configured, a switchover will trigger a reposturing of all currently postured hosts.

AAA Down Policy for NAC Layer 2 IP Validation

With the AAA down policy feature, the validation process operates in the following order:

1. A new session is detected.
2. Before posture validation is triggered, and if the AAA server is unreachable, the AAA down policy is applied and session state is maintained as AAA DOWN.
3. When the AAA server is once again available, a revalidation is retriggered for the host.

Note When the AAA server is down, the AAA down policy is applied only if there is no existing policy associated with the host. During revalidation when the AAA server goes down, the policies being used for the host are retained.

Configuring NAC

This section contains this configuration information:

Default NAC Configuration

By default, NAC Layer 2 IP validation is disabled.

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 or Cisco IOS firewall (CBAC) is 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 switch port to which hosts are connected.
- The ACL must permit EAPoUDP traffic for LPIP to function.
- NAC Layer 2 IP does not validate the posture of IPv6 traffic and does not apply access policies to IPv6 traffic.

- NAC Layer 2 IP is not supported if the switchport 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 switch 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.
- 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 switch port with ACEs permitting DNS packets.
- If you want to forward the 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.
- If NAC Layer 2 IP validation is configured on a switch 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.
- The DHCP traffic should be permitted in the interface default ACL and the host policy for DHCP snooping to function.
- 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 switch port access VLAN. This configuration allows you to bypass the redirection of the HTTP traffic destined for the remediation servers.

Configuring NAC Layer 2 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 ratelimit layer2 ip ip-admission`** *`pps (burst)`*	Enables the rate limiting of the IP admission traffic to the CPU.
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** *rule-name* interface configuration command.
Step 8	`Router(config)#` **`exit`**	Returns to global configuration mode.
Step 9	`Router(config)#` **`aaa new-model`**	Enables AAA.

	Command	Purpose
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.
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* | *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.

	Command	Purpose
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	`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# 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.

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.

This example shows how to configure NAC Layer 2 IP validation on a switch interface:

```
Router# configure terminal
Router(config)# ip admission 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 name 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 enter 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 enter identity-profile configuration mode.
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 return 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 a NAC AAA Down Policy

Note This feature is only available on the Catalyst 6500 series switch and the Catalyst 7600 router.

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 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** *rule-name* interface configuration command.
Step 7	`Router(config)# exit`	Returns to global configuration mode.
Step 8	`Router(config)# aaa new-model`	Enables AAA.
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.

	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*—Set the number of times that the switch sends the ARP probe. The range is from 1 to 5. The default is 3. • **interval** *interval*—Set 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.
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.

	Command	Purpose
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 41-21
- Displaying NAC Information, page 41-21

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.

CHAPTER 42

Configuring IEEE 802.1X Port-Based Authentication

This chapter describes how to configure IEEE 802.1X port-based authentication 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, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding 802.1X Port-Based Authentication

The IEEE 802.1X standard defines a client-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.

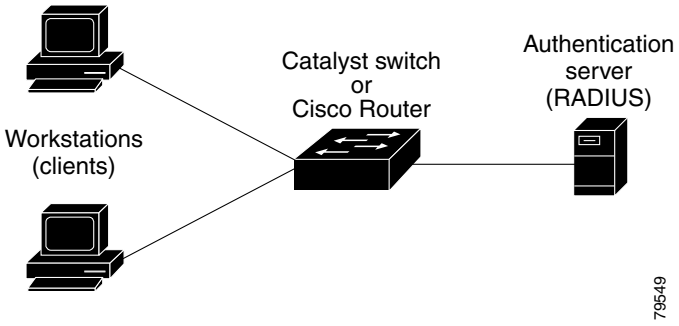
Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

These sections describe IEEE 802.1X port-based authentication:

Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles as shown in Figure 42-1.

Figure 42-1 802.1X Device Roles



The specific roles shown in Figure 42-1 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, 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*)—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.

Authentication Initiation and Message Exchange

The switch or the client can initiate authentication. If you enable authentication on a port by using the **dot1x port-control auto** interface configuration command, 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.

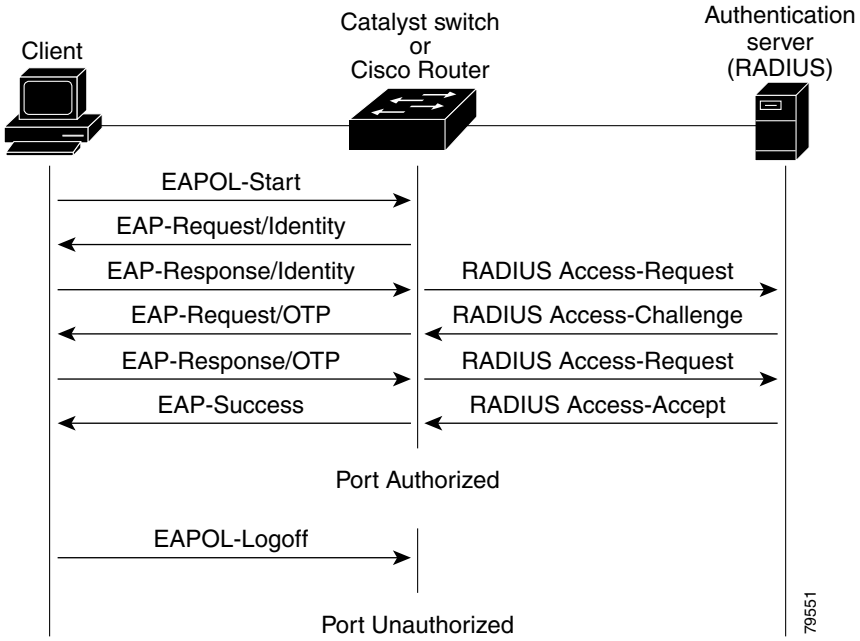
If the client does not receive an EAP-request/identity frame from the switch during bootup, the client can initiate authentication by sending an EAPOL-start frame, which prompts the switch to request the client's identity.

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 42-4.

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 switch port becomes authorized. For more information, see the "Ports in Authorized and Unauthorized States" section on page 42-4.

The specific exchange of EAP frames depends on the authentication method being used. Figure 42-2 shows a message exchange initiated by the client using the One-Time-Password (OTP) authentication method with a RADIUS server.

Figure 42-2 Message Exchange



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 is connected 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 **dot1x port-control** interface configuration command 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.

Supported Topologies

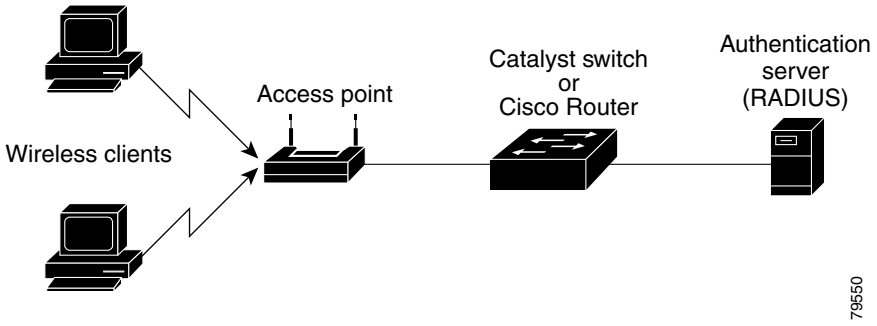
The 802.1X port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration (see Figure 42-1 on page 42-2), only one client can be connected to the 802.1X-enabled switch port. The switch detects the client 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.

Figure 42-3 shows 802.1X port-based authentication in a wireless LAN. The 802.1X port is configured as a multiple-host port that becomes authorized as soon as one client is authenticated. When the port is authorized, all other hosts indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the switch denies access to the network to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and the wireless access point acts as a client to the switch.

Figure 42-3 Wireless LAN Example



Default 802.1X Port-Based Authentication Configuration

Table 42-1 shows the default 802.1X configuration.

Table 42-1 Default 802.1X Configuration

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled
RADIUS server IP address	None specified
RADIUS server UDP authentication port	1812
RADIUS server key	None specified
Per-interface 802.1X protocol enable state	Disabled (force-authorized) **Note** The port transmits and receives normal traffic without 802.1X-based authentication of the client.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet period	60 seconds (number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client)

Table 42-1 Default 802.1X Configuration (continued)

Feature	Default Setting
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)
Maximum retransmission number	2 times (number of times that the switch will send an EAP-request/identity frame before restarting the authentication process)
Multiple host support	Disabled
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)

802.1X Port-Based Authentication Guidelines and Restrictions

When configuring 802.1X port-based authentication, follow these guidelines and restrictions:

- When 802.1X is enabled, ports are authenticated before any other Layer 2 or Layer 3 features are enabled.
- The 802.1X protocol is supported on both Layer 2 static-access ports and Layer 3 routed ports, but it is not supported on these port types:
  - Trunk port—If you try to enable 802.1X on a trunk port, an error message appears, and 802.1X is not enabled. If you try to change the mode of an 802.1X-enabled port to trunk, the port mode is not changed.
  - EtherChannel port—Before enabling 802.1X on the port, you must first remove it from the EtherChannel port-channel interface. If you try to enable 802.1X on an EtherChannel port-channel interface or on an individual active port in an EtherChannel, an error message appears, and 802.1X is not enabled. If you enable 802.1X on a not-yet active individual port of an EtherChannel, the port does not join the EtherChannel.
  - Secure port—You cannot configure a secure port as an 802.1X port. If you try to enable 802.1X on a secure port, an error message appears, and 802.1X is not enabled. If you try to change an 802.1X-enabled port to a secure port, an error message appears, and the security settings are not changed.
  - Switch Port Analyzer (SPAN) destination port—You can enable 802.1X on a port that is a SPAN destination port; however, 802.1X is disabled until the port is removed as a SPAN destination port. You can enable 802.1X on a SPAN source port.

Configuring 802.1X Port-Based Authentication

These sections describe how to configure 802.1X port-based authentication:

Enabling 802.1X Port-Based 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 configure 802.1X port-based authentication, perform this task:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA.
	`Router(config)# no aaa new-model`	Disables AAA.
Step 2	`Router(config)# aaa authentication dot1x {default} method1 [method2...]`	Creates an 802.1X port-based authentication method list.
	`Router(config)# no aaa authentication dot1x {default | list_name}`	Clears the configured method list.
Step 3	`Router(config)# dot1x system-auth-control`	Globally enables 802.1X port-based authentication.
	`Router(config)# no dot1x system-auth-control`	Globally disables 802.1X port-based authentication.
Step 4	`Router(config)# interface type[1] slot/port`	Enters interface configuration mode and specifies the interface to be enabled for 802.1X port-based authentication.
Step 5	`Router(config-if)# dot1x port-control auto`	Enables 802.1X port-based authentication on the interface.
	`Router(config-if)# no dot1x port-control auto`	Disables 802.1X port-based authentication on the interface.

	Command	Purpose
Step 6	`Router(config)# end`	Returns to privileged EXEC mode.
Step 7	`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* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

When you enable 802.1X port-based authentication, note the following information:

- To create a default list that is used when a named list is *not* specified in the **authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.
- Enter at least one of these keywords:
  - **group radius**—Use the list of all RADIUS servers for authentication.
  - **none**—Use no authentication. The client is automatically authenticated by the switch without using the information supplied by the client.

This example shows how to enable AAA and 802.1X on Fast Ethernet port 5/1:

```
Router# configure terminal
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# dot1x system-auth-control
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show dot1x all

Dot1x Info for interface FastEthernet5/1
----------------------------------------------------
AuthSM State      = FORCE UNAUTHORIZED
BendSM State      = IDLE
PortStatus        = UNAUTHORIZED
MaxReq            = 2
MultiHosts        = Disabled
Port Control      = Force UnAuthorized
QuietPeriod       = 60 Seconds
Re-authentication = Disabled
ReAuthPeriod      = 3600 Seconds
ServerTimeout     = 30 Seconds
SuppTimeout       = 30 Seconds
TxPeriod          = 30 Seconds
```

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.
	`Router(config)# no ip radius source-interface`	Prevents the RADIUS packets from having the IP address of the previously 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.
	`Router(config)# no radius-server host {hostname | ip_address}`	Deletes the specified RADIUS server.
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)# end`	Returns to privileged EXEC mode.

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, refer to 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

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, refer to the RADIUS server documentation.

This example shows how to configure the RADIUS server parameters on the switch:

```
Router# configure terminal
Router(config)# ip radius source-interface Vlan80
Router(config)# radius-server host 172.120.39.46
Router(config)# radius-server key rad123
Router(config)# end
```

Enabling Periodic Reauthentication

You can enable periodic 802.1X client reauthentication and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication attempts is 3600.

Automatic 802.1X client reauthentication is a global setting and cannot be set for clients connected to individual ports. To manually reauthenticate the client connected to a specific port, see the "Manually Reauthenticating the Client Connected to a Port" section on page 42-11.

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*	Selects an interface to configure.
Step 2	`Router(config-if)# dot1x reauthentication`	Enables periodic reauthentication of the client, which is disabled by default.
	`Router(config-if)# no dot1x reauthentication`	Disables periodic reauthentication of the client.
Step 3	`Router(config-if)# dot1x timeout reauth-period` *seconds*	Sets the number of seconds between reauthentication attempts. The range is 1 to 65535; the default is 3600 seconds. This command affects the behavior of the switch only if periodic reauthentication is enabled.
	`Router(config-if)# no dot1x timeout reauth-period`	Returns to the default reauthorization period.
Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Router# show dot1x all`	Verifies your entries.

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

This example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000:

```
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* = **ethernet**, **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
Starting reauthentication on 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* = **ethernet**, **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
Starting reauthentication on FastEthernet 5/1
```

Changing 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 idle time is determined by the quiet-period value. 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 smaller number than the default.

To change the quiet period, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects an interface to configure.
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.
	`Router(config-if)# no dot1x timeout quiet-period`	Returns to the default quiet time.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

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

This example shows how to set the quiet time on the switch to 30 seconds:

```
Router(config-if)# dot1x timeout quiet-period 30
```

Changing 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 behavioral problems with certain clients and authentication servers.

To change the amount of time that the switch waits for client notification, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects an interface to configure.
Step 2	`Router(config-if)# dot1x timeout tx-period seconds`	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request. The range is 1 to 65535 seconds; the default is 30.
	`Router(config-if)# dot1x timeout tx-period`	Returns to the default retransmission time.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

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

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)# 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. You may set the amount of time that the switch waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the switch-to-client retransmission time for the EAP-request frames, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects an interface to configure.
Step 2	`Router(config-if)# dot1x timeout supp-timeout seconds`	Sets the switch-to-client retransmission time for the EAP-request frame.
	`Router(config-if)# no dot1x timeout supp-timeout`	Returns to the default retransmission time.
Step 3	`Router# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

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

This example shows how to set the switch-to-client retransmission time for the EAP-request frame to 25 seconds:

```
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. You may set the amount of time that the switch waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the value for the retransmission of Layer 4 packets from the switch to the authentication server, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface type[1] slot/port`	Selects an interface to configure.
Step 2	`Router(config-if)# dot1x timeout server-timeout seconds`	Sets the switch-to-authentication-server retransmission time for Layer 4 packets.
	`Router(config-if)# no dot1x timeout server-timeout`	Returns to the default retransmission time.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

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

This example shows how to set the switch-to-authentication-server retransmission time for Layer 4 packets to 25 seconds:

```
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 behavioral 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*	Selects an interface to configure.
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.
	`Router(config-if)# no dot1x max-req`	Returns to the default retransmission number.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

1. *type* = **ethernet**, **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-if)# dot1x max-req 5
```

Enabling Multiple Hosts

You can attach multiple hosts to a single 802.1X-enabled port as shown in Figure 42-3 on page 42-5. In this mode, only one of the attached hosts must be successfully authorized for all hosts to be granted network access. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), all attached clients are denied access to the network.

To allow multiple hosts (clients) on an 802.1X-authorized port that has the **dot1x port-control** interface configuration command set to **auto**, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Selects an interface to configure.
Step 2	`Router(config-if)# dot1x host-mode multi-host`	Allows multiple hosts (clients) on an 802.1X-authorized port. **Note** Make sure that the **dot1x port-control** interface configuration command set is set to **auto** for the specified interface.
	`Router(config-if)# dot1x host-mode single-host`	Disables multiple hosts on the port.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x interface` *type*[1] *slot/port*	Verifies your entries.

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

This example shows how to enable 802.1X on Fast Ethernet interface 5/1 and to allow multiple hosts:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x host-mode multi-host
```

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*	Selects an interface to configure.
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* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Displaying 802.1X Status

To display global 802.1X administrative and operational status for the switch, use the **show dot1x** privileged EXEC command. To display the 802.1X administrative and operational status for a specific interface, use the **show dot1x interface** *interface-id* privileged EXEC command.

For detailed information about the fields in these displays, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY.

CHAPTER 43

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, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

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, port security applies the shutdown 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 43-6 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 43-4.

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.

Default Port Security Configuration

Table 43-1 shows the default port security configuration for an interface.

Table 43-1 Default Port Security Configuration

Feature	Default Setting
Port security	Disabled on a port
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:

- To bring a secure port out of the error-disabled state with the default port security configuration, enter the **errdisable recovery cause shutdown** global configuration command, or manually reenable it 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 *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, for complete syntax information.
- Port security learns authorized 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 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.

- Port security supports trunks.
- 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.
- Port security and 802.1X port-based authentication cannot both be configured on the same port:
  - 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.
- 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.

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 43-7).

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 switchport.
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` `Router(config-if)# no switchport port-security`	Enables port security on the trunk. Disables 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* = **ethernet**, **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 switchport.
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` `Router(config-if)# no switchport port-security`	Enables port security on the port. Disables 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* = **ethernet**, **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.
Step 2	Router(config-if)# **switchport port-security violation** {**protect** | **restrict** | **shutdown**} Router(config-if)# **no switchport port-security violation**	(Optional) Sets the violation mode and the action to be taken when a security violation is detected. Reverts to the default configuration (**shutdown**).
Step 3	Router(config-if)# **do show port-security interface** *type*[1] *slot/port* | **include** *violation_mode*[2]	Verifies the configuration.

1. *type* = **ethernet**, **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.

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 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.
	`Router(config-if)# no switchport port-security maximum`	Reverts to the default configuration.
Step 3	`Router(config-if)# do show port-security interface type[1] slot/port | include Maximum`	Verifies the configuration.

1. *type* = **ethernet**, **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.
	`Router(config-if)# no switchport port-security mac-address sticky`	Disables port security with sticky MAC addresses on a port.

1. *type* = **ethernet**, **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 | voice | access}]`	Configures a static MAC address as secure on the port. When you specify the **vlan** keyword, the static secure MAC address is configured based on the following argument or keyword: • *vlan_ID*—The MAC address is configured in the specifiedVLAN. The *vlan_ID* argument is supported only on trunk ports. • **voice**—The MAC address is configured in the voice VLAN. This keyword is supported only on multi-VLAN access ports. • **access**—The MAC address is configured in the access (data) VLAN. This keyword is supported only on access ports or multi-VLAN access ports.
	`Router(config-if)# no switchport port-security mac-address [sticky] mac_address`	Clears a static secure MAC address from the port.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show port-security address`	Verifies the configuration.

1. *type* = **ethernet**, **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 43-8).
- 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 by *vlan_ID*.
  - 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 43-10
- Configuring Secure MAC Address Aging Time on a Port, page 43-11

Configuring 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).
	`Router(config-if)# no switchport port-security aging type`	Reverts to the default MAC address aging type.
Step 3	`Router(config-if)# do show port-security interface type[1] slot/port | include Time`	Verifies the configuration.

1. *type* = **ethernet**, **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).
	`Router(config-if)# no switchport port-security aging time`	Disables secure MAC address aging time.
Step 3	`Router(config-if)# do show port-security interface type[1] slot/port | include Time`	Verifies the configuration.

1. *type* = **ethernet**, **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* = **ethernet**, **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
```

CHAPTER 44

Configuring CDP

This chapter contains information about how to configure Cisco Discovery Protocol (CDP) on the Catalyst 6500 series switches, which supplements the information in these publications:

- The *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2, "System Management," "Configuring Cisco Discovery Protocol (CDP)" at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf015.html

- The *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.2, "System Management Commands," "CDP Commands" publication at this URL:

  http://www.cisco.com/en/US/docs/ios/12_2/configfun/command/reference/frf015.html

This chapter consists of these sections:

Understanding How CDP Works

CDP is a protocol that runs over Layer 2 (the data link layer) on all Cisco routers, bridges, access servers, and switches. CDP allows network management applications to discover Cisco devices that are neighbors of already known devices, in particular, neighbors running lower-layer, transparent protocols. With CDP, network management applications can learn the device type and the SNMP agent address of neighboring devices. This feature enables applications to send SNMP queries to neighboring devices.

CDP runs on all LAN and WAN media that support Subnetwork Access Protocol (SNAP).

Each CDP-configured device sends periodic messages to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain the time-to-live, or holdtime information, which indicates the length of time a receiving device should hold CDP information before discarding it.

Configuring CDP

These sections describe how to configure CDP:

Enabling CDP Globally

To enable CDP globally, perform this task:

Command	Purpose
`Router(config)# cdp run`	Enables CDP globally.
`Router(config)# no cdp run`	Disables CDP globally.

This example shows how to enable CDP globally:

```
Router(config)# cdp run
```

Displaying the CDP Global Configuration

To display the CDP configuration, perform this task:

Command	Purpose
`Router# show cdp`	Displays global CDP information.

This example shows how to display the CDP configuration:

```
Router# show cdp
Global CDP information:
        Sending CDP packets every 120 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is enabled
Router#
```

For additional CDP show commands, see the "Monitoring and Maintaining CDP" section on page 44-3.

Enabling CDP on a Port

To enable CDP on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{type[1] slot/port} | {port-channel number}}`	Selects the port to configure.
Step 2	`Router(config-if)# cdp enable`	Enables CDP on the port.
	`Router(config-if)# no cdp enable`	Disables CDP on the port.

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

This example shows how to enable CDP on Fast Ethernet port 5/1:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# cdp enable
```

Displaying the CDP Interface Configuration

To display the CDP configuration for a port, perform this task:

Command	Purpose
Router# **show cdp interface** [{{*type*[1] *slot/port*} | {**port-channel** *number*}}]	Displays information about ports where CDP is enabled.

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

This example shows how to display the CDP configuration of Fast Ethernet port 5/1:

```
Router# show cdp interface fastethernet 5/1
FastEthernet5/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 120 seconds
  Holdtime is 180 seconds
Router#
```

Monitoring and Maintaining CDP

To monitor and maintain CDP on your device, perform one or more of these tasks:

Command	Purpose
Router# **clear cdp counters**	Resets the traffic counters to zero.
Router# **clear cdp table**	Clears information about neighbors from the CDP table.
Router# **show cdp**	Displays global information such as frequency of transmissions and the holdtime for packets being transmitted.
Router# **show cdp entry** *entry_name* [**protocol** | **version**]	Displays information about a specific neighbor. The display can be limited to protocol or version information.
Router# **show cdp interface** [*type*[1] *slot/port*}]	Displays information about interfaces on which CDP is enabled.
Router# **show cdp neighbors** [*type*[1] *slot/port*] [**detail**]	Displays information about neighbors. The display can be limited to neighbors on a specific interface and expanded to provide more detailed information.
Router# **show cdp traffic**	Displays CDP counters, including the number of packets sent and received and checksum errors.
Router# **show debugging**	Displays information about the types of debugging that are enabled. Refer to the *Debug Command Reference* for more information about CDP **debug** commands.

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

This example shows how to clear CDP counter configuration:

```
Router# clear cdp counters
```

This example shows how to display information about the neighboring equipment:

```
Router# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
JAB023807H1      Fas 5/3            127         T S       WS-C2948  2/46
JAB023807H1      Fas 5/2            127         T S       WS-C2948  2/45
JAB023807H1      Fas 5/1            127         T S       WS-C2948  2/44
JAB023807H1      Gig 1/2            122         T S       WS-C2948  2/50
JAB023807H1      Gig 1/1            122         T S       WS-C2948  2/49
JAB03130104      Fas 5/8            167         T S       WS-C4003  2/47
JAB03130104      Fas 5/9            152         T S       WS-C4003  2/48
```

CHAPTER 45

Configuring UDLD

This chapter describes how to configure the UniDirectional Link Detection (UDLD) protocol on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding How UDLD Works

These sections describe how UDLD works:

UDLD Overview

The Cisco-proprietary UDLD protocol allows devices connected through fiber-optic or copper (for example, Category 5 cabling) Ethernet cables connected to LAN ports to monitor the physical configuration of the cables and detect when a unidirectional link exists. When a unidirectional link is detected, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems, including spanning tree topology loops.

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.

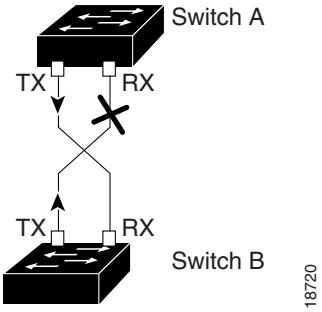
A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does 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.

The Catalyst 6500 series switch periodically transmits UDLD packets to neighbor devices on LAN ports with UDLD enabled. If the packets are echoed back within a specific time frame 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 locally disabled on copper LAN ports to avoid sending unnecessary control traffic on this type of media since it is often used for access ports.

Figure 45-1 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 45-1 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, nonaggressive UDLD with the default 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.

Note In UDLD normal mode, when a unidirectional error is detected, the port is not disabled. In UDLD aggressive mode, when a unidirectional error is detected, the port is disabled.

Default UDLD Configuration

Table 45-1 shows the default UDLD configuration.

Table 45-1 UDLD Default Configuration

Feature	Default Value
UDLD global enable state	Globally disabled
UDLD aggressive mode	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, page 45-3
- Enabling UDLD on Individual LAN Interfaces, page 45-4
- Disabling UDLD on Fiber-Optic LAN Interfaces, page 45-4
- Configuring the UDLD Probe Message Interval, page 45-5
- Resetting Disabled LAN Interfaces, page 45-5

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.
`Router(config)# no udld {enable | aggressive}`	Disables UDLD globally on fiber-optic LAN ports.

Enabling UDLD on Individual LAN Interfaces

To enable UDLD on individual LAN ports, 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 fiber-optic LAN ports, 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* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Disabling UDLD on Fiber-Optic LAN Interfaces

To disable UDLD on individual fiber-optic LAN ports, 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.
	`Router(config-if)# no udld port disable`	Reverts to the **udld enable** global configuration command setting. **Note** This command is only supported on fiber-optic LAN ports.
Step 3	`Router# show udld type[1] slot/number`	Verifies the configuration.

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

Configuring the UDLD Probe Message Interval

To configure the time between UDLD probe messages on 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.
	`Router(config)# no udld message`	Returns to the default value (60 seconds).
Step 2	`Router# show udld type[1] slot/number`	Verifies the configuration.

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

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.

CHAPTER 46

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, refer to these publications:

- The *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.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

- NetFlow version 9 is supported—See this document:

  Cisco IOS NetFlow Configuration Guide.

This chapter contains the following sections:

Understanding NDE

These sections describe how NetFlow Data Export (NDE) works:

NDE Overview

NetFlow collects traffic statistics by monitoring packets that flow through the switch and storing the statistics in the NetFlow table. For more information about NetFlow, see Chapter 47, "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.

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 PISA

NDE on the PISA exports statistics for flows routed in software. The PISA supports NetFlow aggregation, described in this document:

Cisco IOS NetFlow Configuration Guide.

The PISA also supports NetFlow ToS-based router aggregation, described in this document:

Cisco IOS NetFlow Configuration Guide.

NetFlow Sampling is supported on the PISA and is described in this document:

Cisco IOS NetFlow Configuration Guide.

NetFlow version 9 is supported and is described in this document:

Cisco IOS NetFlow Configuration Guide.

NetFlow version 9 record formats are described in this document:

Cisco IOS NetFlow Configuration Guide.

NDE on the PFC3B

NDE on the PFC3B exports statistics for flows routed or bridged in hardware. These sections describe NDE on the PFC3B in more detail:

- NDE Flow Mask, page 46-2
- NDE Versions, page 46-3
- Exporting NetFlow Data, page 46-7
- NetFlow Sampling, page 46-7

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, refer to Chapter 47, “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. Therefore, these fields are blank when you use the **show** command to display the hardware NetFlow table.

NDE Versions

NetFlow version 9 is supported and is described at this URL:

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/12_2/switch/configuration/guide/xcfnfov.html

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 46-1—Version 5 header format
- Table 46-2—Version 7 header format
- Table 46-3—Version 5 flow record format
- Table 46-4—Version 7 flow record format

NetFlow version 9 record formats are describedin this document:

Cisco IOS NetFlow Configuration Guide.

Table 46-1 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 46-2 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 46-3 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 46-11)					
			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. In PFC3BXL or PFC3B mode, for ICMP traffic, contains the ICMP code and type values.
5. Always zero for hardware-switched flows.
6. Populated in PFC3BXL or PFC3B mode.

Table 46-4 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 46-11)					
			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	X	X	X	X	X
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. In PFC3BXL or PFC3B mode, for ICMP traffic, contains the ICMP code and type values.
5. Always zero for hardware-switched flows.

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 46-16 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 46-7
- NetFlow Flow Sampling, page 46-8

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. Therefore, 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 46-5 lists the sampling rates and export intervals for time-based NetFlow flow sampling.

Table 46-5 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 46-4 shows the default NDE configuration.

Table 46-6 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:

- NDE supports IP multicast traffic only with NetFlow version 9.
- NetFlow aggregation must use NDE version 8 or version 9.
- NDE supports bridged IP traffic.
- NDE does not support Internetwork Packet Exchange (IPX) traffic or any other non-IP protocol.

Configuring NDE

These sections describe how to configure NDE:

- Configuring NDE on the PFC3B, page 46-11
- Configuring NDE on the PISA, page 46-13
- Enabling NDE for Ingress-Bridged IP Traffic, page 46-14
- Displaying the NDE Address and Port Configuration, page 46-15
- Configuring NDE Flow Filters, page 46-16
- Displaying the NDE Configuration, page 46-18

Note

- You must enable NetFlow on the PISA Layer 3 interfaces to support NDE on the PFC3B and NDE on the PISA.
- You must enable NDE on the PISA to support NDE on the PFC3B.
- When you configure NAT and NDE on an interface, the PFC3B sends all fragmented packets to the PISA to be processed in software. (CSCdz51590)

Configuring NDE on the PFC3B

These sections describe how to configure NDE on the PFC3B:

- Enabling NDE From the PFC3B, page 46-11
- Populating Additional NDE Fields, page 46-11
- Configuring NetFlow Flow Sampling, page 46-12

Enabling NDE From the PFC3B

To enable NDE from the PFC3B, perform this task:

Command	Purpose
`Router(config)# mls nde sender [version {5 | 7}]`	Enables NDE from the PFC3B and (optionally) configures the NDE version. Do not use this command to enable version 9 records. Instead, use **ip flow-export version 9**, which is explained in the "Configuring NDE on the PISA" section on page 46-13
`Router(config)# no mls nde sender`	Disables NDE from the PFC3B.
`Router(config)# no mls nde sender version`	Reverts to the default (version 7).

Note

- NDE from the PFC3B uses the source interface configured for the PISA (see the "Configuring the PISA NDE Source Layer 3 Interface" section on page 46-13).
- NetFlow version 9 is supported and 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 PFC3B:

```
Router(config)# mls nde sender
```

This example shows how to enable NDE from the PFC3B 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 46-3 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.
`Router(config)# no mls nde interface`	Disables population of the additional fields.

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 PFC3B:

- Configuring NetFlow Flow Sampling Globally, page 46-12
- Configuring NetFlow Flow Sampling on a Layer 3 Interface, page 46-12

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.
	`Router(config)# no mls sampling`	Clears the NetFlow flow sampling configuration.
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.
- With a PFC3, 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.
	`Router(config-if)# no mls netflow sampling`	Disables 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 PISA

These sections describe how to configure NDE on the PISA:

- Configuring the PISA NDE Source Layer 3 Interface, page 46-13
- Configuring the NDE Destination, page 46-14
- Configuring NetFlow Sampling, page 46-14

Configuring the PISA NDE Source Layer 3 Interface

To configure the Layer 3 interface used as the source of the NDE packets containing statistics from the PISA, 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 PISA.
`Router(config)# no ip flow-export source`	Clears the NDE source interface configuration.

When configuring the PISA 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`	Configures the NDE destination IP address and UDP port.
`Router(config)# no ip flow-export destination ip_address udp_port_number`	Clears the NDE destination configuration.

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.
- When you configure two destinations, the RP CPU utilization is increased because 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 PISA supports NetFlow sampling for software-routed traffic.

For additional information, see the following document:

Cisco IOS NetFlow Configuration Guide.

Enabling NDE for Ingress-Bridged IP Traffic

NDE supports ingress-bridged IP traffic.

Note To enable NetFlow for bridged IP traffic on a VLAN, you must create a corresponding VLAN interface, assign it an IP address, and enter the **no shutdown** command to bring up the interface.

NDE is enabled by default when you enable NetFlow on the VLAN. 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.
Router(config)# **no ip flow export layer2-switched vlan** *vlan_ID*[-*vlan_ID*] [, *vlan_ID*[-*vlan_ID*]]	Disables NDE for ingress-bridged IP traffic in the specified VLANs.

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 the NDE export flow IP address and UDP port configuration.
Router# **show ip flow export**	Displays 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 46-18.

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.
Router(config)# **no mls nde flow** {**exclude** | **include**}	Clears the port flow filter configuration.

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.
Router(config)# **no mls nde flow** {**exclude** | **include**}	Clears the port flow filter configuration.

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.
Router(config)# **no mls nde flow** {**exclude** | **include**}	Clears port filter configuration.

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.
Router(config)# **no mls nde flow** {**exclude** | **include**}	Clears port filter configuration.

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 46-18.

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#
```

CHAPTER 47

Configuring NetFlow

This chapter describes how to configure NetFlow statistics collection on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to this publication:

http://www.cisco.com/en/US/docs/ios/netflow/command/reference/nf_book.html

This chapter contains the following sections:

Understanding NetFlow

These sections describe how NetFlow works:

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 PISA captures statistics for flows routed in software and the NetFlow table on the PFC3B 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).

You can configure NetFlow to collect statistics for both routed and bridged traffic.

Collecting and exporting a large volume of statistics can significantly impact supervisor engine and PISA processor 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.
- Sampled NetFlow exports data for a subset of traffic in a flow, which can greatly reduce the volume of statistics exported. Sampled NetFlow 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. Note that NetFlow aggregation increases switch 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 free up table space for new entries.

NetFlow on the PISA

The NetFlow table on the PISA captures statistics for flows routed in software. NetFlow on the PISA supports NetFlow aggregation. For information about the NetFlow aggregation schemes, refer to the following document:

Cisco IOS NetFlow Configuration Guide.

For information about configuring NetFlow aggregation on the PISA, refer to the following document:

Cisco IOS NetFlow Configuration Guide.

NetFlow on the PISA supports ToS-based router aggregation, described in this document:

Cisco IOS NetFlow Configuration Guide.

For information about NetFlow for multicast IP, refer to the NetFlow Multicast Support documentation, available in the following document:

Cisco IOS NetFlow Configuration Guide.

The NetFlow Multicast Support document contains a prerequisite specifying that you need to configure multicast fast switching or multicast distributed fast switching (MDFS). However, this prerequisite does not apply when configuring NetFlow multicast support on the Supervisor Engine 32 PISA.

NetFlow on the PFC3B

The NetFlow table on the PFC3B captures statistics for flows routed in hardware. The PFC3B supports sampled NetFlow and NetFlow aggregation. The PFC3B does not support NetFlow ToS-based router aggregation.

These sections describe NetFlow on the PFC3B in more detail:

- Flow Masks, page 47-3
- Flow Mask Conflicts, page 47-4

Flow Masks

A flow is a unidirectional stream of packets between a given source and a given destination. A flow mask specifies the fields in the incoming packet that NetFlow uses to identify the flow. NetFlow gathers statistics for each flow defined by the flow mask.

The PFC3B supports the following flow masks:

- source-only—A less-specific flow mask. The PFC3B maintains one entry for each source IP address. Statistics for all flows from a given source IP address aggregate into this entry.
- destination—A less-specific flow mask. The PFC3B maintains one entry for each destination IP address. Statistics for all flows to a given destination IP address aggregate into this entry.
- destination-source—A more-specific flow mask. The PFC3B maintains one entry for each source and destination IP address pair. Statistics for all flows between the same source IP address and destination IP address aggregate into this entry.
- destination-source-interface—A more-specific flow mask. Adds the source VLAN SNMP ifIndex to the information in the destination-source flow mask.
- full—A more-specific flow mask. The PFC3B creates and maintains a separate table entry for each IP flow. A full entry includes the source IP address, destination IP address, protocol, and protocol ports.
- full-interface—The most-specific flow mask. Adds the source VLAN SNMP ifIndex to the information in the full-flow mask.

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 source-only, the NetFlow table contains only one entry per source IP address. The statistics for all flows from a given source are accumulated in the one entry. However, if the flow mask is configured as 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 Configuration Guidelines and Restrictions" section on page 47-5 for information about NetFlow table capacity.

Flow Mask Conflicts

Several features use the NetFlow table. Table 47-1 lists the flow mask requirements for each feature.

Table 47-1 Feature Requirements for Flow Masks

Feature	Source	Destination	Destination Source	Destination Source Interface	Full Flow	Interface Full Flow	Non-interface Full Flow
Reflexive ACL						X	
TCP Intercept					X	X	
Context Based Access Control (CBAC)					X		
Web Cache Redirect (WCCP)					X	X	
Server Load Balancing (SLB)					X	X	
Network Address Translation (NAT)						X	X
NetFlow Data Export (NDE)	X	X	X	X	X	X	
Sampled NetFlow						X	
NetFlow Aggregation		X		X	X	X	
Microflow Policing	X	X			X	X	

Because of the variety of feature requirements, potential flow mask conflicts can occur. Note the following flow mask constraints:

- All features must share the same limited set of flow masks.
- The PFC3B can apply only one flow mask to each packet lookup.

The Feature Manager software in the PISA is responsible for resolving feature conflicts. The Feature Manager's main strategy is to select a common flow mask that satisfies all the configured NetFlow features.

However, the Feature Manager may not find a common flow mask for the configured features, because some features have very specific requirements for the flow mask. To resolve the feature conflict, Feature Manager software may direct one of the features to be processed in software on the PISA.

In the extreme case, Feature Manager software gives priority to the feature that is configured first and rejects configuration requests for subsequent features. When you attempt to configure a subsequent feature that the Feature Manager cannot accommodate, you receive a failure message at the CLI.

Follow these guidelines to avoid problems with feature conflicts:

- Configure your highest priority features first. If an unresolvable conflict occurs, your lower priority features may be blocked.
- If possible, configure features only on the interfaces where the feature is required.
- Pay attention to response messages. If the Feature Manager turns off hardware assist for a feature, you need to ensure that feature processing does not overload the RP processor.

Note the following specific feature conflicts:

- CBAC requires the full flow mask, and is given priority over other flow-based features. If a flow mask conflict occurs, the other flow-based features are processed in the PISA.
- In general, NDE is flexible because you configure the minimum flow mask. If you have configured other flow-based features, Feature Manager software may set a more specific flow mask to meet all the feature requirements.
- Sampled NetFlow requires the full-interface flow mask. This may cause conflict with other flow-based features on the same interface.
- NDE conflicts with QoS. NDE and QoS microflow policing cannot be configured on the same interface.
- If NAT is configured on a Layer 3 interface with any feature that uses dynamic ACEs (for example, Web Proxy Authentication or NAC Layer 3 IP validation), trailing fragments may not be NAT translated correctly if NAT is configured for overload.You can use the **mls ip nat netflow-frag-l4-zero** command to ensure that NAT functions correctly in this case.

Default NetFlow Configuration

Table 47-2 shows the default NetFlow configuration.

Table 47-2 Default NetFlow Configuration

Feature	Default Value
NetFlow of routed IP traffic	Disabled
NetFlow of ingress bridged IP traffic	Disabled
Sampled NetFlow	Disabled
NetFlow Aggregation	Disabled

NetFlow Configuration Guidelines and Restrictions

When configuring NetFlow, follow these guidelines and restrictions:

- The CEF table (and not the NetFlow table) implements Layer 3 switching in hardware.
- NetFlow supports bridged IP traffic.
- NetFlow supports multicast IP traffic.
- 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 47-3 lists the recommended maximum utilization levels.

Table 47-3 NetFlow Table Utilization

PFC	Recommended NetFlow Table Utilization	Total NetFlow Table Capacity
PFC3B	117,760 (115 K) entries	131,072 (128 K) entries

Configuring NetFlow

These sections describe how to configure NetFlow:

- Configuring NetFlow on the PFC3B, page 47-6
- Configuring NetFlow on the PISA, page 47-10

Note When you configure NAT on an interface, the PFC3B sends all fragmented packets to the PISA to be processed in software. (CSCdz51590)

Configuring NetFlow on the PFC3B

These sections describe how to configure NetFlow statistics collection on the PFC3B:

- NetFlow PFC3B Commands Summary, page 47-6
- Enabling NetFlow on the PFC3B, page 47-7
- Setting the Minimum IP MLS Flow Mask, page 47-7
- Configuring the MLS Aging Time, page 47-7
- Configuring NetFlow Aggregation on the PFC3B, page 47-9
- Enabling NetFlow for Ingress-Bridged IP Traffic, page 47-9
- Enabling NetFlow for Multicast IP Traffic, page 47-10
- Displaying PFC3B NetFlow Information, page 47-10

NetFlow PFC3B Commands Summary

Table 47-4 shows a summary of the NetFlow commands available on the PFC3B.

Table 47-4 Summary of PFC3B NetFlow commands

Command	Purpose
mls netflow	Enables NetFlow on the PFC3B.
mls flow ip	Sets the minimum flow mask.
mls aging	Sets the configurable aging parameters.
show mls netflow {...}	Displays NetFlow PFC3B information for unicast and multicast traffic.
show mls netflow aggregation flowmask	Displays the NetFlow aggregation flow mask.

Note

- When you configure NetFlow aggregation on the PISA, it is enabled automatically on the PFC3B.
- When you configure NetFlow for Layer 2 traffic on the PISA, it is enabled automatically on the PFC3B.
- When you configure multicast NetFlow on the PISA, it is enabled automatically on the PFC3B.

Enabling NetFlow on the PFC3B

To enable NetFlow statistics collection on the PFC3B, perform this task:

Command	Purpose
`Router(config)# mls netflow`	Enables NetFlow on the PFC3B.
`Router(config)# no mls netflow`	Disables NetFlow on the PFC3B.

This example shows how to disable NetFlow statistics collection on the PFC3B (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 PFC3B. The actual flow mask may be more specific than the level configured in the **mls flow ip** command, if other configured features need a more specific flow mask (see the "Flow Mask Conflicts" section on page 47-4).

To set the minimum IP MLS flow mask, perform this task:

Command	Purpose
`Router(config)# mls flow ip {source | destination | destination-source | interface-destination-source | full | interface-full}`	Sets the minimum IP MLS flow mask for the protocol.
`Router(config)# no mls flow ip`	Reverts to the default IP MLS flow mask (null).

This example shows how to set the minimum IP MLS flow mask:

```
Router(config)# mls flow ip 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: destination address
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 Configuration Guidelines and Restrictions" section on page 47-5), 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.
`Router(config)# no mls aging fast`	Disables fast aging.
`Router(config)# no mls aging {long | normal}`	Reverts to the default MLS aging time.

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 NetFlow Aggregation on the PFC3B

NetFlow Aggregation is configured automatically on the PFC3B when you configure NetFlow Aggregation on the PISA (see the "Configuring NetFlow Aggregation on the PISA" section on page 47-11).

To display NetFlow Aggregation information for the PFC3B, 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 PFC3B does 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#
```

Enabling NetFlow for Ingress-Bridged IP Traffic

NetFlow for ingress-bridged IP traffic on the PFC3B is enabled when you configure NetFlow for ingress-bridged IP traffic on the PISA. See the "Enabling NetFlow for Ingress-Bridged IP Traffic" section on page 47-11.

Enabling NetFlow for Multicast IP Traffic

NetFlow for multicast IP traffic on the PFC3B is enabled when you configure NetFlow for multicast IP traffic on the PISA.

For additional information, see the "Enabling NetFlow for Multicast IP Traffic" section on page 47-12.

Displaying PFC3B NetFlow Information

To display information about NetFlow on the PFC3B, use the following command:

Command	Purpose
`Router(config)# show mls netflow {aggregation | aging | creation | flowmask | ip | ipv6 | mpls | table-contention | usage}`	Displays information about NetFlow on the PFC3B.

Configuring NetFlow on the PISA

These sections describe how to configure NetFlow on the PISA:

Summary of NetFlow Commands on the PISA

Table 47-5 shows the NetFlow commands available on the PISA.

Table 47-5 Summary of PISA NetFlow Commands

Command	Purpose
interface x **ip flow ingress**	Enables NetFlow on the PISA and the PFC3B for the specified interface.
ip flow-aggregation cache	Configure NetFlow aggregation. Note that configuring aggregation on the PISA also enables aggregation for the PFC3B.
export version {8|9}	Specifies aggregation data export format 8 or 9.
mask source minimum x	Specifies the aggregation minimum mask.
ip flow ingress layer2-switched vlan x	Enables NetFlow for Layer 2 switched traffic.
interface x **ip multicast netflow {ingress|egress}**	Enables NetFlow multicast traffic on the specified interface (for PISA and PFC3B).
show ip cache flow aggregation	Shows the configuration for NetFlow aggregation.
show ip cache verbose flow	Shows the configuration for multicast NetFlow.

Enabling NetFlow on the PISA

To enable NetFlow on the PISA, perform this task for each Layer 3 interface from which you want NetFlow:

	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 on the selected interface, for flows routed in hardware or software. You must also enable NetFlow on the PFC3B to enable NetFlow for flows routed in hardware.

Configuring NetFlow Aggregation on the PISA

To configure NetFlow aggregation on the PISA, use the procedures at this document:

Cisco IOS NetFlow Configuration Guide.

To configure NetFlow ToS-based router aggregation on the PISA, use the procedures at this URL:document:

Cisco IOS NetFlow Configuration Guide.

Note

- When you configure NetFlow aggregation on the PISA, it is configured automatically on the PFC3B (see the "Configuring NetFlow Aggregation on the PFC3B" section on page 47-9).
- The PFC3B does not support NetFlow ToS-based router aggregation.

Enabling NetFlow for Ingress-Bridged IP Traffic

NetFlow supports ingress-bridged IP traffic.

Note

- When you enable NetFlow for ingress-bridged IP traffic, the statistics are available to the Sampled NetFlow feature (see the "NetFlow Sampling" section on page 46-7).
- To enable NetFlow for bridged IP traffic on a VLAN, you must create a corresponding VLAN interface, assign it an IP address, and enter the **no shutdown** command to bring up the interface.

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 PFC3B be enabled with the **mls netflow** command.
`Router(config)# no ip flow ingress layer2-switched vlan vlan_ID[-vlan_ID] [, vlan_ID[-vlan_ID]]`	Disables NetFlow for ingress-bridged IP traffic in the specified VLANs.

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
```

Enabling NetFlow for Multicast IP Traffic

To enable NetFlow for multicast IP, 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 on the interface.
Step 3	`Router(config-if)# ip multicast netflow {ingress | egress}`	Enables NetFlow multicast traffic on the specified interface (for PISA and PFC3B). • Specify **ingress** to enable NetFlow multicast ingress accounting • Specify **egress** to enable NetFlow multicast egress accounting

For additional information about NetFlow for multicast IP, refer to the NetFlow Multicast Support documentation, available at the following document:

Cisco IOS NetFlow Configuration Guide.

The NetFlow Multicast Support document contains a prerequisite specifying that you need to configure multicast fast switching or multicast distributed fast switching (MDFS). However, this prerequisite does not apply when configuring NetFlow multicast support on the Supervisor Engine 32 PISA.

CHAPTER 48

Configuring Local SPAN, RSPAN, and ERSPAN

This chapter describes how to configure local Switched Port Analyzer (SPAN), remote SPAN (RSPAN), and Encapsulated RSPAN (ERSPAN) on the Catalyst 6500 series switches.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html
- OSM WAN ports and FlexWAN ports do not support SPAN, RSPAN or ERSPAN.

This chapter consists of these sections:

Understanding How Local SPAN, RSPAN, and ERSPAN Work

These sections describe how local SPAN, RSPAN, and ERSPAN work:

Local SPAN, RSPAN, and ERSPAN Overview

Local SPAN, RSPAN, and ERSPAN sessions allow you to monitor traffic on one or more ports, or one or more VLANs, and send the monitored traffic to one or more destination ports. You can configure per-VLAN filtering on destination trunk ports.

Local SPAN, RSPAN, and ERSPAN all send traffic to a network analyzer such as a SwitchProbe device or other Remote Monitoring (RMON) probe. SPAN does not affect the switching of traffic on source ports or VLANs. SPAN sends a copy of the packets received or transmitted by the source ports and VLANs to the destination port. You must dedicate the destination port for SPAN use.

These sections provide an overview of local SPAN, RSPAN, and ERSPAN:

Local SPAN Overview

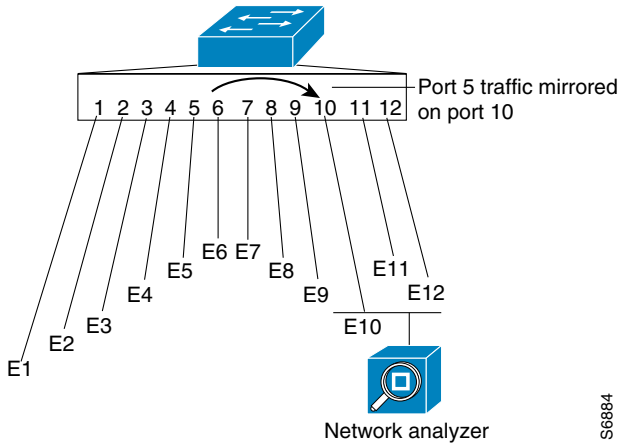
A local SPAN session is an association of source ports and source VLANs with one or more destination ports. You configure a local SPAN session on a single switch. Local SPAN does not have separate source and destination sessions.

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 port for analysis (see Figure 48-1). For example, as shown in Figure 48-1, 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 48-1 Example SPAN Configuration



RSPAN Overview

RSPAN supports source ports, source VLANs, and destination ports on different switches, which provides remote monitoring of multiple switches across your network (see Figure 48-2).

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 destination ports with the RSPAN VLAN.

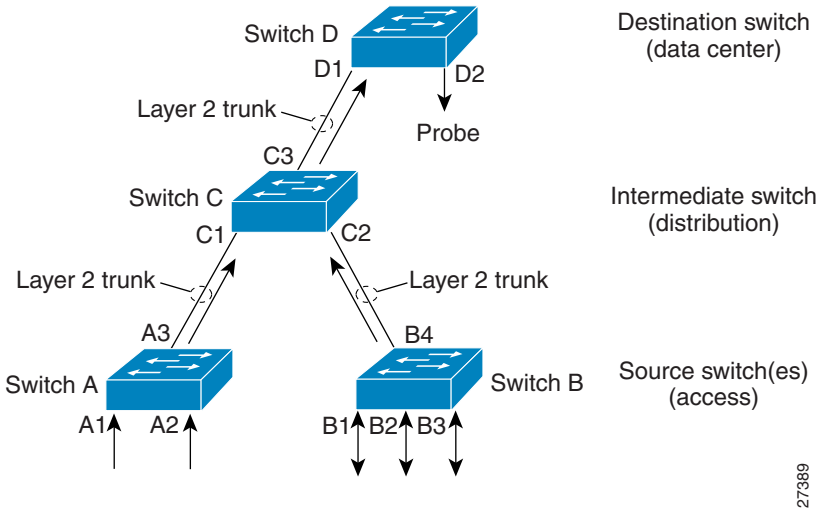
The traffic for each RSPAN session is carried as Layer 2 nonroutable traffic over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches. All participating switches must be trunk-connected at Layer 2.

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 destination ports.

Figure 48-2 RSPAN Configuration



ERSPAN Overview

ERSPAN supports source ports, source VLANs, and destination ports on different switches, which provides remote monitoring of multiple switches across your network (see Figure 48-3).

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.

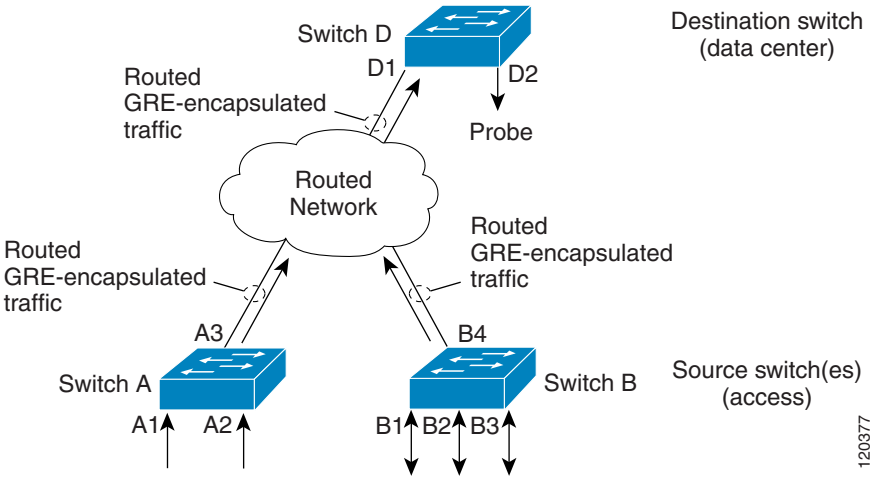
To configure an ERSPAN source session on one switch, you associate a set of source ports or VLANs with a destination IP address, ERSPAN ID number, and optionally with a VRF name. To configure an ERSPAN destination session on another switch, you associate the destination ports with the source IP address, 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 destination ports.

Figure 48-3 ERSPAN Configuration



Monitored Traffic

These sections describe the traffic that local SPAN, RSPAN, and ERSPAN can monitor:

- Monitored Traffic Direction, page 48-4
- Monitored Traffic, page 48-4
- Duplicate Traffic, page 48-4

Monitored Traffic Direction

You can configure local SPAN sessions, RSPAN source sessions, and ERSPAN source sessions to monitor ingress traffic (called ingress SPAN), or to monitor egress traffic (called egress SPAN), or to monitor traffic flowing in both directions.

Ingress SPAN copies traffic received by the source ports and VLANs for analysis at the destination port. Egress SPAN copies traffic transmitted from the source ports and VLANs. When you enter the **both** keyword, SPAN copies the traffic received and transmitted by the source ports and VLANs to the destination port.

Monitored Traffic

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 port. 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 port, 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 Ports, page 48-5
- Source VLANs, page 48-5

Source Ports

A source port is a port monitored for traffic analysis. You can configure both switched and routed ports as SPAN source ports. SPAN can monitor one or more source ports in a single SPAN session. You can configure source ports in any VLAN. Trunk ports can be configured as source ports and mixed with nontrunk source ports. SPAN does not copy the encapsulation from a source trunk port.

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 in the source VLANs become source ports.

Local SPAN, RSPAN, and ERSPAN Destination Ports

A destination port is a Layer 2 or Layer 3 LAN port to which local SPAN, RSPAN, or ERSPAN sends traffic for analysis.

When you configure a port as a destination port, it can no longer receive any traffic. When you configure a port as a destination port, the port is dedicated for use only by the SPAN feature. A SPAN destination port does not forward any traffic except that required for the SPAN session.

You can configure trunk ports as destination ports, which allows destination trunk ports to transmit encapsulated traffic. For local SPAN, you can configure per-VLAN filtering on destination trunk ports using allowed VLAN lists (see the "Configuring Destination Trunk Port VLAN Filtering" section on page 48-21).

Local SPAN, RSPAN, and ERSPAN Configuration Guidelines and Restrictions

These sections describe local SPAN, RSPAN, and ERSPAN configuration guidelines and restrictions:

Feature Incompatiblities

These feature incompatibilities exist with local SPAN, RSPAN, and ERSPAN:

- With a PFC3, 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.
- A port-channel interface (an EtherChannel) cannot be a SPAN destination.
- You cannot configure active member ports of an EtherChannel as SPAN destination ports. Inactive member ports of an EtherChannel can be configured as SPAN destination ports but they are put into the suspended state and carry no traffic.
- Because SPAN destination ports drop ingress traffic, these features are incompatible with SPAN destination ports:
  - 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 destination ports can participate in IEEE 802.3Z Flow Control.

Local SPAN, RSPAN, and ERSPAN Session Limits

These are the PFC3 local SPAN, RSPAN, and ERSPAN session limits:

Total Sessions	Local SPAN, RSPAN Source, or ERSPAN Source Sessions	RSPAN Destination Sessions	ERSPAN Destination Sessions
66	2 (ingress or egress or both)	64	23

These are the PFC3 local SPAN, RSPAN, and ERSPAN source and destination 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 port 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 port. (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 destination ports.
- All the SPAN destination ports receive all of the traffic from all the SPAN sources.

Note You can configure destination trunk port VLAN filtering using allowed VLAN lists (see the “Configuring Destination Trunk Port VLAN Filtering” section on page 48-21).

For local SPAN and RSPAN, you can configure Source VLAN Filtering (see the “Configuring Source VLAN Filtering for Local SPAN and RSPAN” section on page 48-20).

- 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.
- You cannot mix source VLANs and filter VLANs within a session. You can have source VLANs or filter VLANs, but not both at the same time.
- 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 destination ports as trunks to send locally tagged traffic to the traffic analyzer.

  Note A destination port configured as a trunk tags traffic from a Layer 3 LAN source port with the internal VLAN used by the Layer 3 LAN port.

- 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.
- A port specified as a destination port in one SPAN session cannot be a destination port for another SPAN session.
- A port configured as a destination port cannot be configured as a source port.
- Destination ports never participate in any spanning tree instance. Local SPAN includes BPDUs in the monitored traffic, so any BPDUs seen on the destination port are from the source port. RSPAN does not support BPDU monitoring.
- All packets sent through the switch for transmission from a port configured as an egress source are copied to the destination port, including packets that do not exit the switch through the port because STP has put the port into the blocking state, or on a 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:

- For VSPAN sessions with both ingress and egress configured, two packets are forwarded from the destination port 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:

- Supervisor Engine 2 does not support RSPAN if you configure an egress SPAN source for a local SPAN session.
- Supervisor Engine 2 does not support egress SPAN sources for local SPAN if you configure RSPAN.
- All participating switches must be trunk-connected at Layer 2.
- 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:

- 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.
- 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 7-10) 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 48-16).
- 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 48-18).
- 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 destination ports. 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.

Configuring Local SPAN, RSPAN, and ERSPAN

These sections describe how to configure local SPAN, RSPAN, and ERSPAN:

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)#` **`no monitor permit-list`**	Disables use of the destination port permit list.
Step 4	`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 5	`Router(config)#` **`no monitor permit-list destination interface`** *type*[1] *slot/port*[-*port*] [, *type*[1] *slot/port* - *port*]	Deletes from or clears an existing destination port permit list.
Step 6	`Router(config)#` **`do show monitor permit-list`**	Verifies the configuration.

1. *type* = **ethernet**, **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 Local SPAN

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*}	Associates the local SPAN session number and the destination ports.
	`Router(config)# no monitor session` {*session_number* | **all** | **local** | **range** *session_range*[[,*session_range*],...]}	Clears the monitor configuration.

When configuring local SPAN sessions, note the following information:

- *local_span_session_number* can range from 1 to 66.
- *single_interface* is **interface** *type slot/port*; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *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* , ...
- To tag the monitored traffic as it leaves a destination port, you must configure the destination port to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination Port as an Unconditional Trunk" section on page 48-21).

When clearing monitor sessions, note the following information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*.

Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

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 48-23.

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]|[,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` `Router(config-vlan)# no remote-span`	Configures the VLAN as an RSPAN VLAN. Clears the RSPAN VLAN configuration.
Step 4	`Router(config-vlan)# end`	Updates the VLAN database and returns to privileged EXEC mode.

Configuring RSPAN Source Sessions

To configure an RSPAN source session, 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.
Step 4	`Router(config)# no monitor session {`*session_number* `| all | range` *session_range*`[[,`*session_range*`],...] | remote}`	Clears the monitor configuration.

When configuring monitor sessions, note the following information:

- To configure RSPAN VLANs, see the "Configuring RSPAN VLANs" section on page 48-13.
- *RSPAN_source_span_session_number* can range from 1 to 66.
- *single_interface* is **interface** *type slot/port*; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *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* , ...

When clearing monitor sessions, note the following information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*.

Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

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 48-23.

Configuring RSPAN Destination Sessions

Note 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* `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*}}	Associates the RSPAN destination session number with the destination ports.
Step 4	`Router(config)# no monitor session` {*session_number* | `all` | `range` *session_range*[[,*session_range*],...] | `remote`}	Clears the monitor configuration.

When configuring monitor sessions, note the following information:

- 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 Port as an Unconditional Trunk" section on page 48-21).
- *RSPAN_destination_span_session_number* can range from 1 to 66.
- *single_interface* is **interface** *type slot/port*; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *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* , ...

When clearing monitor sessions, note the following information:

- Enter the **no monitor session** *number* command with no other parameters to clear session *session_number*.
- *session_range* is *first_session_number-last_session_number*.

Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

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 48-23.

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 48-16
- Configuring ERSPAN Destination Sessions, page 48-18

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.
	`Router(config)# no monitor session` {*session_number* | **all** | **range** *session_range*[[,*session_range*],...]}	Clears the monitor configuration.
Step 3	`Router(config-mon-erspan-src)# description` *session_description*	(Optional) Describes the ERSPAN source session.
Step 4	`Router(config-mon-erspan-src)# shutdown`	(Default) Inactivates the ERSPAN source session.
	`Router(config-mon-erspan-src)# no shutdown`	Activates the ERSPAN source session.
Step 5	`Router(config-mon-erspan-src)# source` {{*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 source ports or VLANs, and selects the traffic direction to be monitored.
Step 6	`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.

	Command	Purpose
Step 7	`Router(config-mon-erspan-src)# destination`	Enters ERSPAN source session destination configuration mode.
Step 8	`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 48-18, Step 7).
Step 9	`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 48-18, Step 8).
Step 10	`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 11	`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 12	`Router(config-mon-erspan-src-dst)# ip prec ipp_value`	(Optional) Configures the IP precedence value of the packets in the ERSPAN traffic.
Step 13	`Router(config-mon-erspan-src-dst)# ip dscp dscp_value`	(Optional) Configures the IP DSCP value of the packets in the ERSPAN traffic.
Step 14	`Router(config-mon-erspan-src-dst)# vrf vrf_name`	(Optional) Configures the VRF name to use instead of the global routing table.
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 or spaces.

  Note You can enter 240 characters after the **description** command.

- *ERSPAN_source_span_session_number* can range from 1 to 66.
- *single_interface* is **interface** *type slot/port*; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *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.

When clearing monitor sessions, note the following information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*.

Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

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 48-23.

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.
	`Router(config)# no monitor session` {*session_number* | `all` | `range` *session_range*[[,*session_range*],...]}	Clears the monitor configuration.
Step 3	`Router(config-mon-erspan-dst)# description` *session_description*	(Optional) Describes the ERSPAN destination session.

	Command	Purpose
Step 4	`Router(config-mon-erspan-dst)# shutdown` `Router(config-mon-erspan-dst)# no shutdown`	(Default) Inactivates the ERSPAN destination session. Activates the ERSPAN destination session.
Step 5	`Router(config-mon-erspan-dst)# destination {single_interface | interface_list | interface_range | mixed_interface_list}}`	Associates the ERSPAN destination session number with the destination ports.
Step 6	`Router(config-mon-erspan-dst)# source`	Enters ERSPAN destination session source configuration mode.
Step 7	`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 48-16, Step 8.
Step 8	`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 48-16, Step 9.
Step 9	`Router(config-mon-erspan-dst-src)# vrf vrf_name`	(Optional) Configures the VRF name used instead of the global routing table.
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 66.
- *single_interface* is **interface** *type slot/port*; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *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 48-16, Step 8).

- *ERSPAN_flow_id* can range from 1 to 1023.

When clearing monitor sessions, note the following information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*.

Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

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 48-23.

Configuring Source VLAN Filtering for Local SPAN and RSPAN

Source VLAN filtering monitors specific VLANs when the source is a trunk port.

Note To configure source VLAN filtering for ERSPAN, see the "Configuring ERSPAN" section on page 48-16.

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.
	`Router(config)# no monitor session` *session_number* `filter` *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*	Clears source VLAN filtering.

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
```

Configuring a Destination Port as an Unconditional Trunk

To tag the monitored traffic as it leaves a destination port, configure the destination port as a trunk.

To configure the destination port 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*	Selects the LAN port to configure.
Step 3	`Router(config-if)# switchport`	Configures the LAN port for Layer 2 switching (required only if the LAN port is not already configured for Layer 2 switching).
Step 4	`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 5	`Router(config-if)# switchport mode trunk`	Configures the port to trunk unconditionally.
Step 6	`Router(config-if)# switchport nonegotiate`	Configures the trunk not to use DTP.

1. *type* = **ethernet**, **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
Router(config-if)# switchport nonegotiate
```

Configuring Destination Trunk Port VLAN Filtering

Note In addition to filtering VLANs on a trunk, you can also apply the allowed VLAN list to access ports.

When a destination port is a trunk, you can use the list of VLANs allowed on the trunk to filter the traffic transmitted from the destination port. (CSCeb01318)

Destination trunk port VLAN filtering removes the restriction that all destination ports receive all the traffic from all the sources. Destination trunk port VLAN filtering allows you to select, on a per-VLAN basis, the traffic that is transmitted from each destination trunk port to the network analyzer.

To configure destination trunk port VLAN filtering on a destination trunk port, 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* = **ethernet**, **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 port 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
```

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 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
```

CHAPTER 49

Configuring SNMP IfIndex Persistence

This chapter describes how to configure the SNMP ifIndex persistence feature on Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

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 49-2 (Optional)
- Enabling and Disabling SNMP IfIndex Persistence on Specific Interfaces, page 49-2 (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.

	Command	Purpose
Step 2	`Router(config-if)# snmp ifindex persist`	Enables SNMP ifIndex persistence on the specified interface.
	`Router(config-if)# no snmp ifindex persist`	Disables 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
```

CHAPTER 50

Power Management and Environmental Monitoring

This chapter describes the power management and environmental monitoring features in the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

Understanding How Power Management Works

These sections describe power management in the Catalyst 6500 series switches:

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, refer to 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 50-5.

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 50-4).

Table 50-1 describes how the system responds to changes in the power supply configuration.

Table 50-1 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.
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.

Table 50-1 Effects of Power Supply Configuration Changes (continued)

Configuration Change	Effect
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.
	`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.

Power Cycling Power Supplies

If you have redundant power supplies and you power cycle one of the power supplies, only that power supply is power cycled. If you power cycle both power supplies, the system goes down and comes back up in 10 seconds.

If you only have one power supply and you power cycle that power supply, the system goes down and comes back up in 10 seconds.

This example shows how to power cycle a power supply:

```
Router# hw-module power-supply 2 power-cycle
Power-cycling the power supply may interrupt service.
Proceed with power-cycling? [confirm]
Power-cycling power-supply 1
22:10:23: %C6KPWR-SP-2-PSFAIL: power supply 1 output failed.
22:10:25: %C6KENV-SP-4-PSFANFAILED: the fan in power supply 1 has failed
22:10:33: %C6KPWR-SP-4-PSOK: power supply 1 turned on.
22:10:33: %C6KENV-SP-4-PSFANOK: the fan in power supply 1 is OK
Router#
```

Determining System Power Requirements

The power supply size determines the system power requirements. When you use the 1000 W and 1300 W power supplies, you might have configuration limitations depending on the size of chassis and type of modules installed. For information about power consumption, refer to the *Release Notes for Cisco IOS Release 12.2ZY on the Supervisor Engine 32 PISA* publication at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/release/notes/ol_13011.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 the LAN module in the Catalyst 6500 series switch:

```
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%
```

This example shows how to display EOBC-related statistics for the route processor and the switch processor:

```
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
```

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
```

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%
```

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%
```

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
```

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
```

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%
```

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
```

This example shows how to display the capacity and utilization of QoS policer resources for each EARL in the Catalyst 6500 series switch.

```
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%
```

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
```

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 How Environmental Monitoring Works

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

The LEDs 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, giving you notice of a problem that could turn critical if corrective action is not taken.

When the system has an alarm (major or minor), that indicates an overtemperature condition, the alarm is not canceled nor is any action taken (such as module reset or shutdown) for 5 minutes. If the temperature falls 5°C (41°F) below the alarm threshold during this period, the alarm is canceled.

Table 50-2 lists the environmental indicators for the supervisor engine and switching modules.

Note Refer to the *Catalyst 6500 Series Switch Module Installation Guide* for additional information on LEDs, including the supervisor engine SYSTEM LED.

Table 50-2 Environmental Monitoring for Supervisor Engine and Switching Modules

Component	Alarm Type	LED Indication	Action
Supervisor engine temperature sensor exceeds major threshold[1]	Major	STATUS[2] LED red[3]	Generates syslog message and an SNMP trap. If there is a redundancy situation, the system switches to a redundant supervisor engine and the active supervisor engine shuts down. If there is no redundancy situation and the overtemperature condition is not corrected, the system shuts down after 5 minutes.
Supervisor engine temperature sensor exceeds minor threshold	Minor	STATUS LED orange	Generates syslog message and an SNMP trap. Monitors the condition.
Redundant supervisor engine temperature sensor exceeds major or minor threshold	Major	STATUS LED 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	STATUS LED orange	Monitors the condition if a minor alarm is generated.

Table 50-2 Environmental Monitoring for Supervisor Engine and Switching Modules (continued)

Component	Alarm Type	LED Indication	Action
Switching module temperature sensor exceeds major threshold	Major	STATUS LED red	Generates syslog message and SNMP. Powers down the module[4].
Switching module temperature sensor exceeds minor threshold	Minor	STATUS LED orange	Generates syslog message and an SNMP trap. Monitors the condition.

1. Temperature sensors monitor key supervisor engine components including daughter cards.
2. A STATUS LED is located on the supervisor engine front panel and all module front panels.
3. The STATUS LED is red on the failed supervisor engine. If there is no redundant supervisor, the SYSTEM LED is red also.
4. See the "Understanding How Power Management Works" section on page 50-1 for instructions.

CHAPTER 51

Configuring Online Diagnostics

This chapter describes how to configure the online diagnostics on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

This chapter consists of these sections:

For descriptions of the online diagnostics tests, refer to Appendix A, "Online Diagnostic Tests."

Understanding How Online Diagnostics Work

With online diagnostics, you can test and verify the hardware functionality of the supervisor engine, modules, and 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, line card online insertion and removal (OIR), and system reset. The nondisruptive online diagnostic tests run as part of background health monitoring or 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, module OIR, or switchover to a backup supervisor engine; 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 51-2
- Configuring On-Demand Online Diagnostics, page 51-3
- Scheduling Online Diagnostics, page 51-4

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 for the supervisor engine 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.

Note The diagnostic level applies to the entire switch and cannot be configured on a per-module basis.

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. Run the memory tests on the other modules before running them on the supervisor engine.

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. Not all functional areas are present on each module. 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 that is only available on the supervisor engine. 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 not recover after the test is complete. The test takes several minutes to complete.

Disable all health-monitoring tests for the module being tested before running this test by using the **no diagnostic monitor module** *module* **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 for the module being tested before running this test by using the **no diagnostic monitor module** *module* **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.

All modules have exhaustive-memory tests available on them. Because the supervisor engine goes into an unusable state and must be rebooted after the exhaustive-memory tests, run the tests on all other modules first. Some of the exhaustive-memory tests can take several hours to complete because of the large memory size of the modules.

Before running the exhaustive-memory tests, all health-monitoring tests should be disabled on the module that will run the exhaustive-memory tests 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** *module* **test all** command.

Perform the exhaustive-memory tests in the following order (you can skip any tests not available for a particular module):

1. TestFibTcamSSRAM
2. TestAclQosTcam
3. TestNetFlowTcam
4. TestAsicMemory
5. TestAsicMemory

You must reboot the supervisor engine after running the exhaustive-memory tests before it is operational again. You cannot run any other tests on the supervisor engine or other modules after running the exhaustive-memory tests. Do not save the configuration when rebooting as it will have changed during the tests. You will need to power cycle the modules before they can be operational. After a module comes back on line, reenable the health-monitoring tests using the **diagnostic monitor module** *module* **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 for a specific module. 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** *num*} **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 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 module and port:

```
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 and module:

```
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 and module:

```
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 on specified modules while the switch is connected to a live network. You can configure the execution interval for each health-monitoring test, whether or not to generate a system message upon test failure, or to enable or disable 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** *num*} **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 for the specified module. The **no** form of this command will change the interval to the default interval, or zero.
Step 2	Router(config)#[**no**] **diagnostic monitor** {**module** *num*} **test** {*test_id* | *test_id_range* | **all**}	Enables or disables health-monitoring diagnostic tests.

This example shows how to configure the specified test to run every two minutes:

```
Router(config)# diagnostic monitor interval module 1 test 1 min 2
Router(config)#
```

This example shows how to run the test on the specified module 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 for each module 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 51-6
- Displaying Online Diagnostic Tests and Test Results, page 51-6

Starting and Stopping Online Diagnostic Tests

After you configure diagnostic tests to run on the switch or individual modules, 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
diagnostic start {**module** *num*} **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 specific module and port or range of ports.
diagnostic stop {**module** *num*}	Stops a diagnostic test on a specific module.

This example shows how to start a diagnostic test on a specific module:

```
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 on a specific module:

```
Router# diagnostic stop module 3
Router#
```

Displaying Online Diagnostic Tests and Test Results

You can display the online diagnostic tests that are configured for specific modules and check the results of the tests using the **show** commands.

To display the diagnostic tests that are configured for a module, perform this task:

Command	Purpose
show diagnostic content [**module** *num*]	Displays the online diagnostics configured for a module.

This example shows how to display the online diagnostics that are configured on a module:

```
Router# show diagnostic content module 7

Module 7:

  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 reset supervisor / NA
      K/* - Require resetting the line card after the test has completed / NA

                                                        Testing Interval
  ID   Test Name                            Attributes   (day hh:mm:ss.ms)
  ==== ================================== ============ =================
    1) TestScratchRegister -------------> ***N****A**  000 00:00:30.00
    2) TestSPRPInbandPing --------------> ***N****A**  000 00:00:15.00
    3) TestTransceiverIntegrity --------> **PD****I**  not configured
    4) TestActiveToStandbyLoopback -----> M*PDS***I**  not configured
    5) TestLoopback --------------------> M*PD****I**  not configured
    6) TestNewLearn --------------------> M**N****I**  not configured
    7) TestIndexLearn ------------------> M**N****I**  not configured
    8) TestDontLearn -------------------> M**N****I**  not configured
    9) TestConditionalLearn ------------> M**N****I**  not configured
   10) TestBadBpdu ---------------------> M**D****I**  not configured
   11) TestTrap ------------------------> M**D****I**  not configured
   12) TestMatch -----------------------> M**D****I**  not configured
   13) TestCapture ---------------------> M**D****I**  not configured
   14) TestProtocolMatch ---------------> M**D****I**  not configured
   15) TestChannel ---------------------> M**D****I**  not configured
   16) TestFibDevices ------------------> M**N****I**  not configured
   17) TestIPv4FibShortcut -------------> M**N****I**  not configured
   18) TestL3Capture2 ------------------> M**N****I**  not configured
   19) TestIPv6FibShortcut -------------> M**N****I**  not configured
   20) TestMPLSFibShortcut -------------> M**N****I**  not configured
   21) TestNATFibShortcut --------------> M**N****I**  not configured
   22) TestAclPermit -------------------> M**N****I**  not configured
   23) TestAclDeny ---------------------> M**D****I**  not configured
   24) TestQoSTcam ---------------------> M**D****I**  not configured
   25) TestL3VlanMet -------------------> M**N****I**  not configured
   26) TestIngressSpan -----------------> M**N****I**  not configured
   27) TestEgressSpan ------------------> M**N****I**  not configured
   28) TestNetflowInlineRewrite --------> C*PD****I**  not configured
   29) TestFabricSnakeForward ----------> M**N****I**  not configured
   30) TestFabricSnakeBackward ---------> M**N****I**  not configured
   31) TestFibTcamSSRAM ----------------> ***D****IR*  not configured
   32) ScheduleSwitchover --------------> ***D****I**  not configured
```

```
Router#
```

This example shows how to display the online diagnostic results for a module:

```
Router# show diagnostic result module 5
Current bootup diagnostic level:minimal

Module 5:

  Overall Diagnostic Result for Module 5 :PASS
  Diagnostic level at card bootup:minimal

  Test results:(. = Pass, F = Fail, U = Untested)

    1) TestScratchRegister -------------> .
    2) TestSPRPInbandPing --------------> .
    3) TestGBICIntegrity:

      Port  1  2
      ----------
            U  U


    4) TestActiveToStandbyLoopback:

      Port  1  2
      ----------
            U  U


    5) TestLoopback:

      Port  1  2
      ----------
            .  .


    6) TestNewLearn --------------------> .
    7) TestIndexLearn ------------------> .
    8) TestDontLearn -------------------> .
    9) TestConditionalLearn ------------> .
   10) TestBadBpdu ---------------------> .
   11) TestTrap ------------------------> .
   12) TestMatch -----------------------> .
   13) TestCapture ---------------------> .
   14) TestProtocolMatch ---------------> .
   15) TestChannel ---------------------> .
   16) TestIPv4FibShortcut -------------> .
   17) TestL3Capture2 ------------------> .
   18) TestL3VlanMet -------------------> .
   19) TestIngressSpan -----------------> .
   20) TestEgressSpan ------------------> .
   21) TestIPv6FibShortcut -------------> .
   22) TestMPLSFibShortcut -------------> .
   23) TestNATFibShortcut --------------> .
   24) TestAclPermit -------------------> .
   25) TestAclDeny ---------------------> .
   26) TestQoSTcam ---------------------> .
   27) TestNetflowInlineRewrite:

      Port  1  2
      ----------
            U  U
```

```
   28) TestFabricSnakeForward ----------> .
   29) TestFabricSnakeBackward ---------> .
   30) TestFibTcam - RESET -------------> U
Router#
```

This example shows how to display the detailed online diagnostic results for a module:

```
Router# show diagnostic result module 5 detail
Current bootup diagnostic level:minimal

Module 5:

  Overall Diagnostic Result for Module 5 :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#
```

Schedule Switchover

The schedule switchover test is used to check the readiness of the standby supervisor engine to take over in case the active supervisor engine fails or is taken out of service. You can run this test once or schedule it to run on a regular (daily, weekly, or monthly) basis.

Note When setting the time for a schedule switchover on both supervisor engines, the switchover for the active and standby supervisor engines should be scheduled at least 10 minutes apart to reduce system downtime if the switchover fails.

To configure a schedule switchover, perform this task:

	Command	Purpose
Step 1	**show diagnostic content** [**module** *num*]	Displays the online diagnostics configured for a module. Use this command to obtain the test ID for the schedule switchover.
Step 2	Router(config)# **diagnostic schedule module** {*num* | *active-sup-slot*} **test** {*test-id*} {**on** *mm dd yyyy hh:mm*} | {**daily** *hh:mm* } | {**weekly** *day-of-week hh:mm*}	Sets up the schedule switchover test for a specific date and time for the supervisor engine.

This example shows how to schedule a switchover for the active supervisor engine every Friday at 10:00 PM, and switch the standby supervisor engine back to the active supervisor engine 10 minutes after the scheduled switchover from the active supervisor engine occurs.

```
Router(config)# diagnostic schedule module 5 test 32 weekly Friday 22:00
Router(config)# diagnostic schedule module 6 test 32 weekly Friday 22:10
Router(config)#
```

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.
  - Remove all switching modules for testing FIB TCAM and SSRAM on the policy feature card (PFC3B) of the supervisor engine.
  - Reset the system or the module you are testing before returning the system to normal operating mode.

Note Turn off all background health-monitoring tests on the supervisor engine and switching modules using the **no diagnostic monitor module** *num* **test all** command.

CHAPTER 52

Using Top-N Reports

This chapter describes how to use Top-N reports on the Catalyst 6500 series switches.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

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 52-1.

Table 52-1 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 52-3
- Displaying Top-N Reports, page 52-3
- Clearing Top-N Reports, page 52-4

Enabling Top-N Reports Creation

To enable Top-N reports creation, perform this task:

Command	Purpose
Router# **collect top** [*number_of_ports*] **counters interface** {*interface_type*[1] | **all** | **layer-2** | **layer-3**} [**sort-by** *statistic_type*[2]] [**interval** *seconds*]	Enables Top-N reports creation.

1. *interface_type* = **ethernet**, **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
```

CHAPTER 53

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, refer to the *Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference*, Release 12.2ZY, at this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html

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.

Note For more information about CDP, see Chapter 44, "Configuring CDP."

- 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.
- 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#
```

APPENDIX A

Online Diagnostic Tests

This appendix describes the online diagnostic tests and provides recommendations for how to use them.

The online diagnostic tests are included in these categories:

For information about configuring online diagnostic tests refer to Chapter 51, "Configuring Online Diagnostics."

Global Health-Monitoring Tests

These are the global health monitoring tests:

TestSPRPInbandPing

The TestSPRPIinbandPing 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.

Table A-1 TestSPRPInbandPing Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not disable. Test is automatically disabled during CPU-usage spikes in order to maintain accuracy.
Default	On.
Release	12.2(18)ZY.
Corrective action	Reset the active supervisor engine.
Hardware support	Active and standby supervisor engine.

TestSPNPInbandPing

The TestSPNPIinbandPing test verifies the data path between the switch processor (SP) and the network processor (NP). This test sends a Layer 2 frame from the SP inband port and to the NP and the NP loops it back to the SP inband port. This test runs every 15 seconds. 10 consecutive test failures cause the NP to reset. After three consecutive NP resets, 10 consecutive test failures results in a supervisor engine switchover, or in a reload if a redundant supervisor engine is not installed.

Table A-2 TestSPNPInbandPing Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not disable. Test is automatically disabled during CPU-usage spikes in order to maintain accuracy.
Default	On.
Release	12.2(18)ZY2.
Corrective action	Reset the active supervisor engine.
Hardware support	Active and standby supervisor engine.

TestScratchRegister

The TestScratchRegister 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.

Table A-3 TestScratchRegister Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not disable.
Default	On.
Release	12.2(18)ZY.
Corrective action	Reset the malfunctioning supervisor engine or power down the module.
Hardware support	Supervisor Engine 32, WS-X6148-FE-SFP, WS-X6148A-GE-TX, and WS-X6148A-RJ-45 .

Per-Port Tests

The per-port tests consist of the following tests:

- TestNonDisruptiveLoopback, page A-3
- TestLoopback, page A-4
- TestActiveToStandbyLoopback, page A-4
- TestTransceiverIntegrity, page A-5
- TestNetflowInlineRewrite, page A-5

TestNonDisruptiveLoopback

The TestNonDisruptiveLoopback test verifies the data path between the supervisor engine and the network ports of a module. In this test, a Layer2 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.

Table A-4 TestNonDisruptiveLoopback Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not disable.
Default	On.

Table A-4 TestNonDisruptiveLoopback Test Attributes (continued)

Release	12.2(18)ZY.
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 and WS-X6148A-RJ-45.

TestLoopback

The TestLoopback 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.

Table A-5 TestLoopback Test Attributes

Attribute	Description
Disruptive/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).
Release	12.2(18)ZY.
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.

TestActiveToStandbyLoopback

The TestActiveToStandbyLoopback 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 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 engines's inband port listening in on the flooded VLAN.

Table A-6 TestActiveToStandbyLoopback Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive for looped-back ports. Disruption is typically less than one second. Duration of the disruption depends on the configuration of loopback port (for example, Spanning Tree Protocol.).

Table A-6 TestActiveToStandbyLoopback Test Attributes (continued)

Recommendation	Schedule during downtime.
Default	Runs at bootup or after OIR.
Release	12.2(18)ZY.
Corrective action	Error disable a port if the loopback test fails on the port. Reset the supervisor engine if all of the ports fail.
Hardware support	Standby supervisor engine only.

TestTransceiverIntegrity

The TestTransceiverIntegrity test is a security test performed on the transceiver during transceiver online insertion and removal (OIR) or module bootup to make sure that the transceiver is supported.

Table A-7 TestTransceiverIntegrity Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Not applicable.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(18)ZY.
Corrective action	Error disable the port.
Hardware support	All modules with transceivers.

TestNetflowInlineRewrite

The TestNetflowInlineRewrite 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.

Table A-8 TestNetflowInlineRewrite Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	All modules including supervisor engines.

PFC Layer 2 Forwarding Engine Tests

The PFC Layer 2 Forwarding Engine tests consist of the following tests:

TestNewIndexLearn

The TestNewIndexLearn test verifies the Layer 2 source MAC address learning functionality and the Index Learn feature of the Layer 2 forwarding engine and ensures that existing MAC address table entries can be updated. 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. The Layer 2 learning functionality is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-9 TestNewIndexLearn Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines only.

TestDontConditionalLearn

The TestDontConditionalLearn test is a combination of the TestDontLearn and the TestConditionalLearn tests.

The TestDontLearn 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. The “don't learn” feature is verified during diagnostic packet lookup by the Layer 2 forwarding engine.

The TestConditionalLearn 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. The Conditional Learn feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-10 TestDontConditionalLearn Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines only.

TestBadBpduTrap

The TestBadBpduTrap test is a combination of the TestTrap and the TestBadBpdu tests.

The TestTrap 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. The Trap feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

The TestBadBpdu 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. The BPDU feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-11 TestBadBpduTrap Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive.

Table A-11 TestBadBpduTrap Test Attributes (continued)

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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines only.

TestMatchCapture

The TestMatchCapture test is a combination of the TestProtocolMatchChannel and the TestCapture tests.

The TestProtocolMatchChannel 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. The Match feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

The TestCapture 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. The Capture feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-12 TestMatchCapture Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive.
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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines only.

TestStaticEntry

The TestStaticEntry test verifies that static entries are populated in the Layer 2 MAC address table. This functionality is verified during diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-13 TestStaticEntry Test Attributes

Attribute	Description
Disruptive/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	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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

PFC Layer 3 Forwarding Engine Tests

These are the PFC Layer 3 Forwarding Engine tests:

TestFibDevices

The TestFibDevices 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.

Table A-14 TestFibDevices Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines.

TestIPv4FibShortcut

The TestIPv4FibShortcut test verifies the IPV4 FIB forwarding of the Layer 3 forwarding engine is working properly. One diagnostic IPV4 FIB and 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.

Table A-15 TestIPv4FibShortcut Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines.

TestIPv6FibShortcut

The TestIPv6FibShortcut test verifies that the IPV6 FIB forwarding of the Layer 3 forwarding engine is working properly. One diagnostic IPV6 FIB and 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.

Table A-16 TestIPv6FibShortcut Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines.

TestMPLSFibShortcut

The TestMPLSFibShortcut test verifies that the MPLS forwarding of the Layer 3 forwarding engine is working properly. One diagnostic MPLS FIB and 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.

Table A-17 TestMPLSFibShortcut Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines.

TestNATFibShortcut

The TestNATFibShortcut test verifies the ability to rewrite a packet based on the NAT adjacency information (rewrite destination IP address). One diagnostic NAT FIB and 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.

Table A-18 TestNATFibShortcut Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines.

TestL3Capture2

The TestL3Capture2 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 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.

Table A-19 TestL3Capture2 Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines.

TestAclPermit

The TestAclPermit 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.

Table A-20 TestACLPermit Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	This test can also be used as a health-monitoring test. Use as a health-monitoring test if you are using ACLs.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines.

TestAclDeny

The TestAclDeny 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.

Table A-21 TestACLDeny Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive.
Recommendation	Do not disable.
Default	On.
Release	12.2(18)ZY.
Corrective action	Automatic ASIC reset for recovery.
Hardware support	Supervisor engines.

TestNetflowShortcut

The TestNetflowShortcut test verifies that the NetFlow forwarding functionality of the Layer 3 forwarding engine is working properly. One diagnostic NetFlow entry and adjacency entry is installed, and a diagnostic packet is sent to make sure it is forwarded according to the rewritten MAC and VLAN information.

Table A-22 TestNetflowShortcut Test Attributes

Attributes	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

TestQoS

The TestQoS test verifies whether or not the QoS input and output TCAM is functional by programming the QoS input and output TCAM so that the ToS value of the diagnostic packet is changed to reflect either input or output.

Table A-23 TestQoS Test Attributes

Attributes	Description
Disruptive/Nondisruptive	Disruptive for looped back ports. The disruption is 500 ms.
Recommendation	Schedule during downtime.
Default	This test runs by default during bootup or after a reset or OIR
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

Replication Engine Tests

These are the Replication Engine tests:

- TestL3VlanMet, page A-15
- TestIngressSpan, page A-15
- TestEgressSpan, page A-16

TestL3VlanMet

The TestL3VlanMet 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.

Table A-24 TestL3VlanMet Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and WS-65xx, WS-67xx, and WS-68xx modules.

TestIngressSpan

The TestIngressSpan 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.

Table A-25 TestIngressSpan Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and WS-65xx and WS-67xx modules.

TestEgressSpan

The TestEgressSpan test verifies that the egress SPAN replication functionality of the rewrite engine for both SPAN queues is working properly.

Table A-26 TestEgressSpan Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and WS-65xx and WS-67xx modules.

Exhaustive Memory Tests

These are the exhaustive memory tests:

- TestFibTcamSSRAM, page A-16
- TestAsicMemory, page A-17
- TestAclQosTcam, page A-17
- TestNetflowTcam, page A-18
- TestQoSTcam, page A-18

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 51-3.

TestFibTcamSSRAM

The TestFibTcamSSRAM test checks the FIB TCAM and Layer 3 Adjacency SSRAM memory.

Table A-27 TestFibTcamSSRAM Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive. Disruption is several hours.

Table A-27 TestFibTcamSSRAM Test Attributes (continued)

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.
Release	12.2(18)ZY
Corrective action	Not applicable.
Hardware support	All modules including supervisor engines.

TestAsicMemory

The TestAsicMemory test uses an algorithm to test the memory on a module.

Table A-28 TestAsicMemory Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	Not applicable.
Hardware support	All modules including supervisor engines.

TestAclQosTcam

The TestAclQosTcam test tests all the bits and checks the location of both ACL and QOS TCAMs on the PFC3B.

Table A-29 TestAclQosTcam Test Attributes

Attribute	Description
Disruptive/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.

Table A-29 TestAclQosTcam Test Attributes (continued)

Default	Off.
Release	12.2(18)ZY.
Corrective action	Not applicable.
Hardware support	All modules including supervisor engines.

TestNetflowTcam

The TestNetflowTcam test tests all the bits and checks the location of the Netflow TCAM.

Table A-30 TestNetflowTcam Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive. Disruption is several minutes.
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.
Release	12.2(18)ZY.
Corrective action	Not applicable.
Hardware support	All modules including supervisor engines.

TestQoSTcam

The TestQoSTcam test performs exhaustive memory tests for QoS TCAM devices.

Table A-31 TestQoSTcam Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive. Disruption is several minutes.
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.
Release	12.2(18)ZY.
Corrective action	Not applicable.
Hardware support	All modules including supervisor engines.

IPSEC Services Modules Tests

These are the IPsec services modules tests:

- TestIPSecClearPkt, page A-19
- TestHapiEchoPkt, page A-19
- TestIPSecEncryptDecryptPkt, page A-20

TestIPSecClearPkt

The TestIPSecClearPkt 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.

Table A-32 TestIPSecClearPkt Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Run this test on-demand.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	VPN service module.

TestHapiEchoPkt

The TestHapiEchoPkt 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.

Table A-33 TestHapiEchoPkt Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive.
Recommendation	Run this test on-demand. This test cannot be run from on-demand CLI.
Default	On.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	VPN service module.

TestIPSecEncryptDecryptPkt

The TestIPSecEncryptDecryptPkt 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.

Table A-34 TestIPSecEncryptDecryptPkt Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY.
Corrective action	None. See the system message guide for more information.
Hardware support	VPN services module.

Stress Tests

These are the stress tests:

- TestTrafficStress, page A-20
- TestEobcStressPing, page A-21

TestTrafficStress

The TestTrafficStress 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.

Table A-35 TestTrafficStress Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive. Disruption is several minutes.
Recommendation	Use this test to qualify hardware before installing it in your network.
Default	Off.
Release	12.2(18)ZY.
Corrective action	Not applicable.
Hardware support	PFC3.

TestEobcStressPing

The TestEobcStressPing test stress tests 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.

Table A-36 TestEobcStressPing Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive. Disruption is several minutes.
Recommendation	Use this test to qualify hardware before installing it in your network.
Default	Off.
Release	12.2(18)ZY.
Corrective action	Not applicable.
Hardware support	PFC3.

Critical Recovery Test—TestL3HealthMonitoring

The TestL3HealthMonitoring test triggers a set of diagnostic tests involving IPv4 and IPv6 packet switching on a local 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

Table A-37 TestL3HealthMonitoring Test Attributes

Attribute	Description
Disruptive/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	On.
Release	12.2(18)ZY.
Corrective action	Not applicable.
Hardware support	DFC-equipped modules

General Tests

These are the general tests:

- ScheduleSwitchover, page A-22
- TestFirmwareDiagStatus, page A-22

ScheduleSwitchover

The ScheduleSwitchover test allows you to trigger a switchover at any time using the online diagnostics scheduling capability.

Table A-38 ScheduleSwitchover Test Attributes

Attribute	Description
Disruptive/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.
Release	12.2(18)ZY
Corrective action	None
Hardware support	Supervisor engines only.

TestFirmwareDiagStatus

The TestFirmwareDiagStatus test displays the results of the power-on diagnostic tests run by the firmware during the module bootup.

Table A-39 TestFirmwareDiagStatus Test Attributes

Attribute	Description
Disruptive/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
Release	12.2(18)ZY
Corrective action	None. See the system message guide.
Hardware support	All modules, including supervisor engines.

APPENDIX B

Acronyms

Table B-1 defines the acronyms used in this publication.

Table B-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 B-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 B-1 List of Acronyms (continued)

Acronym	Expansion
DTP	Dynamic Trunking Protocol
DTR	data terminal ready
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 B-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
MFD	multicast fast drop
MIB	Management Information Base
MII	media-independent interface
MLS	Multilayer Switching
MLSE	maintenance loop signaling entity
MOP	Maintenance Operation Protocol
MOTD	message-of-the-day
MLSE	maintenance loops signaling entity
MRM	multicast routing monitor
MSDP	Multicast Source Discovery Protocol
MST	multiple spanning tree
MTU	maximum transmission unit
MVAP	multiple VLAN access port
NAM	Network Analysis Module
NBP	Name Binding Protocol
NCIA	Native Client Interface Architecture

Table B-1 List of Acronyms (continued)

Acronym	Expansion
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
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
PEP	policy enforcement point
PFC	Policy Feature Card
PGM	Pragmatic General Multicast
PHY	physical sublayer
PIB	policy information base
PIM	protocol independent multicast
PISA	Programmable Intelligent Services Accelerator
PPP	Point-to-Point Protocol
PRID	Policy Rule Identifiers
PVST+	Per VLAN Spanning Tree+
QDM	QoS device manager
QM	QoS manager
QoS	quality of service
RACL	router interface access control list

Table B-1 List of Acronyms (continued)

Acronym	Expansion
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
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
SRM	single router mode
SSO	stateful switchover

Table B-1 List of Acronyms (continued)

Acronym	Expansion
SPAN	Switched Port Analyzer
SREC	S-Record format, Motorola defined format for ROM contents
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
TopN	Utility that allows the user to analyze port traffic by reports
TOS	type of service
TLV	type-length-value
TTL	Time To Live
TVX	valid transmission
UDLD	UniDirectional Link Detection Protocol
UDP	User Datagram Protocol
UNI	User-Network Interface
UTC	Coordinated Universal Time
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

Table B-1 List of Acronyms (continued)

Acronym	Expansion
WRED	weighted random early detection
WRR	weighted round-robin
XNS	Xerox Network System

INDEX

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X