Cisco 7600 Series Router
Cisco IOS Software Configuration Guide

October 2010

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-10113-10

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: The equipment described in this manual generates and may radiate radio-frequency energy. If it is not installed in accordance with Cisco's installation instructions, it may cause interference with radio and television reception. This equipment has been tested and found to comply with the limits for a Class B digital device in accordance with the specifications in part 15 of the FCC rules. These specifications are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

Modifying the equipment without Cisco's written authorization may result in the equipment no longer complying with FCC requirements for Class A or Class B digital devices. In that event, your right to use the equipment may be limited by FCC regulations, and you may be required to correct any interference to radio or television communications at your own expense.

You can determine whether your equipment is causing interference by turning it off. If the interference stops, it was probably caused by the Cisco equipment or one of its peripheral devices. If the equipment causes interference to radio or television reception, try to correct the interference by using one or more of the following measures:

• Turn the television or radio antenna until the interference stops.

• Move the equipment to one side or the other of the television or radio.

• Move the equipment farther away from the television or radio.

• Plug the equipment into an outlet that is on a different circuit from the television or radio. (That is, make certain the equipment and the television or radio are on circuits controlled by different circuit breakers or fuses.)

Modifications to this product not authorized by Cisco Systems, Inc. could void the FCC approval and negate your authority to operate the product.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Cisco 7600 Series Router Cisco IOS Software Configuration Guide, Release 12.2SR

CONTENTS

Preface

This preface describes who should read the *Cisco 7600 Series Router Cisco IOS Software Configuration Guide*, Release 12.2SR, how it is organized, and its document conventions.

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining Cisco 7600 series routers.

Document Revision History

The Document Revision History records technical changes to this document. The table shows the Cisco IOS software release number and document revision number for the change, the date of the change, and a brief summary of the change.

Release No.	Revision	Date	Change Summary
12.2(33)SRD5	OL-10113-10	October 2010	• Updated Configuring PFC QoS, Configuring Denial of Service Protection, IP Subscriber Awareness over Ethernet, and Configuring Traffic Storm Control. • Added troubleshooting information for Multicast in Chapter 28, "Troubleshooting.", Chapter 25, "Troubleshooting.", Chapter 30, "Troubleshooting." • Added troubleshooting information for MPLS VPN in Chapter 24, "Troubleshooting.". • Added troubleshooting information for MLS QoS in Configuring PFC QoS. • Updated the Configuring Denial of Service Protection restrictions.
12.2(33)SRE1	OL-10113-09	May 2010	• Updated the section Configuring LACP 1:1 Redundancy with Fast-Switchover and updated the table under Configuring VLANs in chapter 14.
12.2(33)SRE1	OL-10113-09	April 2010	• Added Usage Guidelines to Configure Protocol Flow Filter in Chapter 47, "Configuring NetFlow and NDE."
12.2(33)SRE1	OL-10113-09	April 2010	• Extended support for Private Host for VPLS in Chapter 35, "Configuring Private Hosts,"and Bridged Routing Encapsulation on Automatic Protection Service Group in in Chapter 14, "Configuring VLANs".

Release No.	Revision	Date	Change Summary
12.2(33)SRD4	OL-10113-08	February 2010	Support for the following features were introduced: • Private Host for VPLS in Chapter 35, "Configuring Private Hosts." • Bridged Routing Encapsulation on Automatic Protection Service Group in Chapter 14, "Configuring VLANs" • Restrictions on UDLD configurations • Updated a new command **mls qos recirc untrust**.
12.2(33)SRE	OL-10113-07	February 2010	Added a new section on Scalable EoMPLS and Port-mode EoMPLS and corresponding sample configurations in Chapter 24, "Configuring Multiprotocol Label Switching on the PFC,".
12.2(33)SRE	OL-10113-07	January 2010	Updated the NSF Benefits and Restrictions section in the chapter Configuring NSF with SSO Supervisor Engine Redundancy.
12.2(33)SRE	OL-10113-07	December 2009	Updated the priority-queue queue-limit interface command under Configuring PFC QoS chapter.
12.2(33)SRE	OL-10113-07	November 2009	Support for the following features were introduced: • IPv4 Multicast Support on Multicast Forwarding Information Base (MFIB) • Multicast MFIB Bi-Dir, MVPN and P2P • ISSU - IPv4 Multicast • ISSU for IP Multicast
12.2(33)SRD2	OL-10113-06	May 2009	Added support for ARP Scale to 512k.

Release No.	Revision	Date	Change Summary
12.2(33)SRD	OL-10113-05	October 2008	Added chapter on Using the Mini Protocol Analyzer
12.2(33)SRC	OL-10113-04	December 31, 2007	Support for the following features was introduced: • IEEE 802.1x with DHCP • VTP v3 • LACP 1-1 redundancy with fast switchover • Switch Port Analyzer (SPAN)—Input packets with don't learn option • SPAN egress session increase • SPAN destination port support on Etherchannels • Call Home

Organization

This guide is organized as follows:

Chapter	Title	Description
Chapter 1	Product Overview	Presents an overview of the Cisco 7600 series routers.
Chapter 2	Configuring the Router for the First Time	Describes how to configure the router for the first time.
Chapter 3	Configuring a Supervisor Engine 720	Describes how to configure a Supervisor Engine 720.
Chapter 4	Configuring a Route Switch Processor 720	Describes how to configure the Route Switch Processor 720 (RSP720).
Chapter 5	Configuring NSF with SSO Supervisor Engine Redundancy	Describes how to configure NSF with SSO supervisor engine redundancy.
Chapter 6	ISSU and eFSU on Cisco 7600 Series Routers	Describes the In Service Software Upgrade (ISSU) and enhanced Fast Software Upgrade (eFSU) processes, which enable you to upgrade Cisco IOS software while the router is running.
Chapter 7	Configuring RPR and RPR+ Supervisor Engine Redundancy	Describes how to configure RPR and RPR+ supervisor engine redundancy.
Chapter 8	Configuring Interfaces	Describes how to configure non-layer-specific features on LAN interfaces.
Chapter 9	Configuring a Supervisor Engine 32	Describes how to configure Supervisor Engine 32.

Chapter	Title	Description
Chapter 10	Configuring LAN Ports for Layer 2 Switching	Describes how to configure LAN interfaces to support Layer 2 features, including VLAN trunks.
Chapter 11	Configuring Flex Links	Describes how to configure Flex Links.
Chapter 12	Configuring EtherChannels	Describes how to configure Layer 2 and Layer 3 EtherChannel port bundles.
Chapter 13	Configuring VTP	Describes how to configure the VLAN Trunking Protocol (VTP).
Chapter 14	Configuring VLANs	Describes how to configure VLANs.
Chapter 15	Configuring Private VLANs	Describes how to configure private VLANs.
Chapter 16	Configuring Cisco IP Phone Support	Describes how to configure Cisco IP Phone support.
Chapter 17	Configuring IEEE 802.1Q Tunneling	Describes how to configure IEEE 802.1Q tunneling.
Chapter 18	Configuring Layer 2 Protocol Tunneling	Describes how to configure Layer 2 protocol tunneling.
Chapter 19	Configuring STP and MST	Describes how to configure the Spanning Tree Protocol (STP) and Prestandard IEEE 802.1s Multiple Spanning Tree (MST).
Chapter 20	Configuring Optional STP Features	Describes how to configure optional STP features.
Chapter 21	Configuring Layer 3 Interfaces	Describes how to configure LAN interfaces to support Layer 3 features.
Chapter 22	IP Subscriber Awareness over Ethernet	Describes how to configure the IP Subscriber Awareness over Ethernet feature, which provides IP session termination and aggregation on the router.
Chapter 23	Configuring UDE and UDLR	Describes how to configure unidirectional Ethernet (UDE) and unidirectional link routing (UDLR).
Chapter 24	Configuring Multiprotocol Label Switching on the PFC	Describes how to configure Multiprotocol Label Switching (MPLS) on the PFC.
Chapter 25	Configuring IPv4 Multicast VPN Support	Describes how to configure IPv4 Multicast Virtual Private Network (MVPN).
Chapter 26	Configuring IP Unicast Layer 3 Switching	Describes how to configure IP unicast Layer 3 switching.
Chapter 27	Configuring IPv6 Multicast PFC3 and DFC3 Layer 3 Switching	Describes how to configure IPv6 Multicast Multilayer Switching (MMLS).
Chapter 28	Configuring IPv4 Multicast Layer 3 Switching	Describes how to configure IPv4 Multicast Multilayer Switching (MMLS).
Chapter 29	Configuring MLDv2 Snooping for IPv6 Multicast Traffic	Describes how to configure Multicast Listener Discovery version 2 (MLDv2) snooping.
Chapter 30	Configuring IGMP Snooping for IPv4 Multicast Traffic	Describes how to configure Internet Group Management Protocol (IGMP) snooping.
Chapter 31	Configuring PIM Snooping	Describes how to configure protocol independent multicast (PIM) snooping.

Chapter	Title	Description
Chapter 32	Configuring Network Security	Describes how to configure network security features that are unique to the Cisco 7600 series routers.
Chapter 33	Understanding Cisco IOS ACL Support	Describes how Cisco 7600 series routers support Cisco IOS ACLs.
Chapter 34	Configuring VLAN ACLs	Describes how to configure VLAN ACLs.
Chapter 35	Private Hosts (Using PACLs)	Describes how to use port-based ACLs to configure the Private Hosts feature, which provides Layer 2 isolation between hosts on the same VLAN.
Chapter 39	Configuring Denial of Service Protection	Describes how to configure denial of service protection.
Chapter 37	Configuring DHCP Snooping	Describes how to configure DHCP snooping.
Chapter 38	Configuring Dynamic ARP Inspection	Describes how to configure dynamic ARP inspection.
Chapter 39	Configuring Traffic Storm Control	Describes how to configure traffic storm control.
Chapter 40	Unknown Unicast Flood Blocking	Describes how to configure unknown unicast flood blocking.
Chapter 44	Configuring PFC QoS	Describes how to configure quality of service (QoS).
Chapter 43	Configuring MPLS QoS on the PFC	Describes how to configure MPLS QoS.
Chapter 42	Configuring PFC QoS Statistics Data Export	Describes how to configure PFC QoS statistics data export.
Chapter 44	Configuring IEEE 802.1X Port-Based Authentication	Describes how to configure IEEE 802.1X port-based authentication.
Chapter 45	Configuring Port Security	Describes how to configure port security.
Chapter 46	Configuring UDLD	Describes how to configure the UniDirectional Link Detection (UDLD) protocol.
Chapter 47	Configuring NetFlow and NDE	Describes how to configure NetFlow statistics collection and NetFlow Data Export (NDE).
Chapter 48	Configuring Local SPAN, RSPAN, and ERSPAN	Describes how to configure the Switch Port Analyzer (SPAN).
Chapter 49	Configuring SNMP IfIndex Persistence	Describes how to configure SNMP ifIndex persistence.
Chapter 50	Power Management and Environmental Monitoring	Describes how to configure power management and environmental monitoring features.
Chapter 51	Configuring Online Diagnostics	Describes how to configure online diagnostics and run diagnostic tests.
Chapter 52	Configuring Web Cache Services Using WCCP	Describes how to configure the Web Cache Communication Protocol (WCCP).
Chapter 53	Using the Top N Utility	Describes how to use the Top N utility.
Chapter 54	Using the Layer 2 Traceroute Utility	Describes how to use the Layer 2 traceroute utility.

Chapter	Title	Description
Chapter 55	Configuring Call Home	Describes how to configure the Call Home feature.
Chapter 56	Using the Mini Protocol Analyzer	Describes how to configure the Mini Protocol Analyzer.
Appendix A	Online Diagnostic Tests	Provides recommendations for how to use the online diagnostic tests.
Appendix B	Acronyms	Defines the acronyms used in this publication.

Related Documentation

The following publications are available for the Cisco 7600 series routers:

- *Cisco 7600 Series Router Installation Guide*
- *Cisco 7600 Series Router Module Installation Guide*
- *Cisco 7600 Series Router Cisco IOS Command Reference*
- *Cisco 7600 Series Router Cisco IOS System Message Guide*
- *Release Notes for Cisco IOS Release 12.2SX on the Supervisor Engine 720, Supervisor Engine 32, and Supervisor Engine 2*
- *Cisco IOS Configuration Guides and Command References*—Use these publications to help you configure Cisco IOS software features not described in the Cisco 7600 series router 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/univercd/cc/td/doc/product/software/ios122/index.htm

- 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, Obtaining Support, and Security Guidelines

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

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

CHAPTER 1

Product Overview

This chapter contains these sections:

Supported Hardware and Software

For complete information about the chassis, modules, and software features supported by Cisco 7600 series routers, refer to the *Release Notes for Cisco IOS Release 12.2SX on the Supervisor Engine 720, Supervisor Engine 32, and Supervisor Engine 2*.

See Appendix C, "Cisco IOS Release 12.2SRB Software Images," for information about the Cisco IOS software images available for this release.

Note In Cisco IOS Release 12.2SR and later releases, the Supervisor Engine 2, policy feature card 2 (PFC2), and FlexWAN module are no longer supported on Cisco 7600 series routers.

User Interfaces

Release 12.2SR supports configuration using the following interfaces:

- CLI—Refer to "Using the Command-Line Interface" in the Release 12.2 Cisco IOS *Configuration Fundamentals Configuration Guide* at this URL:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/12_2sr/cf_12_2sr_book.html

- SNMP—Refer to the Release 12.2 Cisco IOS *Configuration Fundamentals Configuration Guide* and *Command Reference* documents at this URL:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/12_2sr/cf_12_2sr_book.html

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

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/12_2sr/cf_12_2sr_book.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 router and to provide a GUI-based management and configuration interface. You can download the Java Archive (JAR) files for Embedded CiscoView at:

http://www.cisco.com/kobayashi/sw-center/netmgmt/ciscoview/embed-cview-planner.shtml

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 router web page is the enable-level password of the router.

For more information about web access to the router, refer to "Using the Cisco Web Browser" in the Cisco IOS *Configuration Fundamentals Configuration Guide* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/ffcprt1/fcf005.htm

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 PFC and DFC

These sections describe the hardware support provided by Policy Feature Card 3 (PFC3), Distributed Forwarding Card 3 (DFC3), and Distributed Forwarding Card (DFC):

- Software Features Supported in Hardware by the PFC3, DFC3, and DFC, page 1-3
- Software Features Supported in Hardware by the PFC3 and DFC3, page 1-4

Software Features Supported in Hardware by the PFC3, DFC3, and DFC

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

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

    Note Flows that require ACL logging are processed in software on the MSFC.

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

    Note Idle timeout is processed in software on the MSFC.

  For more information about PFC and DFC support for ACLs, see Chapter 33, "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/fsecur_c.html

- VLAN ACLs (VACLs)—To configure VACLs, see Chapter 34, "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" and "Configuring Policy-Based Routing," at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt1/qcfpbr.htm

  Note If the MSFC3 or MSFC4 address falls within the range of a PBR ACL, traffic addressed to the MSFC is policy routed in hardware instead of being forwarded to the MSFC. To prevent policy routing of traffic addressed to a MSFC3 or MSFC4, configure PBR ACLs to deny traffic addressed to the MSFC.

- Except on MPLS interfaces, TCP intercept—To configure TCP intercept, see the "Configuring TCP Intercept" section on page 32-2.
- Hardware-assisted NetFlow Aggregation—Refer to this URL:

  http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SXF/native/configuration/guide/nde.html

Software Features Supported in Hardware by the PFC3 and DFC3

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

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

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/finter_c/icflogin.htm
- Bidirectional Protocol Independent Multicast (PIM) in hardware—See the "Understanding How IPv4 Bidirectional PIM Works" section on page 28-6.
- Multiple-path Unicast Reverse Path Forwarding (RPF) Check—To configure Unicast RPF Check, see the "Configuring Unicast Reverse Path Forwarding Check" section on page 32-2.
- Except on MPLS interfaces, Network Address Translation (NAT) for IPv4 unicast and multicast traffic.

  Note the following information about hardware-assisted NAT:

  - NAT of UDP traffic is not supported in PFC3A mode.
  - The PFC3 does not support NAT of multicast traffic.
  - The PFC3 does not support NAT configured with a route-map that specifies length.
  - When you configure NAT and NDE on an interface, the PFC3 sends all traffic in fragmented packets to the MSFC3 or MSFC4 to be processed in software. (CSCdz51590)

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

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

  To prevent a significant volume of NAT traffic from being sent to the MSFC3, 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/products/hw/switches/ps700/prod_command_reference_list.html
- The PFC3 and DFC3 support IPv4 multicast over point-to-point GRE tunnels in hardware.

- GRE Tunneling and IP in IP Tunneling—The PFC3 and DFC3 support the following **tunnel** commands:
  - **tunnel destination**
  - **tunnel mode gre**
  - **tunnel mode ipip**
  - **tunnel source**
  - **tunnel ttl**
  - **tunnel tos**

  The MSFC3 and MSFC4 support tunneling configured with any other **tunnel** commands.

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

  The **tunnel tos** command 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/univercd/cc/td/doc/product/software/ios122/122cgcr/finter_c/icflogin.htm

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

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

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s17/12s_tos.htm

  Note the following information about tunnels:

  - Each hardware-assisted tunnel must have a unique source. Hardware-assisted tunnels cannot share a source even if the destinations are different. Use secondary addresses on loopback interfaces or create multiple loopback interfaces. Failure to use unique source addresses may result in control plane failures during software path congestion.
  - Each tunnel interface uses one internal VLAN.
  - Each tunnel interface uses one additional router MAC address entry per router MAC address.
  - The PFC3A does not support any PFC QoS features on tunnel interfaces. All other PFCs do.
  - The MSFC3 and MSFC4 support 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

Configuring the Router for the First Time

This chapter contains information about how to initially configure the Cisco 7600 series router. The information in this chapter 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 *Cisco 7600 Series Router Cisco IOS Command Reference* at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Default Configuration

Table 2-1 shows the default configuration.

Table 2-1 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 Router

These sections describe how to configure the router:

Using the Setup Facility or the setup Command

At initial startup, the router automatically defaults to the setup facility. You can also invoke the setup facility by entering the **setup** command at the enable prompt (#).

The setup facility provides a System Configuration Dialog, which is an interactive CLI mode that guides you through first-time configuration of the router. The dialog prompts you for the information needed to start your router functioning in the network.

The System Configuration Dialog first prompts you to configure global parameters, which are used to control system-wide settings. The dialog then prompts for information to configure interfaces. You must progress through the System Configuration Dialog until you reach an item you want to change.

As you move through the dialog, square brackets beside each prompt show the default setting for that item or the last configured value. To accept the default value for an item, press **Return** or **Enter**. To change the value for that item, enter the desired value.

To display help for a prompt, press the question mark (**?**) key at the prompt.

When you complete your changes, the system automatically displays the configuration file that was created during the setup session. The dialog asks if you want to use this configuration. If you answer Yes, the configuration is saved to NVRAM as the startup configuration file. If you answer No, the configuration is not saved and the process begins again.

To exit setup and return to privileged EXEC mode without making changes and without progressing through the entire dialog, press **Ctrl-C**.

When you complete the configuration process, 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.

Note You can use the **show version** command to check the current state of the router.

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

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

Using Configuration Mode

If you prefer not to use the setup facility, you can configure the router 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 2-5.)

Your router 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 router 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 router 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 default-gateway 172.20.52.35
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 router 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 default-gateway 172.20.52.35
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 router boots, it automatically retrieves the IP address from the BOOTP server.

The router 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 router or a router that has had its startup-config file cleared using the **erase** command.)

To allow your router to retrieve its IP address from a BOOTP server, you must first determine the MAC address of the router 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 router. 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

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 2-12.

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 router 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 Cisco 7600 series router 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 2-12 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 2-12.

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 2-12.

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/fsecur_c.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 router 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 router after you lose or forget the encrypted password. See the "Recovering a Lost Enable Password" section on page 2-12 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 2-12.

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, page 2-11

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 2-12.

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 2-12.

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 router 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 router 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, page 2-13
- Configuring the Software Configuration Register, page 2-14

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 router 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 router 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 2-15. The BOOT environment variable is described in the "Specifying the Startup System Image" section on page 2-17.

Understanding the ROM Monitor

The ROM monitor executes upon power-up, reset, or when a fatal exception occurs. The router enters ROM-monitor mode if the router 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 *Cisco 7600 Series Router Cisco IOS Command Reference* 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 router 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 2-2 lists the meaning of each of the software configuration memory bits, and Table 2-3 defines the boot field.

Caution The recommended configuration register setting is 0x2102 (this is the factory default value). If you configure a setting that leaves break enabled and you send a break sequence over a console connection, the router drops into ROMMON.

Table 2-2 Software Configuration Register Bit Meaning

Bit Number	Hexadecimal	Meaning
00 to 03	0x0000 to 0x000F	Boot field (see Table 2-3)
06	0x0040	Causes system software to ignore NVRAM contents
07	0x0080	OEM[1] 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. OEM = original equipment manufacturer.

Table 2-3 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 router 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 router 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.

	Command	Purpose
Step 3	`Router(config)# config-register value`	Modifies the existing configuration register setting to reflect the way in which you want the router 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 router is running Cisco IOS software, 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 2-2 on page 2-14).

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 2-5. 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 router 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 router 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-bootflash:**, **disk0:**, or **disk1:** device (only the Route Switch Processor 720 and Supervisor Engine 720 have **disk1:**).
- Do not store the system software image in the **bootflash:** device, which is on the MSFC and is not accessible at boot time.

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

Understanding Flash Memory

The following sections describe flash memory:

- Flash Memory Features, page 2-18
- Security Features, page 2-18
- Flash Memory Configuration Process, page 2-18

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 router 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/command/reference/ffun_r.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 2-15, 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/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/index.htm

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

The router 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 router detects a problem with NVRAM or a checksum error, the router enters **setup** mode. See the "Using the Setup Facility or the setup Command" section on page 2-2 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 and CONFIG_FILE environment variables, use the **boot system** and **boot config** global configuration commands.

Refer to the "Specifying the Startup System Image in the Configuration File" section in the "Loading and Maintaining System Images" chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the BOOT environment variable. Refer to the "Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems" section in the "Managing Configuration Files" chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the CONFIG_FILE variable.

Note When you use the **boot system** 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.

You can view the contents of the BOOT and 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 and CONFIG_FILE environment variables:

```
Router# show bootvar
BOOT variable = disk0:c6sup22-jsv-mz.121-5c.EX.bin,1;
CONFIG_FILE variable does not exist
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 3

Configuring a Supervisor Engine 720

This chapter describes how to configure a Supervisor Engine 720 in a Cisco 7600 series router. This chapter contains these sections:

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- With a 3-slot chassis, install the Supervisor Engine 720 in either slot 1 or 2.
- With a 6-slot or a 9-slot chassis, install the Supervisor Engine 720 in either slot 5 or 6.
- With a 13-slot chassis, install the Supervisor Engine 720 in either slot 7 or 8.

Using the Bootflash or Bootdisk on a Supervisor Engine 720

Release 12.2SR supports the Supervisor Engine 720 64-MB bootflash device (**sup-bootflash:**). For information about using WS-CF-UPG=, which is available with Release 12.2(18)SXE5 and rebuilds and Release 12.2(18)SXF, see this publication:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/cfgnotes/78_17277.htm

Note All Sup720 modules require a minimum of 128-MB bootflash to run Release 12.2SRB and later releases. A CompactFlash (CF) adapter with 512-MB bootflash is available for Sup720 modules in Release 12.2(18)SXF and later releases. Use the Cisco part number CF-ADAPTER= for ordering.

Using the Slots on a Supervisor Engine 720

The Supervisor Engine 720 has two CompactFlash Type II slots. The CompactFlash Type II slots support CompactFlash Type II Flash PC cards sold by Cisco Systems, Inc. The keywords for the slots on the active Supervisor Engine 720 are **disk0:** and **disk1:**. The keywords for the slots on a redundant Supervisor Engine 720 are **slavedisk0:** and **slavedisk1:**.

Configuring Supervisor Engine 720 Ports

Supervisor Engine 720 port 1 has a small form-factor pluggable (SFP) connector and has no unique configuration options.

Supervisor Engine 720 port 2 has an RJ-45 connector and an SFP connector (default). To use the RJ-45 connector, you must change the configuration.

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

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet slot/2`	Selects the Ethernet port to be configured.
Step 2	`Router(config-if)# media-type {rj45 | sfp}`	Selects the connector to use.
	`Router(config-if)# no media-type`	Reverts to the default configuration (SFP).

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

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

Configuring and Monitoring the Switch Fabric Functionality

These sections describe how to configure the switching mode and monitor the switch fabric functionality that is included on a Supervisor Engine 720:

Understanding How the Switch Fabric Functionality Works

These sections describe how the switch fabric functionality works:

Switch Fabric Functionality Overview

The switch fabric functionality is built into the Supervisor Engine 720 and creates a dedicated connection between fabric-enabled modules and provides uninterrupted transmission of frames between these modules. In addition to the direct connection between fabric-enabled modules provided by the switch fabric funtionality, fabric-enabled modules also have a direct connection to the 32-Gbps forwarding bus.

Forwarding Decisions for Layer 3-Switched Traffic

Either a PFC3 or a Distributed Feature Card 3 (DFC3) makes the forwarding decision for Layer 3-switched traffic, as follows:

- A PFC3 makes all forwarding decisions for each packet that enters the router through a module without a DFC3.
- A DFC3 makes all forwarding decisions for each packet that enters the router on a DFC3-enabled module in these situations:
  - If the egress port is on the same module as the ingress port, the DFC3 forwards the packet locally (the packet never leaves the module).
  - If the egress port is on a different fabric-enabled module, the DFC3 sends the packet to the egress module, which sends it out the egress port.
  - If the egress port is on a different nonfabric-enabled module, the DFC3 sends the packet to the Supervisor Engine 720. The Supervisor Engine 720 fabric interface transfers the packet to the 32-Gbps switching bus where it is received by the egress module and is sent out the egress port.

Switching Modes

With a Supervisor Engine 720, traffic is forwarded to and from modules in one of the following modes:

- Compact mode—The router uses this mode for all traffic when only fabric-enabled modules are installed. In this mode, a compact version of the DBus header is forwarded over the switch fabric channel, which provides the best possible performance.
- Truncated mode—The router uses this mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the router sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.
- Bus mode—The router uses this mode for traffic between nonfabric-enabled modules and for traffic between a nonfabric-enabled module and a fabric-enabled module. In this mode, all traffic passes between the local bus and the supervisor engine bus.

Table 3-1 shows the switching modes used with fabric-enabled and nonfabric-enabled modules installed.

Table 3-1 Switch Fabric Functionality Switching Modes

Modules	Switching Modes
Between fabric-enabled modules (when no nonfabric-enabled modules are installed)	Compact[1]
Between fabric-enabled modules (when nonfabric-enabled modules are also installed)	Truncated[2]

Table 3-1 Switch Fabric Functionality Switching Modes

Modules	Switching Modes
Between fabric-enabled and nonfabric-enabled modules	Bus
Between non-fabric-enabled modules	Bus

1. In **show** commands, displayed as dcef mode for fabric-enabled modules with DFC3 installed; displayed as fabric mode for other fabric-enabled modules.
2. Displayed as fabric mode in **show** commands.

Configuring the Switch Fabric Functionality

To configure the switching mode, perform this task:

Command	Purpose
`Router(config)# [no] fabric switching-mode allow {bus-mode | {truncated [{threshold [number]}]}}`	Configures the switching mode.

When configuring the switching mode, note the following information:

- To allow the use of nonfabric-enabled modules or to allow fabric-enabled modules to use bus mode, enter the **fabric switching-mode allow bus-mode** command.
- To prevent the use of nonfabric-enabled modules or to prevent fabric-enabled modules from using bus mode, enter the **no fabric switching-mode allow bus-mode** command.

Caution When you enter the **no fabric switching-mode allow bus-mode** command, power is removed from any nonfabric-enabled modules installed in the router.

- To allow fabric-enabled modules to use truncated mode, enter the **fabric switching-mode allow truncated** command.
- To prevent fabric-enabled modules from using truncated mode, enter the **no fabric switching-mode allow truncated** command.
- To configure how many fabric-enabled modules must be installed before they use truncated mode instead of bus mode, enter the **fabric switching-mode allow truncated threshold** *number* command.
- To return to the default truncated-mode threshold, enter the **no fabric switching-mode allow truncated threshold** command.

Monitoring the Switch Fabric Functionality

The switch fabric functionality supports a number of **show** commands for monitoring purposes. A fully automated startup sequence brings the module online and runs the connectivity diagnostics on the ports.

These sections describe how to monitor the switch fabric functionality:

- Displaying the Switch Fabric Redundancy Status, page 3-5
- Displaying Fabric Channel Switching Modes, page 3-5
- Displaying the Fabric Status, page 3-5

Displaying the Switch Fabric Redundancy Status

To display the switch fabric redundancy status, perform this task:

Command	Purpose
`Router# show fabric active`	Displays switch fabric redundancy status.

```
Router# show fabric active
Active fabric card in slot 5
No backup fabric card in the system
Router#
```

Displaying Fabric Channel Switching Modes

To display the fabric channel switching mode of one or all modules, perform this task:

Command	Purpose
`Router# show fabric switching-mode [module {slot_number | all]`	Displays fabric channel switching mode of one or all modules.

This example shows how to display the fabric channel switching mode of all modules:

```
Router# show fabric switching-mode all
%Truncated mode is allowed
%System is allowed to operate in legacy mode

Module Slot     Switching Mode     Bus Mode
    5                   DCEF     Compact
    9                Crossbar     Compact
Router#
```

Displaying the Fabric Status

To display the fabric status of one or all switching modules, perform this task:

Command	Purpose
`Router# show fabric status [slot_number | all]`	Displays fabric status.

This example shows how to display the fabric status of all modules:

```
Router# show fabric status
 slot    channel      speed              module              fabric
                                         status              status
    1          0         8G                  OK                  OK
    5          0         8G                  OK          Up- Timeout
    6          0        20G                  OK         Up- BufError
    8          0         8G                  OK                  OK
```

```
    8          1          8G                    OK                    OK
    9          0          8G          Down- DDRsync                   OK
Router#
```

Displaying the Fabric Utilization

To display the fabric utilization of one or all modules, perform this task:

Command	Purpose
Router# **show fabric utilization** [*slot_number* | **all**]	Displays fabric utilization.

This example shows how to display the fabric utilization of all modules:

```
Router# show fabric utilization all
Lo% Percentage of Low-priority traffic.
Hi% Percentage of High-priority traffic.

 slot    channel      speed  Ingress Lo%   Egress Lo%  Ingress Hi% Egress Hi%
    5          0        20G            0            0            0          0
    9          0         8G            0            0            0          0
Router#
```

Displaying Fabric Errors

To display fabric errors of one or all modules, perform this task:

Command	Purpose
Router# **show fabric errors** [*slot_number* | **all**]	Displays fabric errors.

This example shows how to display fabric errors on all modules:

```
Router# show fabric errors

Module errors:
 slot    channel        crc      hbeat       sync   DDR sync
    1          0          0          0          0          0
    8          0          0          0          0          0
    8          1          0          0          0          0
    9          0          0          0          0          0

Fabric errors:
 slot    channel       sync     buffer    timeout
    1          0          0          0          0
    8          0          0          0          0
    8          1          0          0          0
    9          0          0          0          0
Router#
```

CHAPTER 4

Configuring a Route Switch Processor 720

This chapter describes how to configure a route switch processor 720 (RSP720). The RSP720 is the newest type of supervisor engine available for Cisco 7600 series routers. The RSP720 consists of a full-size board and two integrated daughter cards: the MSFC4 and a PFC3C or PFC3CXL. The RSP720 has an integrated switch fabric that interconnects all of the line cards in the Cisco 7600 router with point-to-point 20-Gbps full-duplex serial channels.

See Appendix C, "Cisco IOS Release 12.2SRB Software Images," for information about the Cisco IOS software images available for the RSP720, Sup720, and Sup32.

This chapter contains these sections:

For complete syntax and usage information for the commands in this chapter, see the Cisco 7600 series routers command references at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

Note

- The RSP720 is supported in all Cisco 7600 chassis except the Cisco 7603 and Cisco OSR-7609.
- With a 4-slot chassis, install the RSP720 in slot 1 or 2.
- With a 6-slot or a 9-slot chassis (including enhanced [-S] chassis), install the RSP720 in slot 5 or 6.
- With a 13-slot chassis, install the RSP720 in slot 7 or 8.

RSP720 PFC Compatibility Matrix

The route switch processor 720 (RSP720) is configured with one of two types of Policy Feature Card: a PFC3C or a PFC3CXL. The PFC on the RSP720 can interoperate with cards that have a lower version number, such as a Distributed Forwarding Card (DFC) at version 3B (a DFC3B). A card's version number indicates its operating mode: 3B, 3BXL, 3C, or 3CXL.

For the RSP720 to interoperate with lower-version cards, the system determines the lowest operating mode of all installed cards (3B, 3BXL, 3C, or 3CXL) and applies this mode to all of the cards. The cards provide the features supported in the selected operating mode (even if a card has a higher version number). For example, a PFC3C operating in 3B mode offers only those features supported by a PFC3B.

Here are some examples of mode setting among different version cards:

- A system with an RSP720-3CXL, a DFC3BXL, and an Ethernet Services Module (ES20-3C) operates in 3BXL mode (which is the lowest operating mode among all cards).
- A system with an RSP720-3CXL and an ES20-3C operates in 3C mode.
- A system with an RSP720-3C and a DFC3B operates in 3B mode.
- A system with an RSP720-3CXL and an ES20-3CXL operates in 3CXL mode.

Note Use the **show platform hardware pfc mode** command to display the PFC operating mode.

Note OSMs are not supported on RSP 720 Supervisor card.

RSP720 Features

The RSP720 is the newest version of supervisor engine for the Cisco 7600 series router. Along with its two new integrated daughter cards (a PFC3C or PFC3CXL and an MSFC4), the RSP720 provides many enhancements and new features over previous supervisor engines. These enhancements and features are described in the sections that follow.

Note The RSP720 supports all of the features as the Supervisor Engine 720 (with PFC3B or PFC3BXL). Unless otherwise noted, the configuration and operation of the features described in later chapters of this document is the same for both types of processors (RSP720 and Sup720).

Hardware

- Two new integrated daughter cards: a PFC3C or PFC3CXL and an MSFC4
- Faster CPUs and more default memory on the route processor (RP) and switch processor (SP)
  - RSP720-3C-GE: 1-GB DRAM on RP and SP
  - RSP720-3CXL-GE: 2-GB DRAM (RP) and 1-GB DRAM (SP)
- Additional memory provides a larger MAC address table
- Layer 2 and Layer 3 functions have been integrated on a single ASIC
- ASIC (hardware) forwarding of IP and MPLS traffic

For information about hardware support for the RSP720, see the "Route Switch Processor 720" section in Chapter 2 of the *Cisco 7600 Series Router Supervisor Engine and Route Switch Processor Guide*.

Performance

- Faster software bootup
- Faster protocol convergence (BGP, OSPF) and ARP learning
- Improved IGMP snooping times
- Faster speeds for establishing DHCP servers, Label Distribution Protocol (LDP) sessions, IP sessions, and Traffic Engineering (TE)
- Faster processing for Bidirectional Forwarding Detection (BFD), Resource Reservation Setup Protocol (RSVP), and other control-plane functions
- Improved speeds for accessing and copying local files

Scalability

- 30 million packets-per-second (Mpps) forwarding rates for Layer 2 and Layer 3 traffic. The RSP720 uses hardware-based Cisco Express Forwarding (CEF). Forwarding rates are:
  - IP forwarding rates—30 Mpps
  - MPLS forwarding rates—20 Mpps
- Support for larger customer configurations and more interfaces:
  - 32000 IP subscriber sessions
  - 1 million routes
  - 96000 MAC addresses maximum (80000 in real life), up from 64000
  - 32000 VLANs
  - 128000 Address Resolution Protocol (ARP) entries

High-Availability Features

- Online insertion and removal (OIR)
- Route processor redundancy (RPR and RPR+)
- Nonstop forwarding with stateful switchover (NSF/SSO)
- Fast-fabric switchover
- In Service Software Upgrade (ISSU) and enhanced Fast Software Upgrade (eFSU) (Cisco IOS Release 12.2SRB1 and later)

IPv6 ACL Enhancements (Security)

Support for 2K access control list (ACL) labels and 16K access control entries (ACEs), up from 1K masks and 8K ACEs

Rate-Limiting of Unknown Unicast Packets

Allows you to limit the number of unknown unicast packets that the router processes and thus keep the packets from flooding the network. If the number of unknown packets received by the router exceeds the specified rate, excess packets are not forwarded. See the next section ("Configuration Guidelines for Unknown Unicast Packet Rate-Limiting") for configuration guidelines.

The following new commands are provided to configure and verify this feature:

- Use the following command to configure rate-limiting, where *pps* is the maximum number of unknown unicast packets to allow per second (from 10 to 1000000) and *packets-in-burst* is an optional packet burst rate (from 1 to 255, with a default value of 10). The **no** form of the command turns off rate-limiting for unknown packets.

```
Router(config)# mls rate-limit layer2 unknown pps [packets-in-burst]
Router(config)# no mls rate-limit layer2 unknown
```

Note If any physical ports on the router are configured for routing, issue the **mac-address-table learning interface** *interface* command (in global configuration mode) on each of those ports. Otherwise, the rate-limiting counts might not be accurate.

- Use the **show mls rate-limit** command to verify that rate-limiting of unknown unicast packets is enabled. If rate-limiting for unknown unicast packets is enabled, the output will include the following rate-limiter type:

```
Router(config)# 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
---------------------   ----------   ---------   -----   -------
UCAST IP TINY FRAG         On          100000     100   Not sharing
```

Configuration Guidelines for Unknown Unicast Packet Rate-Limiting

Observe these guidelines when configuring unknown unicast packet rate-limiting:

- This feature is available only with the PFC3C and PFC3CXL (RSP720). It is not available with the PFC3B or PFC3BXL.
- If you run the Remote Switched Port Analyzer (RSPAN) with unknown unicast rate-limiting configured, be aware that traffic amounts might differ between RSPAN source and destination ports. This difference occurs if the traffic being monitored contains unknown unicast packets. In this case, the unknown unicast traffic is rate-limited before being sent to the RSPAN destination port, resulting in a mismatch between the amount of traffic at the RSPAN source and destination ports.

Packet Fragmentation over GRE Tunnels

Support for packet fragmentation over Generic Routed Encapsulation (GRE) tunnels. With the PFC3C and PFC3CXL, you can use the [**no**] **mls cef tunnelfragment** command to set the don't fragment (DF) bit to zero, which allows the PFC3C or PFC3CXL to reassemble fragmented GRE traffic. The **no** form of the command turns off tunnel fragmentation and causes fragmented GRE traffic to be dropped.

Note To use this feature, the router at the other end of the tunnel must support tunnel fragmentation.

Improved Load Balancing on GE Bundles

Load balancing improvements on Gigabit Ethernet (GE) bundles configured as 802.1q trunks:

- The VLAN ID is now included in the bundle hash for multicast traffic.
- Multicast receivers that handle multicast traffic for multiple VLANs can load balance the traffic across the member links in the bundle.
- The router provides more efficient load balancing of fragmented traffic.

QoS Enhancements

- On the PFC3C and PFC3CXL you can configure ingress and egress policers to operate independently of each other (in *serial mode*). Normally, ingress and egress policers operate in parallel mode, where action by one policer causes a corresponding action in the other. For example, if the egress policer drops a packet, the ingress policer does not count the packet either. Note that this change does not affect marking using policers.

  To enable serial mode for ingress and egress policers on the PFC3C or PFC3CXL, use the following new command in global configuration mode. The **no** form of the command disables serial mode and resets the policing mode to parallel.

  [**no**] **mls qos police serial**

- Marking packets after recirculation. Rather than using the trust of the original input interface, the PFC3C and PFC3CXL treat recirculated packets as untrusted. This enhancement allows recirculated packets to be marked by an ingress policy.
- Ingress IP DSCP and MPLS EXP marking at the IP-to-MPLS edge. This PFC3C and PFC3CXL enhancement allows you to mark both the IP DSCP bits (**set ip dscp**) and the MPLS EXP bits (**set mpls exp**) during MPLS label imposition. Note that if you do not issue the **set mpls exp** command, the router copies the IP DSCP bits to EXP.
- Ingress EXP marking does not affect locally routed IP-to-IP traffic. With the PFC3C and PFC3CXL, you can use the **no mls qos rewrite ip dscp** command to turn off the egress QoS rewrite of PFC QoS logic, which keeps locally routed IP-to-IP traffic from being affected by EXP marking.
- Concurrent CoS and DSCP transparency for Layer 2 VPNs. This PFC3C and PFC3CXL enhancement enables customers to deploy a combination of Layer 2 VPNs and Layer 3 VPNs for use in a triple-play network (video, Voice over IP [VoIP], and data access [Internet]). It also supports Quality of Service (QoS) guarantees for traffic. The feature results in the following enhancements:
  - You can preserve the CoS and DSCP settings for VPLS and SVI-based EoMPLS, by using the **platform vfi dot1q-transparency** command in conjunction with the **no mls qos rewrite ip dscp** command.
  - The **no mls qos rewrite ip dscp** command can now be used with MPLS. Note that the router must be in PFC3C or PFC3CXL mode, which means that the router cannot contain any Cisco 7600 SIP-600 or WS-X6*xxx* cards (with a DFC3B or DFC3BXL).
  - Because the **no mls qos rewrite ip dscp** command is now compatible with MPLS, Layer 3 VPNs can now be terminated on the same provider edge (PE).
- A new cli command **mls qos recirc untrust** is used to prevent QoS data from getting reset during the second pass lookup over internal vlans for the mvpn case.

Note For complete syntax and usage information for the command **mls qos recirc untrust**, refer to the Cisco 7600 Series Routers Command References at this URL:
http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html

Other Enhancements

- Support for Ethernet over MPLS (EoMPLS) control word
- ACL counters have been enhanced to include counts of packets coming from the route processor (RP)
- IPv6 packet fragments handled in the hardware

Unsupported Features

The following Sup720 features are not supported on the RSP720:

- Server load balancing (SLB)

Accessing Flash Memory on the RSP720

Table 4-1 lists the names of the flash memory devices on the RSP720. To access the appropriate flash memory (internal or external), use these keywords when you issue software commands through the command line interface (CLI):

Table 4-1 CLI Keywords for RSP720 Flash Memory Devices

CLI Keyword	Used to access...
bootdisk:	Internal flash memory on the active route processor (RP)
sup-bootdisk:	Internal flash memory on the active switch processor (SP)
slavebootdisk:	Internal flash memory on the redundant route processor (RP)
slavesup-bootdisk:	Internal flash memory on the redundant switch processor (SP)
disk0:	External flash memory on active RSP (Disk 0 on front panel)
disk1:	External flash memory on active RSP (Disk 1 on front panel)
slavedisk0:	External flash memory on redundant RSP (Disk 0 on front panel)
slavedisk1:	External flash memory on redundant RSP (Disk 1 on front panel)

Configuring route switch processor 720 Ports

route switch processor 720 port 1 has a small form-factor pluggable (SFP) connector and has no unique configuration options.

route switch processor 720 port 2 has an RJ-45 connector and an SFP connector (default). To use the RJ-45 connector, you must change the configuration.

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

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet slot/2`	Selects the Ethernet port to be configured.
Step 2	`Router(config-if)# media-type {rj45 | sfp}` or	Selects the connector to use.
	`Router(config-if)# no media-type`	Reverts to the default configuration (SFP module).

This example shows how to configure port 2 of an RSP720 in slot 5 to use the RJ-45 connector:

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

Configuring and Monitoring the Switch Fabric Functionality

These sections describe how to configure the switching mode and monitor the switch fabric functionality that is included on a route switch processor 720:

Understanding How the Switch Fabric Functionality Works

These sections describe how the switch fabric functionality works:

Switch Fabric Functionality Overview

The switch fabric functionality is built into the route switch processor 720 and creates a dedicated connection between fabric-enabled modules and provides uninterrupted transmission of frames between these modules. In addition to the direct connection between fabric-enabled modules provided by the switch fabric functionary, fabric-enabled modules also have a direct connection to the 32-Gbps forwarding bus.

Forwarding Decisions for Layer 3-Switched Traffic

Either a PFC3 or a Distributed Forwarding Card 3 (DFC3) makes the forwarding decision for Layer 3-switched traffic, as follows:

- A PFC3 makes all forwarding decisions for each packet that enters the router through a module without a DFC3.
- A DFC3 makes all forwarding decisions for each packet that enters the router on a DFC3-enabled module in these situations:
  - If the egress port is on the same module as the ingress port, the DFC3 forwards the packet locally (the packet never leaves the module).
  - If the egress port is on a different fabric-enabled module, the DFC3 sends the packet to the egress module, which sends it out the egress port.
  - If the egress port is on a different nonfabric-enabled module, the DFC3 sends the packet to the route switch processor 720. The route switch processor 720 fabric interface transfers the packet to the 32-Gbps switching bus, where it is received by the egress module and is sent out the egress port.

Switching Modes

With a route switch processor 720, traffic is forwarded to and from modules in one of the following modes:

- Compact mode—The router uses this mode for all traffic when only fabric-enabled modules are installed. In this mode, a compact version of the DBus header (32 bytes) is forwarded over the switch fabric channel, which provides the best possible performance.
- Truncated mode—The router uses this mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the router sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.
- Bus mode—The router uses this mode for traffic between nonfabric-enabled modules and for traffic between a nonfabric-enabled module and a fabric-enabled module. In this mode, all traffic passes between the local bus and the supervisor engine or RSP bus.

Table 4-2 shows the switching modes used with fabric-enabled and nonfabric-enabled modules installed.

Table 4-2 Switch Fabric Functionality Switching Modes

Modules	Switching Modes
Between fabric-enabled modules (when no nonfabric-enabled modules are installed)	Compact[1]
Between fabric-enabled modules (when nonfabric-enabled modules are also installed)	Truncated[2]
Between fabric-enabled and nonfabric-enabled modules	Bus
Between nonfabric-enabled modules	Bus

1. In **show** commands, displayed as dcef mode for fabric-enabled modules with DFC3 installed; displayed as fabric mode for other fabric-enabled modules.
2. Displayed as fabric mode in **show** commands.

Configuring the Switch Fabric Functionality

To configure the switching mode, perform this task:

Command	Purpose
`Router(config)# [no] fabric switching-mode allow {bus-mode | {truncated [{threshold [number]}]}}`	Configures the switching mode.

When configuring the switching mode, note the following information:

- To allow the use of nonfabric-enabled modules or to allow fabric-enabled modules to use bus mode, enter the **fabric switching-mode allow bus-mode** command.
- To prevent the use of nonfabric-enabled modules or to prevent fabric-enabled modules from using bus mode, enter the **no fabric switching-mode allow bus-mode** command.

Caution When you enter the **no fabric switching-mode allow bus-mode** command, power is removed from any nonfabric-enabled modules installed in the router.

- To allow fabric-enabled modules to use truncated mode, enter the **fabric switching-mode allow truncated** command.
- To prevent fabric-enabled modules from using truncated mode, enter the **no fabric switching-mode allow truncated** command.
- To configure how many fabric-enabled modules must be installed before they use truncated mode instead of bus mode, enter the **fabric switching-mode allow truncated threshold** *number* command.
- To return to the default truncated-mode threshold, enter the **no fabric switching-mode allow truncated threshold** command.

Monitoring the Switch Fabric Functionality

The switch fabric functionality supports a number of **show** commands for monitoring purposes. A fully automated startup sequence brings the module online and runs the connectivity diagnostics on the ports.

These sections describe how to monitor the switch fabric functionality:

- Displaying the Switch Fabric Redundancy Status, page 4-9
- Displaying Fabric Channel Switching Modes, page 4-10
- Displaying the Fabric Status, page 4-10
- Displaying the Fabric Utilization, page 4-10
- Displaying Fabric Errors, page 4-11

Displaying the Switch Fabric Redundancy Status

To display the switch fabric redundancy status, perform this task:

Command	Purpose
`Router# show fabric active`	Displays switch fabric redundancy status.

This example shows how to display the redundancy status of the switch fabric:

```
Router# show fabric active
Active fabric card in slot 5
No backup fabric card in the system
Router#
```

Displaying Fabric Channel Switching Modes

To display the fabric channel switching mode of one or all modules, perform this task:

Command	Purpose
Router# **show fabric switching-mode** [**module** *slot_number* | **all**]	Displays fabric channel switching mode of one or all modules.

This example shows how to display the fabric channel switching mode of all modules:

```
Router# show fabric switching-mode all
%Truncated mode is allowed
%System is allowed to operate in legacy mode

Module Slot     Switching Mode     Bus Mode
    5                     DCEF     Compact
    9                 Crossbar     Compact
```

Displaying the Fabric Status

To display the fabric status of one or all switching modules, perform this task:

Command	Purpose
Router# **show fabric status** [*slot_number* | **all**]	Displays fabric status.

This example shows how to display the fabric status of all modules:

```
Router# show fabric status all
 slot    channel      speed               module               fabric
                                          status               status
    1          0         8G                   OK                   OK
    5          0         8G                   OK          Up- Timeout
    6          0        20G                   OK         Up- BufError
    8          0         8G                   OK                   OK
    8          1         8G                   OK                   OK
    9          0         8G        Down- DDRsync                   OK
```

Displaying the Fabric Utilization

To display the fabric utilization of one or all modules, perform this task:

Command	Purpose
Router# **show fabric utilization** [*slot_number* | **all**]	Displays fabric utilization.

This example shows how to display the fabric utilization of all modules:

```
Router# show fabric utilization all
Lo% Percentage of Low-priority traffic.
Hi% Percentage of High-priority traffic.

 slot    channel      speed  Ingress Lo%   Egress Lo%  Ingress Hi% Egress Hi%
    5          0        20G            0            0            0          0
    9          0         8G            0            0            0          0
```

Displaying Fabric Errors

To display fabric errors of one or all modules, perform this task:

Command	Purpose
Router# **show fabric errors** [*slot_number* | **all**]	Displays fabric errors.

This example shows how to display fabric errors on all modules:

```
Router# show fabric errors all

Module errors:
 slot    channel        crc      hbeat       sync   DDR sync
    1          0          0          0          0          0
    8          0          0          0          0          0
    8          1          0          0          0          0
    9          0          0          0          0          0

Fabric errors:
 slot    channel       sync     buffer    timeout
    1          0          0          0          0
    8          0          0          0          0
    8          1          0          0          0
    9          0          0          0          0
```

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 Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- NSF with SSO does not support IPv6 multicast traffic. However, for information about how to configure supervisor engine redundancy using route processor redundancy (RPR) and RPR+ (which does support IPv6 multicast traffic), see Chapter 7, "Configuring RPR and RPR+ Supervisor Engine Redundancy."
- Release 12.2SR does not support single router mode (SRM) with SSO.

This chapter contains 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 With a Supervisor Engine 720, if all of the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on both supervisor engines, which ensures that all modules are operating in dCEF mode and simplifies switchover to the redundant supervisor engine. (CSCec05612)
Post SRE release, the uplink ports are also enabled in dcef mode for RSP720-10G supervisor engine. The other supervisor engines continue to have the uplink ports disabled.

Cisco 7600 series routers 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. Cisco 7600 series routers also support route processor redundancy (RPR), route processor redundancy plus (RPR+). For information about these redundancy modes, see Chapter 7, "Configuring RPR and 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 router, 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, 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 or 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 MSFC 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 router 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 MSFC of the active supervisor engine, and they receive routing updates from their neighbor routers. Routing protocols do not run on the MSFC 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 VRRP is not SSO-aware, meaning state information is not maintained between the active and standby supervisor engine during normal operation. VRRP and SSO can coexist but both features work independently. Traffic that relies on VRRP may switch to the VRRP standby in the event of a supervisor switchover.
- All neighboring devices participating in BGP NSF must be NSF-capable and configured for BGP graceful restart.
- OSPF NSF for virtual links is not supported.
- All OSPF networking devices on the same network segment must be NSF-aware (running an NSF software image).
- For IETF IS-IS, all neighboring devices must be running an NSF-aware software image.
- IPv4 Multicast NSF with SSO is supported by the PFC3 only.
- The underlying unicast protocols must be NSF-aware in order to use multicast NSF with SSO.
- Bidirectional forwarding detection (BFD) is not SSO-aware and is not supported by NSF with SSO.

Supervisor Engine Configuration Synchronization

These sections describe supervisor engine configuration synchronization:

- Supervisor Engine Redundancy Guidelines and Restrictions, page 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 router 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 5-10
- Configuration Mode Restrictions, page 5-10

Redundancy Configuration Guidelines and Restrictions

These guidelines and restrictions apply to all redundancy modes:

- With a Supervisor Engine 720, if all the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on both supervisor engines, which ensures that all modules are operating in dCEF mode and simplifies switchover to the redundant supervisor engine.

  Post SRE release, the uplink ports are also enabled in dcef mode for RSP720-10G supervisor engine. The other supervisor engines continue to have the uplink ports disabled.s
- 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 router 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.
- With a Supervisor Engine 720, if a fabric synchronization error occurs, the default behavior is to switchover to the redundant supervisor engine. In some cases, a switchover to the redundant supervisor engine is more disruptive than powering down the module that caused the fabric synchronization error. Enter the **no fabric error-recovery fabric-switchover** command to disable the switchover and power down the module with the fabric synchronization error.

Hardware Configuration Guidelines and Restrictions

For redundant operation, the following guidelines and restrictions must be met:

- Cisco IOS software running on the supervisor engine and the MSFC supports redundant configurations when the supervisor engines and MSFC routers are identical. If they are not identical, one will boot first and become active and hold the other supervisor engine and MSFC in a reset condition.
- Each supervisor engine must have the resources to run the router 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 router 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 multicast forwarding entries should be completed.
Step 4	`Router(config)# mls ip multicast sso leak` *percentage*	Specifies the percentage of multicast flows; valid values are from 1 to 100 percent. The value represents the percentage of the total number of existing PIM sparse mode and PIM dense mode multicast flows that should be flagged for packet leaking.

Verifying Multicast NSF with SSO

To verify the multicast NSF with SSO settings, enter the **show mls ip multicast sso** command:

```
router# show mls ip multicast sso
Multicast SSO is enabled
Multicast HA Parameters
-------------------------------------------------+------+
protocol convergence timeout                      120 secs
flow leak percent                                 10
flow leak interval                                60 secs
```

Configuring CEF NSF

The CEF NSF feature operates by default while the networking device is running in SSO mode. No configuration is necessary.

Verifying CEF NSF

To verify that CEF is NSF-capable, enter the **show cef state** command:

```
router# show cef state

CEF Status [RP]
CEF enabled/running
dCEF enabled/running
CEF switching enabled/running
CEF default capabilities:
Always FIB switching:      yes
Default CEF switching:     yes
Default dCEF switching:    yes
Update HWIDB counters:     no
Drop multicast packets:    no
.
.
.
CEF NSF capable:           yes
IPC delayed func on SSO:   no
RRP state:
I am standby RRP:          no
My logical slot:           0
RF PeerComm:               no
```

Configuring BGP NSF

Note You must configure BGP graceful restart on all peer devices participating in BGP NSF.

To configure BGP for NSF, perform this task 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 family 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.
Step 5	`Router(config-router)# nsf t3 {manual [seconds] | adjacency}`	(Optional) Specifies the time IS-IS will wait for the IS-IS database to synchronize before generating overloaded link-state information for itself and flooding that information out to its neighbors. The **t3** keyword applies only if you selected IETF operation. When you specify **adjacency**, the router that is restarting obtains its wait time from neighboring devices.
Step 6	`Router(config-router)# nsf interface wait seconds`	(Optional) Specifies how long an IS-IS NSF restart will wait for all interfaces with IS-IS adjacencies to come up before completing the restart. The default is 10 seconds.

Verifying IS-IS NSF

To verify IS-IS NSF, you must check that the NSF function is configured on the SSO-enabled networking device. To verify IS-IS NSF, follow these steps:

Step 1 Verify that "nsf" appears in the IS-IS configuration of the SSO-enabled device by entering the **show running-config** command. The display will show either the Cisco IS-IS or the IETF IS-IS configuration. The following display indicates that the device uses the Cisco implementation of IS-IS NSF:

```
Router# show running-config
<...Output Truncated...>
router isis
nsf cisco
<...Output Truncated...>
```

Step 2 If the NSF configuration is set to **cisco**, enter the **show isis nsf** command to verify that NSF is enabled on the device. Using the Cisco configuration, the display output will be different on the active and redundant RPs. The following display shows sample output for the Cisco configuration on the active RP. In this example, note the presence of "NSF restart enabled":

```
router# show isis nsf

NSF is ENABLED, mode 'cisco'

RP is ACTIVE, standby ready, bulk sync complete
NSF interval timer expired (NSF restart enabled)
Checkpointing enabled, no errors
Local state:ACTIVE,  Peer state:STANDBY HOT,  Mode:SSO
```

The following display shows sample output for the Cisco configuration on the standby RP. In this example, note the presence of "NSF restart enabled":

```
router# show isis nsf

NSF enabled, mode 'cisco'
RP is STANDBY, chkpt msg receive count:ADJ 2, LSP 7
NSF interval timer notification received (NSF restart enabled)
Checkpointing enabled, no errors
Local state:STANDBY HOT,  Peer state:ACTIVE,  Mode:SSO
```

Step 3 If the NSF configuration is set to **ietf**, enter the **show isis nsf** command to verify that NSF is enabled on the device. The following display shows sample output for the IETF IS-IS configuration on the networking device:

```
router# show isis nsf

NSF is ENABLED, mode IETF
NSF pdb state:Inactive
NSF L1 active interfaces:0
NSF L1 active LSPs:0
NSF interfaces awaiting L1 CSNP:0
Awaiting L1 LSPs:
NSF L2 active interfaces:0
NSF L2 active LSPs:0
NSF interfaces awaiting L2 CSNP:0
Awaiting L2 LSPs:
Interface:Serial3/0/2
    NSF L1 Restart state:Running
    NSF p2p Restart retransmissions:0
    Maximum L1 NSF Restart retransmissions:3
    L1 NSF ACK requested:FALSE
    L1 NSF CSNP requested:FALSE
    NSF L2 Restart state:Running
    NSF p2p Restart retransmissions:0
    Maximum L2 NSF Restart retransmissions:3
    L2 NSF ACK requested:FALSE
Interface:GigabitEthernet2/0/0
    NSF L1 Restart state:Running
    NSF L1 Restart retransmissions:0
    Maximum L1 NSF Restart retransmissions:3
    L1 NSF ACK requested:FALSE
    L1 NSF CSNP requested:FALSE
    NSF L2 Restart state:Running
    NSF L2 Restart retransmissions:0
    Maximum L2 NSF Restart retransmissions:3
    L2 NSF ACK requested:FALSE
    L2 NSF CSNP requested:FALSE
```

```
Interface:Loopback1
    NSF L1 Restart state:Running
    NSF L1 Restart retransmissions:0
    Maximum L1 NSF Restart retransmissions:3
    L1 NSF ACK requested:FALSE
    L1 NSF CSNP requested:FALSE
    NSF L2 Restart state:Running
    NSF L2 Restart retransmissions:0
    Maximum L2 NSF Restart retransmissions:3
    L2 NSF ACK requested:FALSE
    L2 NSF CSNP requested:FALSE
```

Configuring EIGRP NSF

To configure EIGRP NSF, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# router eigrp` *as-number*	Enables an EIGRP routing process, which places the router in router configuration mode.
Step 3	`Router(config-router)# nsf`	Enables EIGRP NSF. Use this command on the restarting router and all of its peers.

Verifying EIGRP NSF

To verify EIGRP NSF, you must check that the NSF function is configured on the SSO-enabled networking device. To verify EIGRP NSF, follow these steps:

Step 1 Verify that "nsf" appears in the EIGRP configuration of the SSO-enabled device by entering the **show running-config** command:

```
Router# show running-config
.
.
.
router eigrp 100
 auto-summary
 nsf
.
.
.
```

Step 2 Enter the **show ip protocols** command to verify that NSF is enabled on the device:

```
Router# show ip protocols
*** IP Routing is NSF aware ***
Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
```

```
EIGRP maximum hopcount 100
EIGRP maximum metric variance 1
Redistributing: eigrp 100
EIGRP NSF-aware route hold timer is 240s
EIGRP NSF enabled
   NSF signal timer is 20s
   NSF converge timer is 120s
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
Routing Information Sources:
  Gateway         Distance      Last Update
Distance: internal 90 external 170
```

Synchronizing the Supervisor Engine Configurations

During normal operation, the startup-config and config-registers configurations are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.

Copying Files to the Redundant Supervisor Engine

Enter this command to copy a file to the **disk0:** device on a redundant supervisor engine:

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

Enter this command to copy a file to the **bootflash:** device on a redundant supervisor engine:

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

Enter this command to copy a file to the **bootflash:** device on a redundant MSFC:

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

CHAPTER 6

ISSU and eFSU on Cisco 7600 Series Routers

This chapter provides information about how to perform a software upgrade on a Cisco 7600 series router using the In Service Software Upgrade feature.

This chapter contains the following sections:

ISSU and eFSU Overview

In most networks, software upgrades require system downtime. With the In Service Software Upgrade (ISSU) feature, however, you can upgrade the router software while the router continues to forward traffic. Thus, ISSU increases network availability and reduces the downtime caused by software upgrades.

The Cisco 7600 series router supports the following types of upgrade procedures. The same ISSU commands and upgrade procedure are used for both types of upgrades.

- ISSU—Provides software upgrades with minimal system downtime. This feature is available for software upgrades between Cisco IOS software releases that have the same line card software. (Available in Cisco IOS Release 12.2SRB1 and later releases.)
- enhanced Fast Software Upgrade (eFSU)—A subset of ISSU, eFSU helps to minimize outage time during a software upgrade by preloading new line card software images onto supported line cards. This feature is available for upgrades between releases that have different line card software. (Available in Cisco IOS Release 12.2SRB and later releases.)

ISSU uses the existing features of NonStop Forwarding (NSF) with Stateful SwitchOver (SSO) to perform the software upgrade. In a redundant system with two supervisor engines or route switch processors (RSPs), one of the processors is active while the other operates in standby mode, ready to take over processing if the active processor goes down.

During an in service software upgrade (ISSU or eFSU), new software is loaded onto the standby processor while the active processor continues to operate using the old software. As part of the upgrade, a switchover occurs between the active and standby processors, and the standby processor becomes active and begins running the new software. You can continue with the upgrade to load the new software onto the other processor, or you can abort the upgrade and resume operation with the old software.

If the new software release contains new line card software and the line cards in the router support eFSU, the upgrade process preloads the new line card software onto the line cards. When the switchover occurs (between the active and standby processors), the line cards are restarted with the new software image. By preloading the new software image onto the line cards, eFSU helps to minimize outage time during the software upgrade.

For detailed information about ISSU, see the *Cisco IOS In Service Software Upgrade Process* document at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/newft/122srb33/sbisefsu.htm

ISSU Overview

Cisco IOS Releases 12.2SRB1 and later support ISSU on Cisco 7600 series routers. During an ISSU upgrade, the router continues to route and forward traffic, which allows you to upgrade from one software release to another with minimal system downtime (between 0 and 3 seconds).

ISSU is available for NSF/SSO compliant protocols and software features on the router. See the "ISSU Support" section on page 6-5 for a list of Cisco 7600 protocols and features that support ISSU.

Note To perform an ISSU upgrade, the old and new Cisco IOS software releases must have the same line card software. If the releases have different line card software, the router performs an eFSU upgrade.

eFSU Overview

eFSU helps to minimize outage time during an in service software upgrade by preloading a new software image onto line cards that support the feature. During the software upgrade process, a switchover occurs between the active and standby supervisor engines or RSPs. When the switchover occurs, the line cards are restarted. Line cards that support eFSU are restarted with the new, preloaded software image.

The following Cisco 7600 line cards support eFSU:

- SIP-400 and SIP-600
- ESM-2x10GE and ESM-20x1GE-3C
- 67*xx* line cards

All other Cisco 7600 line cards undergo a hard reset at switchover, and the software image is loaded after the line card is restarted.

Note To support eFSU, a line card must have 512 MB of memory, with enough memory available to hold the new software image. If enough memory is not available, the software preload fails and the line card undergoes a reset during the switchover.

Outage Time and Support Considerations

During an eFSU upgrade, line cards are restarted or reset after the switchover that occurs between processors. Because the line cards are restarted or reset, any links attached to the line cards flap and traffic processing is disrupted until protocols and software features are brought back online. The length of time that line card processing is disrupted (outage time) depends on whether the eFSU process was able to preload a new software image onto the line card.

- For line cards that support eFSU, the outage time is similar to that in RPR+ mode.
- For line cards that do not support eFSU, the outage time is similar to that in RPR mode.

Once the new software is downloaded (**issu loadversion**), you can use the **show issu outage slot all** command to display the maximum outage time for installed line cards. See the "Displaying Maximum Outage Time for Installed Line Cards (Optional)" section on page 6-14 for a command example.

If you attempt to load an earlier version of software onto the router and the new (earlier) version does not support a currently installed line card, one of two things happens:

- If you use the **force** option in the **issu loadversion** command, the router is placed in RPR mode.
- If you omit the **force** option, the eFSU process is aborted and error messages are displayed to indicate that there is a problem with the line card.

Reserving Line Card Memory

On line cards that support eFSU, the router automatically reserves memory on the line card to store the new software image (decompressed format). The amount of memory needed varies according to line card type.

Although we do not recommend it, you can issue the following command to keep the router from reserving memory for the software preload (where *slot-num* specifies which slot the line card is installed in):

```
no mdr download reserve memory image slot slot-num
```

Note If a line card does not have enough memory available to hold the new software image, software preload fails and the card undergoes a reset during the switchover. Outage time is similar to that with RPR (because the new line card image must be loaded after the line card is restarted).

To determine how much memory will be reserved on the line card, use the **show mdr download image** command, as shown in the following example (for a Cisco 7600 SIP-600):

```
SIP-600# show mdr download image
Pre-download information
Slot CPU In-Progress Complete LC Mem Resv (bytes)
 1   0     N           N        0
 1   1     N           N        0
 2   0     N           N        0
 2   1     N           N        0
 3   0     N           N        0
 3   1     N           N        0
 4   0     N           N        0
 4   1     N           N        0
 5   0     N           N        0
 5   1     N           N        0
 6   0     N           Y        36175872
 6   1     N           N        0
 7   0     N           N        0
 7   1     N           N        0
 8   0     N           N        0
 8   1     N           N        0
```

```
 9   0     N          Y        35127296
 9   1     N          N        0
10   0     N          Y        31195136
10   1     N          N        0
11   0     N          N        0
11   1     N          N        0
12   0     N          N        0
12   1     N          N        0
13   0     N          Y        31195136
13   1     N          N        0

SIP-600#
```

eFSU Operation

During a software upgrade, the router performs the following steps on line cards that support eFSU. These steps occur automatically during the upgrade process, and no user intervention is required.

- Reserves the necessary memory for the new Cisco IOS software image on each installed line card (if the line card supports software preload).
- Preloads a new software image onto supported line cards as part of the **issu loadversion** command.
- Restarts line cards with the new software image when switchover occurs (**issu runversion**).
- If a rollback or abort occurs, the router restores the line card software to its original version. To provide as little disruption as possible, the router preloads the original software version back onto the line card. Once the rollback or abort is completed, the line card is restarted with the original software version.

Error Handling for Line Card Software Preload

If problems occur during line card software preload, the router takes the following actions:

- Line card crash during load version—The line card is reset when switchover occurs.
- Line card not active when eFSU started—No power is provided to the line card during the software upgrade, and the line card is reset when the process ends. The same action is applied to a line card that is inserted into the router after the software upgrade process has begun.
- Line card crash during run version or during rollback—The line card boots with the software image version that corresponds to the software image that is present on the active supervisor engine or RSP.

Cisco 7600 ISSU and eFSU Support

During an ISSU upgrade, NSF/SSO compliant protocols and software features (such as those below) continue to operate and minimal system downtime occurs. Routing protocols and software features that are not NSF/SSO compliant are restarted during the upgrade, which means that they stop operating for awhile after the restart until they are brought back online.

To perform an ISSU upgrade, the old and new Cisco IOS software releases must have the same line card software. If the releases have different line card software, the router performs an eFSU upgrade.

ISSU Support

ISSU is supported on the following Cisco 7600 hardware and software:

Availability

- Cisco IOS release 12.2(33) SRB1 and later

Hardware

- All supported Cisco 7600 chassis (including enhanced [-S] chassis)
- RSP720-3C, RSP720-3CXL, Sup720-3B, Sup720-3BXL, Sup32
- All Cisco 7600 line cards, DFCs, and other modules

Software

- 802.1q
- 802.1x
- ARP
- ATM
- BGP
- Etherchannel (PagP and LACP)
- GLBP
- HDLC
- HSRP
- IPv4
- L2 multicast
- MLP (Multilink PPP)
- MPLS (including LDP, TE, and VPN)—See the *ISSU MPLS Client* document for information about the steps you should perform during the upgrade.
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb28/iscli28.htm
- NetFlow
- PPP
- QoS
- RIB (routing information base)
- SNMP
- SPAN and Remote SPAN
- STP
- VRF (Virtual Routing and Forwarding)

ISSU Compatibility Matrix for Cisco IOS Software Releases

The following table lists the compatibility matrix for ISSU between various Cisco IOS releases for C7600. If the ISSU is supported between two releases, corresponding column is marked as yes and if ISSU is not supported between two releases the column is marked as no.

Table 6-1 ISSU Compatibility Matrix

I

Cisco IOS Release	12.2 (33) SRB	12.2 (33) SRC	12.2 (33) SRD	12.2 (33) SRE	15.0 (1) S and later releases
12.2 (33) SRB	Yes	Yes	No	No	No
12.2 (33) SRC	Yes	Yes	No	No	No
12.2 (33) SRD	No	No	Yes	Yes	No
12.2 (33) SRE	No	No	Yes	Yes	No
15.0(1) S and later releases	No	No	No	No	Yes

eFSU Support

eFSU is supported on the following Cisco 7600 hardware and software:

Availability

- Cisco IOS release 12.2 (33) SRB and later
- RSP720 and Sup32 support introduced in release 12.2(33) SRB1

Hardware

- All Cisco 7600 chassis (except the 3-slot chassis, CISCO7603)
- RSP720-3C, RSP720-3CXL, Sup720-3B, Sup720-3BXL, Sup32
- DFC3B, DFC3BXL, DFC3C, DFC3CXL
- ESM-2x10GE, ESM-20x1GE-3C
- SIP-400, SIP-600, 67*xx* line cards

Unsupported Hardware

- Enhanced FlexWAN, OSM-GE-WAN
- 68*xx*, 65*xx*, 64*xx*, 63*xx*, and 61*xx* line cards

Software Support

During an eFSU, Cisco 7600 line cards are restarted, and software features and routing protocols are not available during the restart. Outage time depends on whether the line cards support eFSU (see the "eFSU Overview" section on page 6-2).

Cisco 7600 ISSU and eFSU Guidelines and Limitations

Following is a list of guidelines and limitations for performing an in service software upgrade on Cisco 7600 series routers. Unless otherwise noted, the guidelines apply to both ISSU and eFSU.

- Unsupported Cisco 7600 hardware and software can co-exist with ISSU or eFSU (that is, both can be present in the router). In addition, the router gracefully restarts any unsupported protocols to prevent "black hole" situations.

- To perform an in service software upgrade, a router requires two route processors (RPs): an active RP and a standby RP. On the Cisco 7600 router, two supervisor engines or route switch processors (RSPs) are required because they contain the route processors for the router.
- Both the active and standby supervisor engines or RSPs must have at least 256 MB of flash memory in which to store both the old and new software images prior to the upgrade process.
- The same ISSU commands and upgrade procedure are used for both ISSU and eFSU. The only difference is that during an ISSU upgrade, the line cards are not restarted as they are during an eFSU.
- The router examines the old and new software images and automatically performs the appropriate process (ISSU or eFSU) to upgrade the software image:
  - If the line card software is the same in both the old and new software images, the router performs an ISSU to upgrade the software. System downtime is from 0 to 3 seconds.
  - If the line card software in both images is different, line cards are restarted or reset during the upgrade process. System downtime depends on whether the line cards support eFSU (see the "Outage Time and Support Considerations" section on page 6-3 for more information).
- The ISSU upgrade feature is supported for all software features that support NSF/SSO. Software features that do not support NSF/SSO stop operating for awhile, until they are brought back online after the switchover that occurs during the software upgrade.
- All line cards that support eFSU must have at least 512 MB of memory for software preload to succeed. Otherwise, the preload fails for those line cards.
- Line cards that support eFSU must have enough memory available to hold the new software image. If enough memory is not available, the software preload fails and the cards undergo a reset during the switchover (that occurs between the active and standby supervisor engines or RSPs).
- ISSU and eFSU are supported only in SSO mode. They are not supported in RPR and RPR+ mode.
- Online insertion and replacement (OIR) is not supported during an in service software upgrade. If you attempt to insert a new line card in the router while the upgrade is active, the router does not provide power for the card. When the upgrade ends, the router resets the newly inserted line card.
- Do not perform a manual switchover between supervisor engines or RSPs during the upgrade. Although the router allows it, we strongly discourage this.
- ISSU commands (which are also used for eFSU) are available in the command-line interface (CLI) only if a supported processor is installed in the router. The commands are not available if another type of supervisor engine or RSP is installed.
  - In Release 12.2SRB1 or later, the commands are available with the RSP720, Sup720, or Sup32.
  - In Release 12.2SRB, the commands are available only if a Sup720 is installed.
- Make sure that the configuration register is set to allow autoboot (the lowest byte of the register should be set to 2).
- Before you issue the **issu abortversion** command (to abort a software upgrade), make sure that the standby supervisor engine or RSP is Up (STANDBY HOT [in SSO] or COLD [in RPR]).
- Use the Fast Software Upgrade (FSU) process to upgrade from an earlier software version to Cisco IOS Release 12.2SRB or later. During this process, the line card software image is also upgraded on those line cards that support eFSU.
- On modules that do not support eFSU, you can upgrade software images in Route Processor Redundancy (RPR) mode.

Performing an In Service Software Upgrade

The following sections describe the process for performing an in service software upgrade (ISSU or eFSU) on the Cisco 7600 router. The following steps are discussed:

Each section briefly describes a particular step in the upgrade process and provides command examples. In the command examples, important fields in the command output are shown in boldface. Check these fields to verify the status of the command.

For detailed information about any of the commands, see the *Cisco IOS In Service Software Upgrade Process* document (feature guide) on the 12.2SRB new feature documentation site at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/newft/122srb33/sbisefsu.htm

Software Upgrade Process Summary

Here is a summary of the tasks required to upgrade (or downgrade) a software image on the Cisco 7600 router. The following sections provide examples of the software upgrade process on the router.

The same set of ISSU commands and upgrade procedure are used for both ISSU and eFSU.

Note The ISSU upgrade process is available only for Cisco IOS releases that share the same line card software. If the line card software in the releases is different, the router performs an eFSU upgrade.

	Command or Action	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Router# copy tftp disk_name`	Uses TFTP to copy the new software image to flash memory on the active and standby RPs (disk0: and slavedisk0:). Answer the prompts to identify the name and location of the new software image.
Step 3	`Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 4	`Router(config)# no service image-version efsu` `Router(config)# end`	Disables the compatibility matrix check, which is necessary to perform a software upgrade on the Cisco 7600 router. The **end** command returns you to privileged EXEC mode. **Note** Remove the **no service image-version efsu** configuration by the default **service image-version efsu** after the ISSU upgrade is completed.

Step 5	`Router# show version | in image`
	`Router# show bootvar`
	
	`Router# show redundancy`
	`Router# show issu state [detail]`
		These **show** commands verify that the router is ready to run ISSU or eFSU. The **show version** and **show bootvar** commands verify the boot image settings.
		The **show redundancy** and **show issu state** commands verify that redundancy mode is enabled and that SSO and NSF are configured.
		Note Use **show redundancy** and **show issu state** throughout the upgrade (as shown in the following examples) to verify the status of the upgrade.

Step 6	`Router# issu loadversion` *active-slot active-image standby-slot standby-image* `[force]`	Starts the upgrade process and loads the new software image onto the standby RP. It may take several seconds for the new image to load and for the standby RP to transition to SSO mode.
Step 7	`Router# show issu outage slot all`	(Optional) Displays the maximum outage time for installed line cards. Issue the command on the switch processor (SP) of the supervisor engine or RSP.
Step 8	`Router# issu runversion` *slot image*	Forces a switchover, causing the standby supervisor engine or RSP to become active and begin running the new software. The previously active processor becomes standby and boots with the old image.
Step 9	`Router# issu acceptversion` {*active slot-number* | **active slot-name** *slot-name*}	(Optional) Halts the rollback timer to ensure that the new software image is not automatically aborted during the upgrade process.
Step 10	`Router# issu commitversion` *slot active-image*	Loads the new software image onto the standby supervisor engine or RSP in the specified slot.
Step 11	`Router# show redundancy` `Router# show issu state [detail]`	Verifies the status of the upgrade process. If the upgrade was successful, both the active and standby supervisor engines or RSPs are running the new software version.

Step 12	`Router# configure terminal`
	`Router(config)# service image-version efsu`
	`Router(config)# end`
		Enters global configuration mode and enables the compatibility matrix check.

Preparing for the Upgrade

Before attempting to perform a software upgrade, be sure to review the "Cisco 7600 ISSU and eFSU Guidelines and Limitations" section on page 6-6.

To prepare for ISSU or eFSU, perform the tasks in the following sections:

- Disabling the Compatibility Matrix Check, page 6-10
- Verifying the Boot Image Version and Boot Variable, page 6-10

Disabling the Compatibility Matrix Check

To perform a software upgrade on the Cisco 7600 router, you must first disable the compatibility matrix check by issuing the following command in global configuration mode:

```
Router(config)# no service image-version efsu
```

Verifying the Boot Image Version and Boot Variable

Before starting, it is a good idea to issue the **show version** and **show bootvar** commands to verify the boot image version and BOOT environment variable for the router, as shown in the following examples:

```
Router# show version | in image
System image file is "disk0:oct22"

Router# show bootvar
BOOT variable = disk0:oct22,1;
CONFIG_FILE variable =
BOOTLDR variable =
Configuration register is 0x2102

Standby BOOT variable = disk0:oct22,1;
Standby CONFIG_FILE variable =
Standby BOOTLDR variable =
Standby Configuration register is 0x2102
```

Verifying Redundancy Mode

It is also a good idea to verify that redundancy mode is enabled and that NSF and SSO are configured. The following command example shows how to verify redundancy:

```
Router# show redundancy

Redundant System Information :
------------------------------
       Available system uptime = 9 minutes
Switchovers system experienced = 0
              Standby failures = 0
        Last switchover reason = none

                 Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 5
        Current Software state = ACTIVE
       Uptime in current state = 8 minutes
                 Image Version = Cisco IOS Software, s72033_rp Software
(s72033_rp-ADVENTERPRISEK9_WAN_DBG-M), Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Mon 23-Oct-06 02:43 by
```

```
                        BOOT = disk0:oct22,1;
                   CONFIG_FILE =
                      BOOTLDR =
        Configuration register = 0x2102

Peer Processor Information :
----------------------------
              Standby Location = slot 6
        Current Software state = STANDBY HOT
       Uptime in current state = 8 minutes
                 Image Version = Cisco IOS Software, s72033_rp Software
(s72033_rp-ADVENTERPRISEK9_WAN_DBG-M), Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Mon 23-Oct-06 02:43 by
                        BOOT = disk0:oct22,1;
                   CONFIG_FILE =
                      BOOTLDR =
        Configuration register = 0x2102
```

Verifying ISSU State

You should also verify the ISSU state, as shown here:

```
Router# show issu state

                        Slot = 5
                   RP State = Active
                  ISSU State = Init
                Boot Variable = disk0:oct22,1;

                        Slot = 6
                   RP State = Standby
                  ISSU State = Init
                Boot Variable = disk0:oct22,1;
```

Copying the New Software Image

Before starting the ISSU or eFSU process, you must copy the new software image to flash memory (disk0: and slavedisk0:) on the active and standby route processors, which are located on the Cisco 7600 supervisor engine or route switch processor.

```
Router# copy tftp disk0:

Address or name of remote host []? sys1
Source filename []? /auto/tftpboot/image/c7600s72033-adventerprisek9_wan-mz
Destination filename [c7600s72033-adventerprisek9_wan-mz]? c7600s72033
Accessing tftp://sys1//auto/tftpboot/image/c7600s72033-adventerprisek9_wan-mz
Loading /auto/tftpboot/image/c7600s72033-adventerprisek9_wan-mz from 192.0.2.245 (via
GigabitEthernet5/1):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
(command output omitted)


[OK - 124978660 bytes]

124978660 bytes copied in 259.868 secs (480931 bytes/sec)

Router# copy disk0:c7600s72033 slavedisk0:

Destination filename [c7600s72033]?
```

```
Copy in progress...CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```
(command output omitted)

```
124978660 bytes copied in 308.488 secs (405133 bytes/sec)
```

Loading the New Software onto the Standby RP

Issue the **issu loadversion** command to start the upgrade process. This step reboots the standby supervisor engine or RSP and loads the new software image onto the standby's route processor. When the download is complete, you are prompted to issue the **runversion** command.

```
Router# issu loadversion 5 disk0:c7600s72033 6 slavedisk0:c7600s72033
Router#
*Oct 27 21:26:20.119: %OIR-SP-3-PWRCYCLE: Card in module 6, is being power-cycled (RF
request)
```

(The above line shows that the standby supervisor engine in slot 6 is rebooting. When the reboot is complete, the router loads the new image onto the standby.)

```
*Oct 27 21:26:20.775: %PFREDUN-SP-6-ACTIVE: Standby processor removed or reloaded,
changing to Simplex mode
*Oct 27 21:29:46.123: SP: pf_redun_check_img_compatibility: MATRIX result is compatible!!!
Of course...
*Oct 27 21:29:47.135: %PFREDUN-SP-6-ACTIVE: Standby initializing for SSO mode
*Oct 27 21:29:47.431: %SYS-SP-3-LOGGER_FLUSHED: System was paused for 00:00:00 to ensure
console debugging output.
*Oct 27 21:29:50.647: %PFINIT-SP-5-CONFIG_SYNC: Sync'ing the startup configuration to the
standby Router.
*Oct 27 21:30:29.687: %FABRIC-SP-5-CLEAR_BLOCK: Clear block option is off for the fabric
in slot 6.
*Oct 27 21:30:29.783: %FABRIC-SP-5-FABRIC_MODULE_BACKUP: The Switch Fabric Module in slot
6 became standby
*Oct 27 21:30:30.523: %DIAG-SP-6-RUN_MINIMUM: Module 6: Running Minimal Diagnostics...
*Oct 27 21:30:32.459: %DIAG-SP-6-DIAG_OK: Module 6: Passed Online Diagnostics
*Oct 27 21:30:32.675: %OIR-SP-6-INSCARD: Card inserted in slot 6, interfaces are now
online
*Oct 27 21:29:46.071: %SYS-SP-STDBY-3-LOGGER_FLUSHED: System was paused for 00:00:00 to
ensure console debugging output.
*Oct 27 21:30:14.123: %SPANTREE-SP-STDBY-5-EXTENDED_SYSID: Extended SysId enabled for type
vlan
*Oct 27 21:30:14.539: SP-STDBY: SP: Currently running ROMMON from S (Gold) region
*Oct 27 21:30:17.067: %DIAG-SP-STDBY-6-RUN_MINIMUM: Module 6: Running Minimal
Diagnostics...
*Oct 27 21:30:29.331: %DIAG-SP-STDBY-6-DIAG_OK: Module 6: Passed Online Diagnostics
*Oct 27 21:31:30.431: %SYS-SP-STDBY-5-RESTART: System restarted --
Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ), Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Thu 26-Oct-06 03:49 by
*Oct 27 21:33:14.231: %ISSU_PROCESS-SP-7-DEBUG: Peer state is [ STANDBY HOT ]; Please
issue the runversion command
*Oct 27 21:33:13.471: %PFREDUN-SP-STDBY-6-STANDBY: Ready for SSO mode
*Oct 27 21:33:14.239: %RF-SP-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
*Oct 27 21:33:13.655: %SYS-SP-STDBY-3-LOGGER_FLUSHED: System was paused for 00:00:00 to
ensure console debugging output.
```

Note At this point, it is a good idea to check the status of the upgrade using the **show redundancy** and **show issu state detail** commands (see the following examples). When **issu loadversion** has finished, the standby RP should be loaded with the new software image and the RP should be in SSO mode. It might take several seconds for **issu loadversion** to complete; therefore, if you enter the **show** commands too soon you might not see the information you need.

```
Router# show redundancy

Redundant System Information :
------------------------------
       Available system uptime = 38 minutes
Switchovers system experienced = 0
              Standby failures = 1
        Last switchover reason = none

                  Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 5
        Current Software state = ACTIVE
       Uptime in current state = 37 minutes
                 Image Version = Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ),
Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Mon 23-Oct-06 02:43 by
                          BOOT = disk0:oct22,12
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2102

Peer Processor Information :
----------------------------
              Standby Location = slot 6
        Current Software state = STANDBY HOT
       Uptime in current state = 13 minutes
                 Image Version = Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ),
Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Thu 26-Oct-06 03:21 by
                          BOOT = disk0:c7600s72033,12;disk0:oct22,12
                   CONFIG_FILE =
                       BOOTLDR =
        Configuration register = 0x2102


Router# show issu state detail

                          Slot = 5
                     RP State = Active
                    ISSU State = Load Version
                  Boot Variable = disk0:oct22,12
                 Operating Mode = sso
                Primary Version = disk0:oct22
              Secondary Version = disk0:c7600s72033
                Current Version = disk0:oct22
                 Variable Store = PrstVbl
```

```
          ROMMON CV = [disk0:oct22]

               Slot = 6
           RP State = Standby
         ISSU State = Load Version
       Boot Variable = disk0:c7600s72033,12;disk0:oct22,12
     Operating Mode = sso
    Primary Version = disk0:oct22
  Secondary Version = disk0:c7600s72033
    Current Version = disk0:c7600s72033
```

Displaying Maximum Outage Time for Installed Line Cards (Optional)

Once the new software is downloaded, you can issue the **show issu outage slot all** command on the switch processor (SP) to display the maximum outage time for installed line cards:

```
Router# show issu outage slot all

Slot # Card Type                                MDR Mode          Max Outage Time
------ ---------------------------------------  -----------       ---------------
     1 CEF720 24 port 1000mb SFP                WARM_RELOAD        300 secs
     2 1-subslot SPA Interface Processor-600    WARM_RELOAD        300 secs
     3 4-subslot SPA Interface Processor-400    WARM_RELOAD        300 secs
     4 2+4 port GE-WAN                          RELOAD             360 secs

Router#
```

Note For an ISSU upgrade, the command output displays NSF_RELOAD for MDR Mode, which indicates that the line cards will not be restarted or reloaded and maximum outage time will be 0 to 3 seconds.

Forcing a Switchover from Active to Standby

Issue the **issu runversion** command to force a switchover between the active and standby supervisor engines or RSPs. The standby supervisor engine or RSP, which has the new software image loaded, becomes active. The previously active supervisor engine or RSP becomes the standby and boots with the old software image (in case the software upgrade needs to be aborted and the old image restored).

```
Router# issu runversion 6

This command will reload the Active unit.  Proceed ? [confirm]
*Oct 27 21:43:18.901: %SYS-SP-3-LOGGER_FLUSHING: System pausing to ensure console
debugging output.

*Oct 27 21:43:18.901: %OIR-SP-6-CONSOLE: Changing console ownership to switch processor
```

(Switchover between supervisors occurs now. The previous standby becomes active and is running the new software version [c7600s72033]. The previous active, now standby, boots with the old software [oct22].)

```
*Oct 27 21:43:19.105: %SYS-SP-3-LOGGER_FLUSHED: System was paused for 00:00:00 to ensure
console debugging output.

*Oct 27 21:43:21.702: %SYS-SP-3-LOGGER_FLUSHING: System pausing to ensure console
debugging output.

***
*** --- SHUTDOWN NOW ---
***
```

```
*Oct 27 21:43:21.702: %SYS-SP-5-RELOAD: Reload requested by Delayed Reload. Reload Reason:
Reason unspecified.
*Oct 27 21:43:21.702: %OIR-SP-6-CONSOLE: Changing console ownership to switch processor

*Oct 27 21:43:22.067: %SYS-SP-3-LOGGER_FLUSHED: System was paused for 00:00:00 to ensure
console debugging output.

System Bootstrap, Version 8.1(3)
Copyright (c) 1994-2004 by cisco Systems, Inc.
Cat6k-Sup720/SP processor with 1048576 Kbytes of main memory

Autoboot executing command: "boot disk0:oct22"
Loading image, please wait ...

Initializing ATA monitor library...
```

(command output omitted, new active log)

```
Press RETURN to get started!

*Oct 27 21:30:06.611: STDBY: RP: Currently running ROMMON from S (Gold) region
*Oct 27 21:31:07.923: %SPANTREE-STDBY-5-EXTENDED_SYSID: Extended SysId enabled for type
vlan
*Oct 27 21:31:30.183: %SYS-STDBY-5-RESTART: System restarted --
Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ), Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Thu 26-Oct-06 03:21 by
*Oct 27 21:31:30.307: %SYS-STDBY-6-LOGGINGHOST_STARTSTOP: Logging to host 172.19.126.3
Port 514 started - CLI initiated
*Oct 27 21:43:22.067: %PFREDUN-SP-STDBY-6-ACTIVE: Initializing as ACTIVE processor
*Oct 27 21:43:22.071: %FABRIC-SP-STDBY-5-FABRIC_MODULE_ACTIVE: The Switch Fabric Module in
slot 6 became active.
*Oct 27 21:43:22.715: %SYS-SP-STDBY-3-LOGGER_FLUSHED: System was paused for 00:00:00 to
ensure console debugging output.
*Oct 27 21:43:24.363: %ISSU_PROCESS-SP-7-DEBUG: Initializing timers
*Oct 27 21:43:24.363: %ISSU_PROCESS-SP-7-DEBUG: rollback timer process has been started
*Oct 27 21:43:24.554: %C6KPWR-SP-4-PSOK: power supply 2 turned on.
*Oct 27 21:43:24.650: %OIR-SP-3-SOFT_RESET: Module 1 is being soft reset as a part of
swichover error recovery
*Oct 27 21:43:24.674: %LINK-SP-3-UPDOWN: Interface TenGigabitEthernet2/1, changed state to
down
*Oct 27 21:43:24.754: %OIR-SP-3-SOFT_RESET: Module 2 is being soft reset as a part of
swichover error recovery
*Oct 27 21:43:24.854: %OIR-SP-3-SOFT_RESET: Module 3 is being soft reset as a part of
swichover error recovery
*Oct 27 21:43:24.906: %OIR-SP-3-SOFT_RESET: Module 4 is being soft reset as a part of
swichover error recovery
*Oct 27 21:43:24.962: %OIR-SP-3-SOFT_RESET: Module 7 is being soft reset as a part of
swichover error recovery
```

(command output omitted)

```
*Oct 27 21:48:34.787: %SYS-SP-STDBY-6-BOOTTIME: Time taken to reboot after reload =  314
seconds
*Oct 27 21:50:31.059: %ISSU_PROCESS-SP-7-DEBUG: Peer state is [ STANDBY HOT ]; Please
issue the acceptversion command
*Oct 27 21:50:31.067: %PFREDUN-SP-STDBY-6-STANDBY: Ready for SSO mode
*Oct 27 21:50:31.067: %RF-SP-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
*Oct 27 21:50:31.251: %SYS-SP-STDBY-3-LOGGER_FLUSHED: System was paused for 00:00:00 to
ensure console debugging output.
```

```
Router# enable
```

Note At this point, the new active supervisor engine or RSP is running the new software image and the standby is running the old software image. You should verify the state of the active and standby supervisor engines or RSPs as shown in the following examples (**show redundancy** and **show issu state detail**).

```
Router# show redundancy

Redundant System Information :
------------------------------
       Available system uptime = 57 minutes
Switchovers system experienced = 1
             Standby failures = 0
        Last switchover reason = user initiated

                 Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 6
        Current Software state = ACTIVE
       Uptime in current state = 17 minutes
                 Image Version = Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ),
Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Thu 26-Oct-06 03:21 by
                         BOOT = disk0:c7600s72033,12;disk0:oct22,12
                    CONFIG_FILE =
                        BOOTLDR =
        Configuration register = 0x2102

Peer Processor Information :
----------------------------
              Standby Location = slot 5
        Current Software state = STANDBY HOT
       Uptime in current state = 15 minutes
                 Image Version = Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ),
Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Mon 23-Oct-06 02:43 by
                         BOOT = disk0:oct22,12
                    CONFIG_FILE =
                        BOOTLDR =
        Configuration register = 0x2102

router# show issu state detail

                         Slot = 6
                     RP State = Active
                    ISSU State = Run Version
                  Boot Variable = disk0:c7600s72033,12;disk0:oct22,12
                 Operating Mode = sso
                Primary Version = disk0:c7600s72033
              Secondary Version = disk0:oct22
                Current Version = disk0:c7600s72033
                 Variable Store = PrstVbl
                      ROMMON CV = [disk0:c7600s72033]
```

```
                Slot = 5
            RP State = Standby
          ISSU State = Run Version
         Boot Variable = disk0:oct22,12
       Operating Mode = sso
     Primary Version = disk0:c7600s72033
   Secondary Version = disk0:oct22
     Current Version = disk0:oct22
```

Note To complete the upgrade process, issue the **issu acceptversion** (optional) and **issu commitversion** commands (as described in the following sections).

Accepting the New Software Version and Stopping the Rollback Process (Optional)

You must either accept or commit the new software image, or the rollback timer will expire and stop the upgrade process. If that occurs, the software image reverts to the previous software version. The rollback timer is a safeguard to ensure that the upgrade process does not leave the router nonoperational.

The following command sequence shows how **issu acceptversion** stops the rollback timer to enable you to examine the functionality of the new software image. When you are satisfied that the new image is acceptable, issue the **issu commitversion** command to end the upgrade process.

```
Router# show issu rollback-timer
        Rollback Process State = In progress
      Configured Rollback Time = 45:00
       Automatic Rollback Time = 27:08

Router# issu acceptversion 6
% Rollback timer stopped. Please issue the commitversion command.
```

Now view the rollback timer to see that the rollback process has been stopped:

```
Router# show issu rollback-timer
        Rollback Process State = Not in progress
      Configured Rollback Time = 45:00
```

Committing the New Software to the Standby

Issue the **issu commitversion** command to load the new software image onto the standby supervisor engine or RSP and complete the software upgrade process. In the following example, the new image (c7600s72033) is loaded onto the standby supervisor engine in slot 5:

```
Router# issu commitversion 5

Building configuration...

*Oct 27 22:09:57.239: %PFINIT-SP-5-CONFIG_SYNC: Sync'ing the startup configuration to the
standby Router. [OK]
feeder#
*Oct 27 22:10:15.613: %OIR-SP-3-PWRCYCLE: Card in module 5, is being power-cycled (RF
request)
```

(The standby supervisor engine in slot 5 begins rebooting. It then loads the new image.)

```
*Oct 27 22:10:15.639: %PFREDUN-SP-6-ACTIVE: Standby processor removed or reloaded,
changing to Simplex mode
*Oct 27 22:13:40.723: SP: pf_redun_check_img_compatibility: MATRIX result is compatible!!!
Of course...
*Oct 27 22:13:41.731: %PFREDUN-SP-6-ACTIVE: Standby initializing for SSO mode
*Oct 27 22:13:42.027: %SYS-SP-3-LOGGER_FLUSHED: System was paused for 00:00:00 to ensure
console debugging output.
*Oct 27 22:13:44.999: %PFINIT-SP-5-CONFIG_SYNC: Sync'ing the startup configuration to the
standby Router.
*Oct 27 22:14:24.019: %FABRIC-SP-5-CLEAR_BLOCK: Clear block option is off for the fabric
in slot 5.
*Oct 27 22:14:24.115: %FABRIC-SP-5-FABRIC_MODULE_BACKUP: The Switch Fabric Module in slot
5 became standby
```

(command output omitted)

```
*Oct 27 22:15:23.310: %SYS-SP-STDBY-5-RESTART: System restarted --
Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ), Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Thu 26-Oct-06 03:49 by
*Oct 27 22:15:44.751: %PFREDUN-SP-STDBY-6-STANDBY: Ready for SSO mode
*Oct 27 22:15:45.135: %RF-SP-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
*Oct 27 22:15:44.935: %SYS-SP-STDBY-3-LOGGER_FLUSHED: System was paused for 00:00:00 to
ensure console debugging output.
```

Note The software upgrade process is now complete. Both the active and standby supervisor engines or RSPs are running the new software version.

Verifying the Software Installation

You should verify the status of the software upgrade. If the upgrade was successful, both the active and standby supervisor engines or RSPs are running the new software version.

```
Router# show redundancy

Redundant System Information :
------------------------------
       Available system uptime = 1 hour, 13 minutes
Switchovers system experienced = 1
              Standby failures = 1
        Last switchover reason = user initiated

                 Hardware Mode = Duplex
    Configured Redundancy Mode = sso
     Operating Redundancy Mode = sso
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 6
        Current Software state = ACTIVE
       Uptime in current state = 33 minutes
                 Image Version = Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ),
Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Thu 26-Oct-06 03:21 by
                         BOOT = disk0:c7600s72033,12;disk0:oct22,1;
                  CONFIG_FILE =
```

```
                        BOOTLDR =
        Configuration register = 0x2102

Peer Processor Information :
----------------------------
              Standby Location = slot 5
        Current Software state = STANDBY HOT
       Uptime in current state = 4 minutes
                 Image Version = Cisco IOS Software (c7600s72033-ADVENTERPRISEK9_WAN_MZ),
Version 12.2
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Thu 26-Oct-06 03:21 by
                          BOOT = disk0:c7600s72033,12;disk0:oct22,1;
                   CONFIG_FILE =
                        BOOTLDR =
        Configuration register = 0x2102


router# show issu state detail

                          Slot = 6
                     RP State = Active
                    ISSU State = Init
                  Boot Variable = disk0:c7600s72033,12;disk0:oct22,12
                 Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:c7600s72033
                Variable Store = PrstVbl
                     ROMMON CV = [disk0:c7600s72033]

                          Slot = 5
                     RP State = Standby
                    ISSU State = Init
                  Boot Variable = disk0:c7600s72033,12;disk0:oct22,12
                 Operating Mode = sso
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = disk0:c7600s72033
```

Aborting the Upgrade Process

You can manually abort the software upgrade at any stage by issuing the **issu abortversion** command. The upgrade process also aborts on its own if the software detects a failure.

If you abort the process after you issue the **issu loadversion** command, the standby supervisor engine or RSP is reset and reloaded with the original software.

The following is an example of the **issu abortversion** *slot image* command that shows how to abort the software upgrade process:

```
Router# issu abortversion 6 c7600s72033
```

Note Before you issue the **issu abortversion** command, make sure that the standby supervisor engine or RSP is Up (STANDBY HOT [in SSO] or COLD [in RPR]).

Upgrading a Non-eFSU Image to an eFSU Image

If the new Cisco IOS software image does not support eFSU, you must manually upgrade the software image. To do so, you must upgrade the software image on the standby supervisor engine or RSP and then perform a manual switchover so that the standby takes over processing with the new image. You can then upgrade the software image on the previously active, and now standby, supervisor engine or RSP. For instructions, see the "Performing a Fast Software Upgrade" section on page 7-8.

Command Reference

All of the standard ISSU commands are supported on Cisco 7600 series routers. For information about these commands, see the *Cisco IOS In Service Software Upgrade Process* document at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/newft/122srb33/sbisefsu.htm

CHAPTER 7

Configuring RPR and RPR+ Supervisor Engine Redundancy

This chapter describes how to configure supervisor engine redundancy using route processor redundancy (RPR) and RPR+.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- RPR and RPR+ support IPv6 multicast traffic.
- Release 12.2SR does not support single router mode (SRM) with stateful switchover (SSO).

This chapter contains these sections:

Understanding RPR and RPR+

These sections describe supervisor engine redundancy using RPR and RPR+:

Supervisor Engine Redundancy Overview

Note See the "Supervisor Engine Redundancy Guidelines and Restrictions" section on page 7-4 for important information about supervisor engine redundancy.

Cisco 7600 series routers support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. Cisco 7600 series routers support these redundancy modes:

- RPR—Supports a switchover time of 2 or more minutes.
- Route processor redundancy plus (RPR+)—Supports a switchover time of 30 or more seconds.

The following events cause a switchover:

- A hardware failure on the active supervisor engine
- Clock synchronization failure between supervisor engines
- A manual switchover

RPR Operation

RPR supports the following features:

- Auto-startup and bootvar synchronization between active and redundant supervisor engines
- Hardware signals that detect and decide the active or redundant status of supervisor engines
- Clock synchronization every 60 seconds from the active to the redundant supervisor engine
- A redundant supervisor engine that is booted but not all subsystems are up: if the active supervisor engine fails, the redundant supervisor engine become fully operational
- An operational supervisor engine present in place of the failed unit becomes the redundant supervisor engine
- Support for fast software upgrade (FSU) (See the "Performing a Fast Software Upgrade" section on page 7-8.)

When the router 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 MSFC and PFC on the redundant supervisor engine come out of reset but are not operational.

In a switchover, the redundant supervisor engine become fully operational and the following occurs:

- All switching modules power up again
- Remaining subsystems on the MSFC (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.

RPR+ Operation

When RPR+ mode is used, the redundant supervisor engine is fully initialized and configured, which shortens the switchover time. The active supervisor engine checks the image version of the redundant supervisor engine when the redundant supervisor engine comes online. If the image on the redundant supervisor engine does not match the image on the active supervisor engine, RPR redundancy mode is used.

With RPR+, the redundant supervisor engine is fully initialized and configured, which shortens the switchover time if the active supervisor engine fails or if a manual switchover is performed.

When the router is powered on, RPR+ runs between the two supervisor engines. The supervisor engine that boots first becomes the active supervisor engine. The Multilayer Switch Feature Card and Policy Feature Card become fully operational. The MSFC and PFC on the redundant supervisor engine come out of reset but are not operational.

RPR+ enhances RPR by providing the following additional benefits:

- Reduced switchover time

  Depending on the configuration, the switchover time is 30 or more seconds.

- Installed modules are not reloaded

  Because both the startup configuration and the running configuration are continually synchronized from the active to the redundant supervisor engine, installed modules are not reloaded during a switchover.

- Online insertion and removal (OIR) of the redundant supervisor engine

  RPR+ allows OIR of the redundant supervisor engine for maintenance. When the redundant supervisor engine is inserted, the active supervisor engine detects its presence and begins to transition the redundant supervisor engine to fully initialized state.

- Synchronization of OIR events
- Manual user-initiated switchover using the **redundancy force-switchover** command

Supervisor Engine Configuration Synchronization

These sections describe supervisor engine configuration synchronization:

- RPR Supervisor Engine Configuration Synchronization, page 7-3
- RPR+ Supervisor Engine Configuration Synchronization, page 7-4

Note Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the router 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 and with RPR+, reload the redundant supervisor engine and MSFC.

RPR Supervisor Engine Configuration Synchronization

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.

RPR+ Supervisor Engine Configuration Synchronization

With RPR+ mode, the following operations trigger configuration synchronization:

- When a redundant supervisor engine first comes online, the startup-config file is copied from the active supervisor engine to the redundant supervisor engine. This synchronization overwrites any existing startup configuration file on the redundant supervisor engine.
- When configuration changes occur during normal operation, redundancy performs an incremental synchronization from the active supervisor engine to the redundant supervisor engine. Redundancy synchronizes user-entered CLI commands incrementally line-by-line from the active supervisor engine to the redundant supervisor engine.

Even though the redundant supervisor engine is fully initialized, it only interacts with the active supervisor engine to receive incremental changes to the configuration files as they occur. You cannot enter CLI commands on the redundant supervisor engine.

Supervisor Engine Redundancy Guidelines and Restrictions

These sections describe supervisor engine redundancy guidelines and restrictions:

Redundancy Guidelines and Restrictions

These guidelines and restrictions apply to RPR and RPR+ redundancy modes:

- With a Supervisor Engine 720, if all the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on both supervisor engines, which ensures that all modules are operating in dCEF mode and simplifies switchover to the redundant supervisor engine. (CSCec05612)

  Post SRE release, the uplink ports are also enabled in dcef mode for RSP720-10G supervisor engine. The other supervisor engines continue to have the uplink ports disabled.
- Supervisor engine redundancy does not provide supervisor engine mirroring or supervisor engine load balancing. Only one supervisor engine is active at any one time.
- Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the router 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 and with RPR+, reload the redundant supervisor engine and MSFC.
- 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.

RPR+ Guidelines and Restrictions

These guidelines and restrictions apply to RPR+:

- Network services are disrupted until the redundant supervisor engine takes over and the router recovers.
- The Forwarding Information Base (FIB) tables are cleared on a switchover. As a result, routed traffic is interrupted until route tables reconverge.
- Static IP routes are maintained across a switchover because they are configured from entries in the configuration file.
- Information about dynamic states maintained on the active supervisor engine is not synchronized to the redundant supervisor engine and is lost on switchover.

  These are examples of dynamic state information that is lost at switchover:

  - Frame Relay Switched Virtual Circuits (SVCs)

  Note Frame Relay-switched DLCI information is maintained across a switchover because Frame Relay-switched DLCI configuration is in the configuration file.

  - All terminated PPP sessions
  - All ATM SVC information
  - All terminated TCP and other connection-oriented Layer 3 and Layer 4 sessions
  - BGP sessions
  - All Automatic Protection System (APS) state information
- Both supervisor engines must run the same version of Cisco IOS software. If the supervisor engines are not running the same version of Cisco IOS software, the redundant supervisor engine comes online in RPR mode.
- Supervisor engine redundancy does not support nondefault VLAN data file names or locations. Do not enter the **vtp file** *file_name* command on a router that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- Supervisor engine redundancy does not support configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy (see Chapter 14, “Configuring VLANs”).

Hardware Configuration Guidelines and Restrictions

For redundant operation, the following guidelines and restrictions must be met:

- Cisco IOS software running on the supervisor engine and the MSFC supports redundant configurations where the supervisor engines and MSFC routers are identical. If they are not identical, one will boot first and become active and hold the other supervisor engine and MSFC in a reset condition.
- Each supervisor engine must have the resources to run the router 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 7-9).

Note If a newly installed redundant supervisor engine has the Catalyst operating system installed, remove the active supervisor engine and boot the router 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 *Cisco IOS Configuration Fundamentals Configuration Guide* for more information).

Note There is no support for booting from the network.

Configuration Mode Restrictions

The following configuration restrictions apply during the startup synchronization process:

- You cannot perform configuration changes during the startup (bulk) synchronization. If you attempt to make configuration changes during this process, the following message is generated:

```
Config mode locked out till standby initializes
```

- If configuration changes occur at the same time as a supervisor engine switchover, these configuration changes are lost.

Configuring Supervisor Engine Redundancy

These sections describe how to configure supervisor engine redundancy:

Configuring Redundancy

To configure redundancy, perform this task:

	Command	Purpose
Step 1	`Router(config)# redundancy`	Enters redundancy configuration mode.
Step 2	`Router(config-red)# mode { rpr | rpr-plus}`	Configures RPR or RPR+. When this command is entered, the redundant supervisor engine is reloaded and begins to work in RPR or RPR+ mode.
Step 3	`Router# show running-config`	Verifies that RPR or RPR+ 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-plus
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 Plus
Redundancy Mode (Configured)  = Route Processor Redundancy Plus
     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 software 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 software 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 **bootflash:** device on a redundant supervisor engine:

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

Use the following command to copy a file to the **bootflash:** device on a redundant MSFC:

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

CHAPTER 8

Configuring Interfaces

This chapter describes how to configure interfaces on the Cisco 7600 series routers. It contains these sections:

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

Understanding Interface Configuration

Many features in the software are enabled on a per-interface basis. Each interface corresponds to a port on a module installed in a particular slot on the router:

- Slot number—The slot in which the module is installed. On the Cisco 7600 series router, slots are numbered starting with 1, from top to bottom or right to left.
- Port number—The physical port number on the module. On the Cisco 7600 series router, the port numbers always begin with 1. When facing the rear of the router, ports are numbered from the left to the right or top to bottom.

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.

Several LAN and WAN protocols (such as ATM and Frame Relay) support subinterfaces. Subinterfaces are virtual interfaces that are associated with a physical hardware interface. Each subinterface can be configured to support a different network. Thus, subinterfaces enable a single physical interface to support several networks.

For additional information on configuring interfaces on Cisco routers, see the *Cisco IOS Interface Configuration Guide* at this URL:

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

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 Enter the **interface** command. Specify interface type, along with the slot number/port number to identify the interface to configure. For example, to configure the interface for port 5 on the Fast Ethernet card installed in slot 5 (that is, FastEthernet port 5/5), enter:

```
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 3 After each **interface** command, enter the appropriate interface configuration commands to configure the interface. These interface configuration commands 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 exit interface configuration mode.

Note You can use **show interfaces** EXEC command to see a list of the interfaces on the router.

You can use the **show hardware** EXEC command to see a list of system software and hardware.

Step 4 After you configure an interface, you can check its status by using the EXEC **show** commands listed in "Monitoring and Maintaining Interfaces" section on page 8-15.

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:

- For information about macros, see the "Defining and Using Interface-Range Macros" section on page 8-4.
- You can enter up to five comma-separated ranges.
- You need not enter spaces before or after the comma.
- You need not add a space between the interface numbers and the dash when using the **interface range** command.
- The **no interface range** command supports VLAN interfaces.

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

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 Cisco 7600 series router 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 router automatically sets the duplex mode to auto.
- If you enter the **no speed** command, the router 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 router prompts you to also configure the duplex mode on the port.

Note Cisco 7600 series routers 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 8-1 shows the four possible link negotiation configurations and the resulting link status for each configuration.

Table 8-1 Link Negotiation Configuration and Possible Link Status

Link Negotiation State		Link Status	
Local Port	**Remote Port**	**Local Port**	**Remote Port**
Off	Off	Up	Up
On	On	Up	Up
Off	On	Up	Down
On	Off	Down	Up

To configure link negotiation on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface gigabitethernet` *slot/port*	Selects the port to be configured.
Step 2	`Router(config-if)# speed nonegotiate`	Disables link negotiation.
	`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 8-8
- Configuring MTU Sizes, page 8-10

Caution The following switching modules support a maximum ingress frame size of 8092 bytes:

- WS-X6516-GE-TX when operating at 100 Mbps
- WS-X6148-RJ-45, WS-X6148-RJ-45V and WS-X6148-RJ21, WS-X6148-RJ21V
- WS-X6248-RJ-45 and WS-X6248-TEL
- WS-X6248A-RJ-45 and WS-X6248A-TEL
- WS-X6348-RJ-45, WS-X6348-RJ45V and WS-X6348-RJ-21, WX-X6348-RJ21V

When jumbo frame support is configured, these modules drop ingress frames larger than 8092 bytes.

Note The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX do not support jumbo frames.

Understanding Jumbo Frame Support

These sections describe jumbo frame support:

- Jumbo Frame Support Overview, page 8-9
- Ethernet Ports, page 8-9
- VLAN Interfaces, page 8-10

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 MSFC.
- Jumbo frame support does not fragment bridged traffic.

Bridged and Routed Traffic Size Check at Ingress 10, 10/100, and 100 Mbps Ethernet and 10-Gigabit Ethernet Ports

Jumbo frame support compares ingress traffic size with the global LAN port MTU size at ingress 10, 10/100, and 100 Mbps Ethernet and 10-Gigabit Ethernet LAN ports that have a nondefault MTU size configured. The port drops traffic that is oversized. You can configure the global LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-11).

Bridged and Routed Traffic Size Check at Ingress Gigabit Ethernet Ports

Gigabit Ethernet LAN ports configured with a nondefault MTU size accept frames containing packets of any size larger than 64 bytes. With a nondefault MTU size configured, Gigabit Ethernet LAN ports do not check for oversize ingress frames.

Routed Traffic Size Check on the PFC

For traffic that needs to be routed, Jumbo frame support on the PFC compares traffic sizes to the configured MTU sizes and provides Layer 3 switching for jumbo traffic between interfaces configured with MTU sizes large enough to accommodate the traffic. Between interfaces that are not configured with large enough MTU sizes, if the "do not fragment bit" is not set, the PFC sends the traffic to the MSFC to be fragmented and routed in software. If the "do not fragment bit" is set, the PFC drops the traffic.

Bridged and Routed Traffic Size Check at Egress 10, 10/100, and 100 Mbps Ethernet Ports

10, 10/100, and 100 Mbps Ethernet LAN ports configured with a nondefault MTU size transmit frames containing packets of any size larger than 64 bytes. With a nondefault MTU size configured, 10, 10/100, and 100 Mbps Ethernet LAN ports do not check for oversize egress frames.

Bridged and Routed Traffic Size Check at Egress Gigabit Ethernet and 10-Gigabit Ethernet Ports

Jumbo frame support compares egress traffic size with the global egress LAN port MTU size at egress Gigabit Ethernet and 10-Gigabit Ethernet LAN ports that have a nondefault MTU size configured. The port drops traffic that is oversized. You can configure the global LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-11).

Ethernet Ports

These sections describe configuring nondefault MTU sizes on Ethernet ports:

Ethernet Port Overview

Configuring a nondefault MTU size on a 10, 10/100, or 100 Mbps Ethernet port limits ingress packets to the global LAN port MTU size and permits egress traffic of any size larger than 64 bytes.

Configuring a nondefault MTU size on a Gigabit Ethernet port permits ingress packets of any size larger than 64 bytes and limits egress traffic to the global LAN port MTU size.

Configuring a nondefault MTU size on a 10-Gigabit Ethernet port limits ingress and egress packets to the global LAN port MTU size.

Configuring a nondefault MTU size on an Ethernet port limits routed traffic to the configured MTU size.

You can configure the MTU size on any Ethernet port.

Layer 3 Ethernet Ports

On a Layer 3 port, you can configure an MTU size on each Layer 3 Ethernet port that is different than the global LAN port MTU size.

Note Traffic through a Layer 3 Ethernet LAN port that is configured with a nondefault MTU size is also subject to the global LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-11).

Layer 2 Ethernet Ports

On a Layer 2 port, you can only configure an MTU size that matches the global LAN port MTU size (see the "Configuring the Global Egress LAN Port MTU Size" section on page 8-11).

VLAN Interfaces

You can configure a different MTU size on each Layer 3 VLAN interface. Configuring a nondefault MTU size on a VLAN interface limits traffic to the nondefault MTU size. You can configure the MTU size on VLAN interfaces to support jumbo frames.

Configuring MTU Sizes

These sections describe how to configure MTU sizes:

Configuring the MTU Size

To configure the MTU size, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{vlan vlan_ID} | {{type[1] slot/port} | {port-channel port_channel_number} slot/port}}`	Selects the interface to configure.
Step 2	`Router(config-if)# mtu mtu_size`	Configures the MTU size.
	`Router(config-if)# no mtu`	Reverts to the default MTU size (1500 bytes).

	Command	Purpose
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show running-config interface [{gigabitethernet | tengigabitethernet} slot/port]`	Displays the running configuration.

1. *type* = **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 8-11).

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 Cisco 7600 series routers 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 up and 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.
- WS-X6502-10GE is not supported.
- Both fiber and copper media are supported for 10 Gbps ports.
- Media-only changes are detected.

Table 8-2 lists the time delay that occurs before notification of a link change.

Table 8-2 Port Debounce Timer Delay Time

Port Type	Debounce Timer Disabled	Debounce Timer Enabled
Ports operating at 10 Mpbs or 100 Mbps	300 milliseconds	3100 milliseconds
Ports operating at 1000 Mpbs or 10 Gbps over copper media	300 milliseconds	3100 milliseconds
Ports operating at 1000 Mpbs or 10 Gbps over fiber media except WS-X6502-10GE	10 milliseconds	100 milliseconds
WS-X6502-10GE 10-Gigabit ports	1000 milliseconds	3100 milliseconds

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 Cisco 7600 series routers 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 *Cisco 7600 Series Router Installation Guide*.

When a module has been removed or installed, the Cisco 7600 series router 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 router 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 router 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

Following is a list of several of the commands that you can use to monitor and maintain interfaces on the Cisco 7600 series router.

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.
`Router(config)# interface type slot/port` `Router(config-if)# shutdown`	Shuts down the specified interface, which disables all functions on the interface and shows the interface as unavailable. In addition, the interface is not included in any routing updates.
`Router(config)# interface type slot/port` `Router(config-if)# no shutdown`	Reenables an interface that is shut down.
`Router# clear counters [{type[1] slot/port} | {vlan vlan_ID} | {port-channel channel_ID}]`	Clears interface counters. If you do not specify an optional interface type, the command clears all current counters for the specified interface. **Note** This command clears counters displayed with the EXEC **show interfaces** command, not counters retrieved using SNMP.
`Router# clear interface type[1] slot/port`	Resets an interface.

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

Note

- For detailed command syntax and usage information, see the *Cisco 7600 Series Cisco IOS Command Reference* at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sr/cmdref/index.htm

- For additional information on monitoring and maintaining interfaces on Cisco routers, see the Release 12.2 *Cisco IOS Interface Configuration Guide* at this URL:

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

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 router, 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 12.2SX on the Supervisor Engine 720, Supervisor Engine 32, and Supervisor Engine 2* 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 9

Configuring a Supervisor Engine 32

This chapter describes how to configure a Supervisor Engine 32 in a Cisco 7600 series router. This chapter contains these sections:

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- With Cisco IOS software, this is the minimum required Supervisor Engine 32 memory:
  - 512 MB DRAM on the Supervisor Engine 32
  - 512 MB DRAM on the MSFC2A
- The Supervisor Engine 32 is supported in the WS-6503 and WS-6503-E (3-slot) chassis, but not the Cisco 7603 chassis.
- With a 4-slot chassis, install the Supervisor Engine 32 in either slot 1 or 2.
- With a 6-slot or a 9-slot chassis, install the Supervisor Engine 32 in either slot 5 or 6.
- With a 13-slot chassis, install the Supervisor Engine 32 in either slot 7 or 8.
- The Supervisor Engine 32 has a PFC3B and operates in PFC3B mode.
- The Supervisor Engine 32 does not support switch fabric connectivity.
- For information about the hardware and software features supported by the Supervisor Engine 32, see the *Release Notes for Cisco IOS Release 12.2SX on the Supervisor Engine 720, Supervisor Engine 32, and Supervisor Engine 2* at this URL:
  http://www.cisco.com/en/US/docs/ios/12_2sr/release/notes/122SRrn.html#wp4769974

Flash Memory on a Supervisor Engine 32

The Supervisor Engine 32 has:

- **disk0:**—One external CompactFlash Type II slot (supports CompactFlash Type II Flash PC cards)
- **sup-bootdisk:**—256 MB internal CompactFlash Flash memory (from ROMMON, it is **bootdisk:**)

Both Sup32 modules require a minimum of 128-MB bootflash to run Release 12.2SRB and later releases. A CompactFlash (CF) adapter with 512-MB bootflash is available for Sup32 modules in Release 12.2(18)SXF and later releases. Use the Cisco part number CF-ADAPTER= for ordering.

Supervisor Engine 32 Ports

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

WS-SUP32-GE-3B ports 1 through 8 have small form-factor pluggable (SFP) connectors and port 9 is a 10/100/1000 Mbps RJ-45 port.

WS-SUP32-10GE ports 1 and 2 are 10 Gigabit Ethernet ports that accept XENPAKs and port 3 is a 10/100/1000 Mbps RJ-45 port.

CHAPTER 10

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 Cisco 7600 series routers. 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 Cisco' 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- To configure Layer 3 interfaces, see Chapter 21, "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 Cisco 7600 series routers:

Understanding Layer 2 Ethernet Switching

These sections describe Layer 2 Ethernet switching:

Layer 2 Ethernet Switching Overview

Cisco 7600 series routers support simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

Cisco 7600 series routers 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 Cisco 7600 series router 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 router considers each LAN port to be an individual segment. When stations connected to different LAN ports need to communicate, the router 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 router maintains an address table. When a frame enters the router, it associates the MAC address of the sending network device with the LAN port on which it was received.

Building the Address Table

Cisco 7600 series routers build the address table by using the source address of the frames received. When the router 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 router adds its relevant source address and port ID to the address table. The router 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 router 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 Cisco 7600 series routers:

Trunking Overview

Note For information about VLANs, see Chapter 14, "Configuring VLANs."

A trunk is a point-to-point link between the router 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 12, "Configuring EtherChannels."

Ethernet trunk ports support several trunking modes (see Table 10-2 on page 10-4). 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 13, "Configuring VTP."

Encapsulation Types

Table 10-1 lists the Ethernet trunk encapsulation types.

Table 10-1 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 10-3).

Table 10-1 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 10-2 lists the Layer 2 LAN port modes and describes how they function on LAN ports.

Table 10-2 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 10-3 shows the Layer 2 LAN port default configuration.

Table 10-3 Layer 2 LAN Interface Default Configuration

Feature	Default
Interface mode:	
• Before entering the **switchport** command	Layer 3 (unconfigured)
• After entering the **switchport** command	**switchport mode dynamic desirable**
Trunk encapsulation	**switchport trunk encapsulation negotiate**
Allowed VLAN range	VLANs 1 to 4094, except reserved VLANs (see Table 14-1 on page 14-2)
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 router to a non-Cisco router through an 802.1Q trunk, the MST of the non-Cisco router and the native VLAN spanning tree of the Cisco router 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 router 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 Cisco 7600 series routers:

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).

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, page 10-8
- Configuring the Layer 2 Trunk to Use DTP, page 10-9
- Configuring the Layer 2 Trunk Not to Use DTP, page 10-9
- Configuring the Access VLAN, page 10-10
- Configuring the 802.1Q Native VLAN, page 10-10

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 10-7 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 10-9) 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 10-12 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 10-7 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 10-2 on page 10-4 for information about trunking modes.

Note Complete the steps in the "Completing Trunk Configuration" section on page 10-12 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 10-7 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 10-8).
- To support the **switchport nonegotiate** command, you must enter the **switchport mode trunk** command.
- Enter the **switchport mode dynamic trunk** command. See Table 10-2 on page 10-4 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 10-8) 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 10-9).

Note Complete the steps in the "Completing Trunk Configuration" section on page 10-12 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 10-7 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 14-1 on page 14-2).
`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 10-12 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 10-7 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 14-1 on page 14-2).
- The access VLAN is not automatically used as the native VLAN.

Note Complete the steps in the "Completing Trunk Configuration" section on page 10-12 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 10-7 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 10-12 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 10-7 before performing the tasks in this section.

To configure the list of prune-eligible VLANs on the Layer 2 trunk, perform this task:

Command	Purpose
`Router(config-if)# switchport trunk pruning vlan {none |{{add | except | remove} vlan[,vlan[,vlan[,...]]]}}`	(Optional) Configures the list of prune-eligible VLANs on the trunk (see the "Understanding VTP Pruning" section on page 13-5).
`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 14-1 on page 14-2), 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 Cisco 7600 series routers 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 10-12 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 14-10).

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`	Configures the LAN port as a Layer 2 access port.
	`Router(config-if)# no switchport mode`	Reverts to the default switchport mode (**switchport mode dynamic desirable**).
Step 6	`Router(config-if)# switchport access vlan` *vlan_ID*	Places the LAN port in a VLAN. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 14-1 on page 14-2).
	`Router(config-if)# no switchport access vlan`	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.
- See the *Release Notes for Cisco IOS Release 12.2SX on the Supervisor Engine 720, Supervisor Engine 32, and Supervisor Engine 2* for a list of the modules that support custom IEEE 802.1Q EtherType field values.

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.

This example shows how to configure the EtherType field value to 0x1234:

```
Router (config-if)# switchport dot1q ethertype 1234
Router (config-if)#
```

CHAPTER 11

Configuring Flex Links

This chapter describes how to configure Flex Links on the Cisco 7600 series router.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

The chapter consists of these sections:

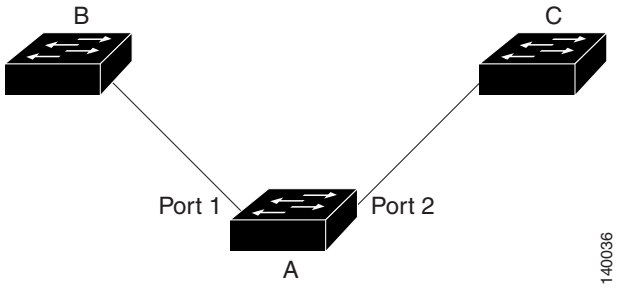
Understanding Flex Links

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.

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 11-1, ports 1 and 2 on router A are connected to uplink routers 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 router B; the link between port 2 (the backup link) and router C is not forwarding traffic. If port 1 goes down, port 2 comes up and starts forwarding traffic to router C. When port 1 comes back up, it goes into standby mode and does not forward traffic; port 2 continues to forward traffic.

Figure 11-1 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

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.
- The max number of Flexlinks supported on a Cicso 7600 Series Router is 16.
- 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 router, be sure that there are no Layer 2 loops in the network topology.

- Do not configure any STP features (for example, PortFast, BPDU Guard, and so forth) on Flex Links ports or the ports to which the links connect.

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 router 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 11-1 shows the privileged EXEC command for monitoring the Flex Links configuration.

Table 11-1 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 router and the state of each active and backup interface (up or standby mode).

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

CHAPTER 12

Configuring EtherChannels

This chapter describes how to configure EtherChannels on the Cisco 7600 series router Layer 2 or Layer 3 LAN ports.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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 Cisco 7600 series router supports a maximum of 128 EtherChannels.

You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Cisco 7600 series router. 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 Cisco 7600 series router 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 router, 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

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 12-1 lists the user-configurable EtherChannel modes.

Table 12-1 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	(Default for PAgP) 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.
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	(Default for LACP) 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.
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.

Table 12-2 provides a summary of these combinations.

Table 12-2 PaGP EtherChannel Modes

Router A	Router B	Result
auto mode	**auto** mode	No EtherChannel group is created.
auto mode	**desirable** mode	EtherChannel group is created.
desirable mode	**auto** mode	EtherChannel group is created.
desirable mode	**desirable** mode	EtherChannel group is created.

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.

Table 12-3 provides a summary of these combinations.

Table 12-3 LACP EtherChannel Modes

Router A	Router B	Result
passive mode	**passive** mode	No EtherChannel group is created.
passive mode	**active** mode	EtherChannel group is created.
active mode	**passive** mode	EtherChannel group is created.
active mode	**active** mode	EtherChannel group is created.

LACP uses the following parameters:

- LACP system priority—You must configure an LACP system priority on each router running LACP. The system priority can be configured automatically or through the command line interface (CLI) (see the "Configuring the LACP System Priority and System ID" section on page 12-10). LACP uses the system priority with the router 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 router.

- 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 12-8). 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.

  Note Port priority is only effective when it is configured on a device with an LACP system priority higher than the peer.

- LACP administrative key—LACP automatically configures an administrative key value equal to the channel group identification number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port's ability to aggregate with other ports is determined by these factors:
  - Port physical characteristics, such as data rate, duplex capability, and point-to-point or shared medium
  - Configuration restrictions that you establish

On ports configured to use LACP, LACP tries to configure the maximum number of compatible ports in an EtherChannel, up to the maximum allowed by the hardware (eight ports). If LACP cannot aggregate all the ports that are compatible (for example, the remote system might have more restrictive hardware limitations), then all the ports that cannot be actively included in the channel are put in hot standby state and are used only if one of the channeled ports fails. You can configure an additional 8 standby ports (total of 16 ports associated with the EtherChannel).

Understanding LACP 1:1 Redundancy

With Release 12.2(33)SRC and later, the LACP 1:1 redundancy feature provides an EtherChannel configuration with one active link and fast switchover to a hot standby link.

To use LACP 1:1 redundancy, you configure an LACP EtherChannel with two ports (one active and one standby). If the active link goes down, the EtherChannel stays up and the system performs fast switchover to the hot standby link. When the failed link becomes operational again, the EtherChannel performs another fast switchover to revert to the original active link.

For the LACP 1:1 redundancy feature to work correctly (especially the fast switchover capability) the feature needs to be enabled at both ends of the link.

Understanding Port-Channel Interfaces

Each EtherChannel has a numbered port-channel interface. 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 to which 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 router. 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 Cisco 7600 series routers, including the ports on the supervisor engine and a redundant supervisor engine.
- The WS-X6548-GE-TX and WS-X6548V-GE-TX switching modules support more than 1 Gbps of traffic per EtherChannel.

- 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 Inter-Switch Link Protocol (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 still compatible for the formation of an EtherChannel.
  - An EtherChannel will not form if protocol filtering is set differently on the LAN ports.
- You can configure a maximum of 256 port-channel interfaces, numbered from 1 to 256.
- 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 those LAN ports to which you apply the configuration.
- When quality of service (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.
- ES20 and ES+ cross-bundling is not supported and any LAN card and ES20/ES+ cross-bundling is also not supported.

Configuring EtherChannels

These sections describe how to configure EtherChannels:

Note Make sure that the LAN ports are configured correctly (see the "EtherChannel Feature Configuration Guidelines and Restrictions" section on page 12-5).

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 12-8).
- 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 12-8).
- 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 in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface port-channel number`	Creates the port-channel interface.
Step 2	`Router(config-if)# ip address ip_address mask`	Assigns an IP address and subnet mask to the EtherChannel.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show running-config interface port-channel group_number`	Verifies the configuration. *group-number*—1 through 256, up to a maximum of 256 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 12-7), 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 in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type slot/port*	Selects a LAN port to configure.
Step 2	`Router(config-if)# no ip address`	Ensures that there is no IP address assigned to the LAN port.
Step 3	`Router(config-if)# channel-protocol {lacp | pagp}`	(Optional) On the selected LAN port, restricts the **channel-group** command to the EtherChannel protocol configured with the **channel-protocol** command.
Step 4	`Router(config-if)# channel-group` *number* `mode {active | auto | desirable | on | passive}`	Configures the LAN port in a port channel and specifies the mode (see Table 12-1 on page 12-2). PAgP supports only the auto and desirable modes. LACP supports only the active and passive modes.
Step 5	`Router(config-if)# lacp port-priority` *priority_value*	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.

	Command	Purpose
Step 6	`Router(config-if)# end`	Exits configuration mode.
Step 7	`Router# show running-config interface` *type slot/port* `Router# show interfaces` *type slot/port* **etherchannel**	Verifies the configuration. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, **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 8-2 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 Interface
Fa5/2     d     U1/S1           1s       0        128        Any      0 5/2

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 router.

To configure the LACP system priority and system ID, perform this task in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# lacp system-priority` *priority_value*	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show lacp sys-id`	Verifies the configuration.

This example shows how to configure the LACP system priority:

```
Router# configure terminal
Router(config)# lacp system-priority 23456
Router(config)# end
Router(config)#
```

This example shows how to verify the configuration:

```
Router# show lacp sys-id
23456,0050.3e8d.6400
Router#
```

The system priority is displayed first, followed by the MAC address of the router.

Configuring EtherChannel Load Balancing

To configure EtherChannel load balancing, perform this task in global configuration mode:

	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. The load-balancing keywords indicate the following information: • **dst-ip**—Destination IP addresses • **dst-mac**—Destination MAC addresses • **dst-port**—Destination Layer 4 port • **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
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show etherchannel load-balance`	Verifies the configuration.

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 in interface configuration mode:

	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. Default is one.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show running-config interface type slot/port` `Router# show interfaces type slot/port etherchannel`	Verifies the configuration. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**.

Note Although the EtherChannel Min-Links feature works correctly when configured only on one end of an EtherChannel, for best results, configure the same number of minimum links on both ends of the EtherChannel.

This example shows how to configure port-channel interface 1 to be inactive if fewer than 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
```

Configuring LACP 1:1 Redundancy with Fast-Switchover

For the LACP 1:1 redundancy feature, the LACP EtherChannel must contain exactly two links, of which only one is active. The link with the lower port priority number (and therefore a higher priority) will be the active link, and the other link will be in a hot standby state. The LACP max-bundle must be set to 1.

To configure the LACP 1:1 redundancy feature, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface port-channel group_number`	Selects an LACP port channel interface.
Step 2	`Router(config-if)# lacp fast-switchover`	Enables the fast switchover feature for this EtherChannel.
Step 3	`Router(config-if)# lacp max-bundle 1`	Sets the maximum number of active member ports to 1.

	Command	Purpose
Step 4	`Router(config-if)# end`	Exits configuration mode.
Step 5	`Router# show running-config interface port-channel` group_number `Router# show interfaces` *type slot/port* **etherchannel**	Verifies the configuration. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**.

Note For the LACP 1:1 redundancy feature to work correctly, especially the fast switchover capability, the feature needs to be enabled at both ends of the EtherChannel.

This example shows how to configure an LACP EtherChannel with 1:1 redundancy. Because Fast Ethernet port 5/6 is configured with a higher port priority number (and therefore a lower priority) than the default of 32768, it will be the standby port.

```
Router# configure terminal
Router(config)# lacp system-priority 33000
Router(config)# interface range fastethernet 5/6 -7
Router(config-if)# channel-protocol lacp
Router(config-if)# channel-group 1 mode active
Router(config)# interface fastethernet 5/6
Router(config-if)# lacp port-priority 33000
Router(config)# interface port-channel 1
Router(config-if)# lacp fast-switchover
Router(config-if)# lacp max-bundle 1
Router(config-if)# end
```

CHAPTER 13

Configuring VTP

This chapter describes how to configure the VLAN Trunking Protocol (VTP) on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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 14, "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 Cisco 7600 series router is in VTP server mode and is in the no-management domain state until the router receives an advertisement for a domain over a trunk link or you configure a management domain.

If the router receives a VTP advertisement over a trunk link, it inherits the management domain name and the VTP configuration revision number. The router ignores advertisements with a different management domain name or an earlier configuration revision number.

If you configure the router as VTP transparent, you can create and modify VLANs but the changes affect only the individual router.

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 Cisco 7600 series router 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.

  Note In VTP version 3, manipulation of VLANs can be done only to primary servers.

- Client—VTP clients behave the same way as VTP servers, but you cannot create, change, or delete VLANs on a VTP client.
- Transparent—In VTP version 1, 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 and VTP version 3, transparent network devices do forward VTP advertisements that they receive out their trunking LAN ports.
- Off—In VTP off mode, a network device functions in the same manner as a VTP transparent device except that it does not forward VTP advertisements.

Note Cisco 7600 series routers automatically change from VTP server mode to VTP client mode if the router detects a failure while writing configuration to nonvolatile random-access memory (NVRAM). If this happens, the router 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 (Inter-Switch Link [ISL] and IEEE 802.1Q)
- Emulated LAN names (for ATM LAN Emulation Services [LANE])
- IEEE 802.10 Security Association Identifier (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 Versions

If you use VTP in your network, you must decide whether to use VTP version 1, version 2, or version 3.

Note If you are using VTP in a Token Ring environment, you must use version 2 or version 3.

VTP 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 14-1.
- 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.

VTP Version 3

VTP version 3 supports the following features not supported in version 1 or version 2:

- Hidden Password Support—VTP version 3 supports the option of configuring the password as **hidden** or **secret**.

  When the **hidden** keyword is specified, that password must be reentered if a takeover command is issued in the domain. The secret key generated from the password string is saved in the const_nvram:vlan.dat file. When configured with this option, the password does not appear in plain text in the configuration. Instead, the secret key associated with the password is saved in hexadecimal format in the running configuration. If the **hidden** keyword is not specified, the password is saved in clear text in the const_nvram:vlan.dat file as in VTP version 1 and VTP version 2.

  When the **secret** keyword is specified, the password secret key can be directly configured.

- Support for extended VLAN Database Propagation—In VTP version 2, VLAN configuration information is propagated only for VLANs numbered 1 to 1000. In VTP version 3, information also is propagated for extended-range VLANs (VLANs numbered 1006 to 4094).

- On Cisco 7600 series routers running VTP version 1, VTP version 2, or VTP version 3, default VLANs 1 and 1002 to 1005 cannot be modified.

Note VTP pruning continues to apply only to VLANs numbered 1 to 1000.

- Propagation of Any Database in a Domain—In addition to propagating VLAN database information, VTP can propagate Multiple Spanning Tree (MST) protocol database information.

- Disabling VTP—When VTP is disabled on a trunking port, it applies to all VTP instances on that port. When VTP is disabled globally, the setting applies to all the trunking ports in the system.

- In VTP version 1 and VTP version 2, the role of a VTP server is to back up the database to NVRAM and to allow the administrator to change database information. VTP version 3 introduces the roles of VTP Primary Server and VTP Secondary Server. A VTP Primary Server is used to update the database information. The updates sent out are honored by all the devices in the system. A VTP Secondary Server can only back up to its NVRAM the VTP configuration received via updates from the VTP Primary Server.

  The status of primary and secondary servers is a runtime status and is not a configurable option. By default, all devices are initiated as secondary servers. Primary server status is needed only when database updates are needed, and is obtained when the administrator issues a takeover message in the domain. (See the "Starting a Takeover" section on page 13-13.)

  Primary server status is lost upon reload of the device, or when switchover or domain parameters change. Secondary servers back up the configuration and continue to propagate it. Because of that, it is possible to have a working VTP domain without any primary servers.

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 13-1 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 Cisco 7600 series router (see the "Enabling VTP Pruning" section on page 13-9). You configure pruning on Layer 2 trunking LAN ports (see the "Configuring LAN Interfaces for Layer 2 Switching" section on page 10-6).

Figure 13-1 Flooding Traffic without VTP Pruning

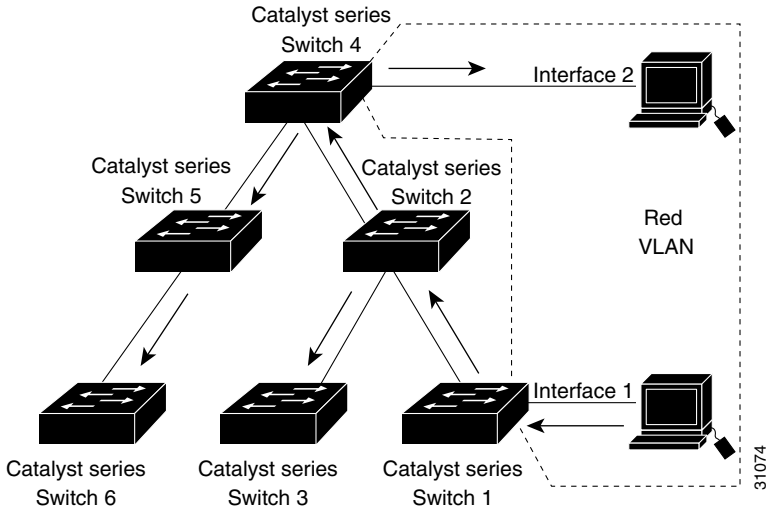
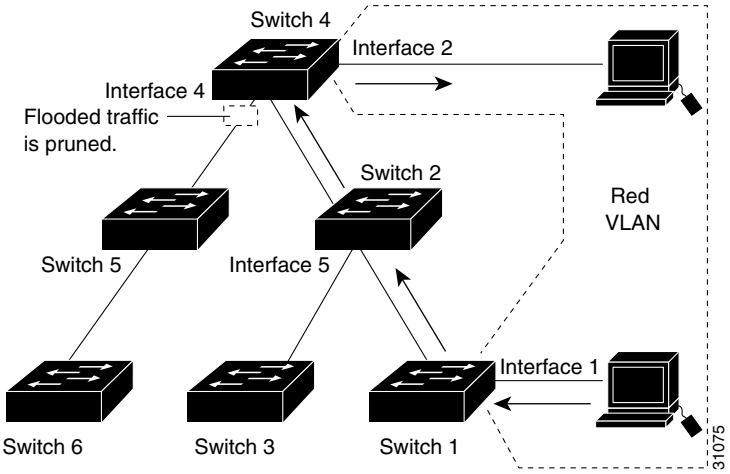


Figure 13-2 shows the same switched network with VTP pruning enabled. The broadcast traffic from Switch 1 is not forwarded to Switches 3, 5, and 6 because traffic for the Red VLAN has been pruned on the links indicated (port 5 on Switch 2 and port 4 on Switch 4).

Figure 13-2 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 LAN Interfaces for Layer 2 Switching" section on page 10-6). VTP pruning operates when a LAN port is trunking. You can set VLAN pruning eligibility whether VTP pruning is enabled or disabled for the VTP domain, whether any given VLAN exists or not, and whether the LAN port is currently trunking or not.

VTP Default Configuration

Table 13-1 shows the default VTP configuration.

Table 13-1 VTP Default Configuration

Feature	Default Value
VTP domain name	Null
VTP mode	Server
VTP version	Version 1
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 router 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.
- When a VTP version 3 device on a trunk port receives messages from a VTP version 2 device, it will send a scaled-down version of the VLAN database on that particular trunk in a VTP version 2 format. A VTP version 3 device will not send out VTP version 2 formatted packets on a trunk port unless it first receives VTP version 2 packets on that trunk.
- Even when a VTP version 3 device detects a VTP version 2 device on a trunk port, it will continue to send VTP version 3 packets in addition to VTP version 3 packets, to allow co-existence of two kinds of neighbors off the trunk.
- A VTP version 3 device will not accept configuration information from a VPT version 2 or version 1 device.
- Unlike in VPT version 2, when VTP is configured to be version 3, this will not configure all the version-3-capable devices in the domain to start behaving as VPT version 3 systems.
- When a VTP version 1 device, capable of version 2 or version 3, receives a VTP version 3 packet, the device is configured as a VTP version 2 device provided a VTP version 2 conflict does not exist.
- Devices that are only VTP version 1 capable cannot interoperate with VTP version 3 devices.
- In a Token Ring environment, you must enable VTP version 2 or version 3 for Token Ring VLAN switching to function properly.
- Two VPT version 3 regions can only communicate in transparent mode over a VTP version 1 or VTP version 2 region.
- 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 router. 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 router only, not on all network devices in the VTP domain.
- In VTP version 1 and version 2, VTP does not propagate configuration information for extended-range VLANs (VLAN numbers 1006 to 4094). You must configure extended-range VLANs manually on each network device.
- 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 Cisco 7600 series routers 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 10-11.

- The VLAN database is saved in the NVRAM file in a format compliant with the VTP version running on the system. Since older images supporting only VTP version 2 do not recognize the VTP version 3 file format, the NVRAM VLAN database information is lost if the system is downgraded from a new image supporting VTP to one that does not.

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, use these commands:

Command	Purpose
`Router(config)# vtp password` *password_string* `[hidden | secret]`	Sets a password, which can be from 8 to 64 characters long, for the VTP domain. In VTP version 3 the keywords **hidden** and **secret** are available. • If the **hidden** keyword is used, the secret key generated from the password string is saved in the const_nvram:vlan.dat file. If a takeover command is issued, that password must be reentered. • If the **secret** keyword is used, the password secret key can be directly configured. The **secret** password must contain 32 hexadecimal characters.
`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.

This example shows how to configure a **hidden** password:

```
Router# configure terminal
Router(config)# vtp password WATER hidden
Generating the secret associated to the password.
Router(config)#
```

This example shows how the password WATER is displayed when it is configured with the **hidden** keyword.

```
Router# show vtp password
VTP Password: 89914640C8D90868B6A0D8103847A733
Router#
```

Enabling VTP Pruning

To enable VTP pruning in the management domain, perform this task in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vtp pruning`	Enables VTP pruning in the management domain.
Step 2	`Router# show vtp status | include pruning`	(Optional) 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 10-11.

Enabling the VTP Version Number

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 or VTP version 3 for Token Ring VLAN switching to function properly on devices that support Token Ring interfaces.

To enable the VTP version, perform this task in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vtp version {1 | 2 | 3}`	Enables the VTP version.
Step 2	`Router# show vtp status | include {v1 | v2 | v3}`	(Optional) 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 in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vtp mode {client | server | transparent | off}`	Configures the VTP mode.

	Command	Purpose
Step 2	`Router(config)# vtp domain domain_name`	(Optional; for server mode only) Defines the VTP domain name, which can be up to 32 characters long. VTP server mode requires a domain name. If the router has a trunk connection to a VTP domain, the router 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`	(Optional) 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 router as a VTP server:

```
Router# configure terminal
Router(config)# vtp mode server
Setting device to VTP SERVER mode.
Router(config)# vtp domain Lab_Network
Setting VTP domain name to Lab_Network
Router(config)# end
Router#
```

This example shows how to configure the router as a VTP client:

```
Router# configure terminal
Router(config)# vtp mode client
Setting device to VTP CLIENT mode.
Router(config)# end
Router#
```

This example shows how to disable VTP on the router:

```
Router# configure terminal
Router(config)# vtp mode transparent
Setting device to VTP TRANSPARENT mode.
Router(config)# end
Router#
```

This example shows how to disable VTP on the router and to disable VTP advertisement forwarding:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# vtp mode off
Setting device to VTP OFF mode.
Router(config)# end
Router#
```

This example shows an example of the VTP configuration parameters when the device is running VTP version 1:

```
Router# show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : Lab_Network
VTP Pruning Mode                : Enabled
VTP Traps Generation            : Disabled
Device ID                       : 0016.9c6d.5300
```

```
Configuration last modified by 127.0.0.12 at 10-18-07 10:12:42
Local updater ID is 127.00.12 at 10-18-07 10:2:42

Feature VLAN:
--------------
VTP Operating Mode              : Server
Maximum number of existing VLANs : 5
Configuration Revision          : 1
MD5 digest                      : 0x92 0xF1 0xE8 0x52 0x2E ox5C 0x36 0x10 0x70 0x61 0xB8
                                0x24 0xB6 0x93 0x21 0x09
Router#
```

This example shows an example of the VTP configuration parameters when the device is running VTP version 2:

```
Router# show vtp status
VTP Version capable          : 1 to 3
VTP version running          : 2
VTP Domain Name              : Lab_Network
VTP Pruning Mode             : Disabled
VTP Traps Generation         : Disabled
Device ID                    : 0012.44dc.b800
Configuration lst modified by 127.0.0.12 at 10-18-07 10:38:45
Local updater ID is 127.0.0.12 on interface EO 0/0 (first interface found)

Feature VLAN:
--------------
VTP Operating Mode           : Server
Maximum VLANs supported locally: 1005
Number of existing VLANs     : 1005
Configuration Revision       : 1
MD5 digest                   : 0x2E 0x6B 0x99 0x58 0xA2 0x4F 0xD5 0x150x70 0x61 0xB8
                             0x24 0xB6 0x93 0x21 0x09
Router#
```

This example shows an example of the VTP configuration parameters when the device is running VTP version 3:

```
Router# show vtp status
VTP Version capable          : 1 to 3
VTP version running          : 3
VTP Domain Name              : Lab_Network
VTP Pruning Mode             : Disabled
VTP Traps Generation         : Disabled
Device ID                    : 0012.44dc.b800

Feature VLAN:
--------------
VTP Operating Mode              : Server
Number of existing VLANs        : 1005
Number of existing extended VLANs: 3074
Configuration Revision          : 18
Primary ID                      : 0012.4371.9ec0
Primary Description             :
Router#
```

Starting a Takeover

This process applies to VTP version 3 only. To start a takeover, perform this task:

Command	Purpose
`Router# vtp primary-server [vlan | mst]| [force]`	Changes the operational state of a router from a secondary to a primary server and advertises the configuration to the whole domain. (If the password for this device is configured with the **hidden** keyword, the user is prompted to re-enter it.) **Note** Using the **force** keyword overwrites the configuration of any conflicting servers. If not using the **force** keyword, you will be prompted for confirmation prior to proceeding with the takeover. Specify where to direct the takeover by selecting the appropriate feature (**vlan** or **mst**). If no feature is selected, the takeover is directed to the VLAN database.

This example shows how to start a takeover and direct it to the **vlan** database:

```
Router# vtp primary-server vlan
Enter VTP password:password
This system is becoming primary for feature vlan

VTP Feature Conf Revision Primary Server Device ID      Description
----------- ------------      -------------- -------- --------------------
MST         Yes  4        0012.4371.9ec0=0012.4371.9ec0 R1
Do you want to continue? (confirm)
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
```

Displaying VTP Devices in a Domain

To display information for all the VTP devices in a domain, perform this task in privileged EXEC mode:

Command	Purpose
Router# **show vtp devices** [**conflicts**]	Gathers and displays information for all the VTP devices in the domain. **Note** No information is gathered or displayed from routers set to **vtp modes off** or to **transparent** for a particular feature. The **conflicts** keyword (optional) displays the information of devices that have conflicting primary servers.

This example shows how to display information for VTP devices in a domain:

```
Router# show vtp devices
Retrieving information from the VTP domain, please wait for 5 seconds.
VTP Feature Conf Revision Primary Server Device ID         Device Description
----------- ---- -------- ------------------------------   ------------------
VLAN        No   18       0016.9c6d.5300 0012.011a.0d00    R2
VLAN        No   18       0016.9c6d.5300 0012.4371.9ec0    R1
MST         Yes  4        0012.4371.9ec0=0012.4371.9ec0    R1

Router#
```

CHAPTER 14

Configuring VLANs

This chapter describes how to configure VLANs on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter contains 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 19-2).

Cisco 7600 series routers 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 14-1 describes the VLAN ranges.

Table 14-1 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.
- 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 19-2).

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:

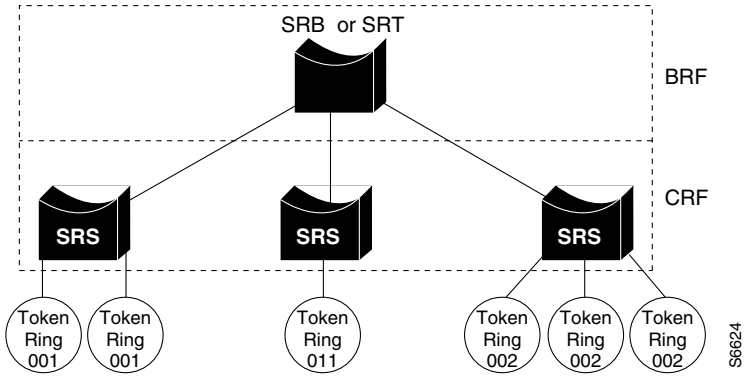
- Token Ring TrBRF VLANs, page 14-3
- Token Ring TrCRF VLANs, page 14-4

Note Cisco 7600 series routers do not support Inter-Switch Link (ISL)-encapsulated Token Ring frames. When a Cisco 7600 series router is configured as a VTP server, you can configure Token Ring VLANs from the router.

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 14-1). 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 14-1 Interconnected Token Ring TrBRF and TrCRF VLANs



For source routing, the Cisco 7600 series router 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 Interaction with Bridged Routed Encapsulation within an Automatic Protection Switching Group" section on page 14-8.

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

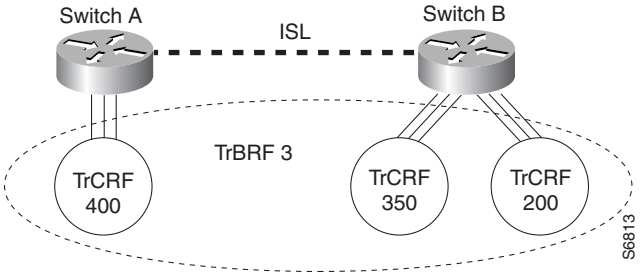
Token Ring TrCRF VLANs

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 14-2). The parent TrBRF acts as a multiport bridge, forwarding traffic between the undistributed TrCRFs.

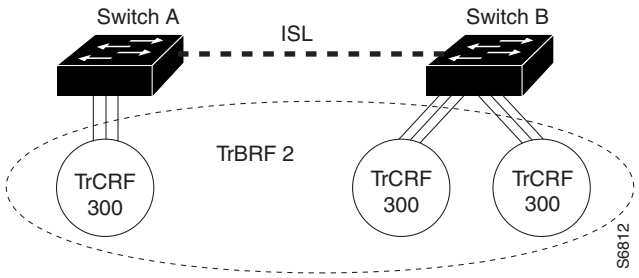
Note To pass data between rings located on separate network devices, you can associate the rings to the same TrBRF and configure the TrBRF for an SRB.

Figure 14-2 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 14-3), and traffic is passed between the default TrCRFs located on separate network devices if the network devices are connected through an ISL trunk.

Figure 14-3 Distributed TrCRF



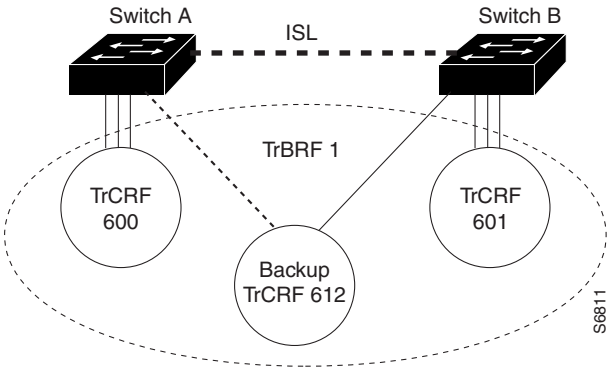
Within a TrCRF, source-route switching forwards frames based on either MAC addresses or route descriptors. The entire VLAN can operate as a single ring, with frames switched between ports within a single TrCRF.

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 14-4 illustrates the backup TrCRF.

Figure 14-4 Backup TrCRF



VLAN Default Configuration

Tables 14-2 through 14-6 show the default configurations for the different VLAN media types.

Table 14-2 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 14-3 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 14-3 FDDI VLAN Defaults and Ranges (continued)

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

Table 14-4 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 14-5 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 14-6 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 14-6 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 Interaction with Bridged Routed Encapsulation within an Automatic Protection Switching Group

In a Bridged Routed Encapsulation (BRE) scenario, an IP routed AAL5SNAP packet is bridged over the Ethernet side, adding a MAC header with a fake SRC MAC and configured distributed storage (DST). The ATM PVC traffic is relayed over an IP and not Ethernet. However, you cannot configure more than one virtual connection (VC) on the same VLAN. To configure more than one VC, customers configure two different VLANS on the protect and working interface of the Automatic Protection Switching (APS) group. This workaround is not a viable long- term solution because it results in high convergence time and an inefficient use of the VLANS. To resolve these limitations, you can use the BRE+APS feature to configure two VCs for the same VLAN, provided their parent interfaces too belong to the same Automatic Protection Switching (APS) group.

For information on configuring an APS group, see Cisco 7600 Series Router SIP, SSC, and SPA Software Configuration Guide at
http://www.cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/7600series/76cfgatm.html

Supported Line Cards

This feature is supported on the SIP-200 and SIP-400 line cards. For more information on the SIP implementation, see Cisco 7600 Series Router SIP, SSC, and SPA Software Configuration Guide at http://www.cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/7600series/sipspasw.html.

Requirements and Restrictions

Follow these requirements and restrictions when you configure the BRE+APS feature:

- You can configure BRE-Connect VLANS for two different VCs if the new VC:
  - belongs to the same APS group to which the first VC belongs.
  - does not belong to the same ATM interface as the first VC.
- Before you change the APS parameters of an interface (changing the APS group or removing the APS configurations), first ensure that the BRE configurations on the interface are removed.
- When you configure BRE on an ATM interface, you cannot configure a L2 IP address at the BRE end, but you can configure an IP address at the L3 non BRE end.

Table 14-7 Show command for ATM VLAN BRE

	Command	Purpose
Step 1	`Router(config)# show atm vlan bre`	Verifies the configuration and displays the status of the PVC. An Active VC is displayed as **UP** and an inactive VC as **DN** (down).

This example shows how to verify the configuration of BRE ATM VLAN:

```
Router# show atm vlan bre
Interface       Bre VCD            VPI/VCI   Vlan Learned MAC    Virtual MAC    State
ATM3/0/0.1      1                    0/11     100 0000.0000.0000 0000.0300.0001   UP
ATM3/0/0.2      2                    1/13     200 0000.0000.0000 0000.0300.0002   UP
ATM4/0/0.2      2                    1/13     300 0000.0000.0000 0000.0400.0002   DN
```

Warning Messages

Consider instances where you have configured APS on the main interface, and have configured BRE within a main interface and subinterface. The warning message "%ATM2/0/0 - Remove BRE configs on this interface before changing APS configs"appears when you attempt to modify the APS configurations in the main interface, without removing the BRE configurations first.

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 router that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- VLAN database mode, which was available in Release 12.2(18)SXD and earlier releases, is no longer supported. In addition, RPR+ redundancy does not support configurations entered in VLAN database mode. Use global configuration mode with RPR+ redundancy.
- Before you can create a VLAN, the Cisco 7600 series router must be in VTP server mode or VTP transparent mode. For information on configuring VTP, see Chapter 13, "Configuring VTP."
- The VLAN configuration is stored in the vlan.dat file, which is stored in nonvolatile memory. You can cause inconsistency in the VLAN database if you manually delete the vlan.dat file. If you want to modify the VLAN configuration or VTP, use the commands described in this guide and in the *Cisco 7600 Series Router Cisco IOS Command Reference* publication.
- To do a complete backup of your configuration, include the vlan.dat file in the backup.
- Cisco 7600 series routers do not support Token Ring or FDDI media. The router does not forward FDDI, FDDI-Net, TrCRF, or TrBRF traffic, but it can propagate the VLAN configuration through VTP.
- When a Cisco 7600 series router is configured as a VTP server, you can configure FDDI and Token Ring VLANs from the router.
- 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 *Cisco 7600 Series Router Cisco IOS Command Reference* publication.

VLAN Configuration Background Information

If the router is in VTP server or transparent mode (see the "Configuring VTP" section on page 13-8), 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 router 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 router boots, if the VTP domain name and VTP mode in the startup-config and vlan.dat files do not match, the router uses the configuration in the vlan.dat file.
- VLAN database mode, which was available in Release 12.2(18)SXD and earlier releases, is no longer supported.
- RPR+ redundancy does not support configurations entered in VLAN database mode. Use global configuration mode with RPR+ redundancy.

Creating or Modifying an Ethernet VLAN

User-configured VLANs have unique IDs from 1 to 4094, except for reserved VLANs (see Table 14-1 on page 14-2). 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 14-6 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*
	`Router(config)# no vlan` *vlan_ID*
	`Router(config-vlan)#`
	or
	`Router(vlan)# no 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).
		
		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:

- RPR+ redundancy does not support a configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy.
- Because Layer 3 ports and some software features require internal VLANs allocated from 1006 and up, configure extended-range VLANs starting with 4094.
- 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 router 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 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 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#
```

You can also use the **sh vlan free** and **sh vlan free summary** command to To list and view the total number of free vlans and display the vlan usage summary information in the system.

```
Router#show vlan free ?
  Summary  Total number of free vlans in the system
  |        Output modifiers
  <cr>

Router#show vlan free

Free VLANs
----------
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

Router#show vlan free ?
  Summary  Total number of free vlans in the system
  |        Output modifiers
Router#show vlan free summ
Router#show vlan free summary ?
  |  Output modifiers

Router#show vlan free summary

======= VLAN free/usage Summary =======

Total number of available vlans = 4094

Total number of free vlans = 4074

Total number of used vlans = 20

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 10-6.

Configuring the Internal VLAN Allocation Policy

For more information about VLAN allocation, see the "VLAN Ranges" section on page 14-2.

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 need not 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.
- If you enable a vlan translation within an interface, the inteface is reset.
- 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-SUP720-3BXL WS-SUP720-3B WS-SUP720	2	1	1–2	32	802.1Q
WS-SUP32-10GE	3	2	1, 2–3	16	ISL 802.1Q
WS-SUP32-GE	9	1	1–9	16	ISL 802.1Q
WS-X6704-10GE	4	4	1 port in each group	128	ISL 802.1Q
WS-X6708-10GE	8	8	1 port in each group	16	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
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 10-7.

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.
	`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 Cisco 7600 series router.
- 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 Cisco 7600 series router. 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
```

CHAPTER 15

Configuring Private VLANs

This chapter describes how to configure private VLANs on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding How Private VLANs Work

These sections describe how private VLANs work:

Private VLAN Domains

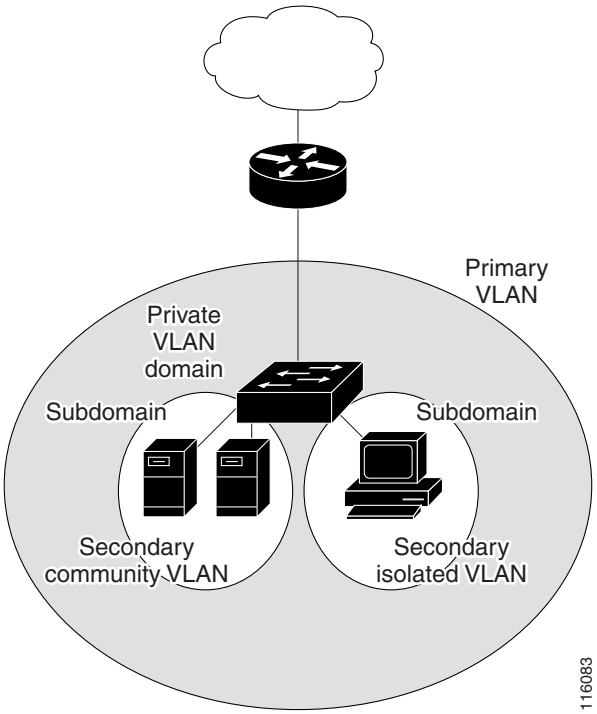
The private VLAN feature addresses two problems that service providers encounter when using VLANs:

- The router 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 15-1).

Figure 15-1 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 router 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 router 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:

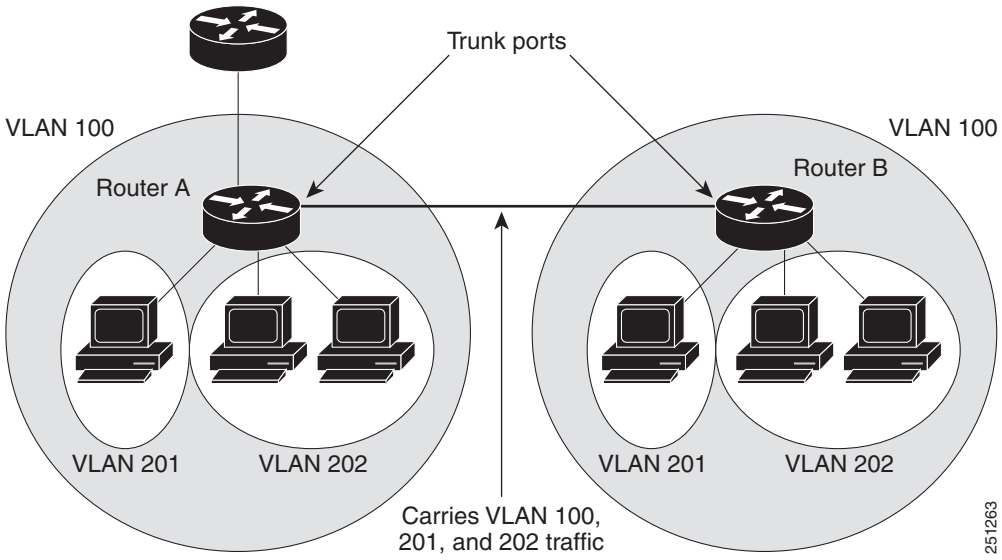
- Assigning a block of addresses to a customer VLAN can result in unused IP addresses.
- If the number of devices in the VLAN increases, the number of assigned addresses might not be large enough to accommodate them.

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 Routers

As with regular VLANs, private VLANs can span multiple routers. A trunk port carries the primary VLAN and secondary VLANs to a neighboring router. The trunk port deals with the private VLAN as any other VLAN. A feature of private VLANs across multiple routers is that traffic from an isolated port in router A does not reach an isolated port on Router B. (See Figure 15-2.)

Figure 15-2 Private VLANs Across Routers



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

Because VTP does not support private VLANs, you must manually configure private VLANs on all routers in the Layer 2 network. If you do not configure the primary and secondary VLAN association in some routers in the network, the Layer 2 databases in these routers are not merged. This situation can result in unnecessary flooding of private VLAN traffic on those routers.

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 15-6
- Private VLANs and SVIs, page 15-6

See also the "Private VLAN Configuration Guidelines and Restrictions" section on page 15-6.

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 router 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 15-7
- Private VLAN Port Configuration, page 15-9
- Limitations with Other Features, page 15-9

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 cannot change the VTP mode to client or server. For information about VTP, see Chapter 13, "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 Chapter 14, "Configuring VLANs."
- 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 router 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 router, 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 34, "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 44, "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 39-28.

- 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 39-28. 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 34-7.) 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 32, "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 20, "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 routers with private VLANs.
- A port is only affected by the private VLAN feature if it is currently in private VLAN mode and its private VLAN configuration indicates that it is a primary, isolated, or community port. If a port is in any other mode, such as Dynamic Trunking Protocol (DTP), it does not function as a private port.
- Do not configure private VLAN ports on interfaces configured for these other features:
  - Port Aggregation Protocol (PAgP)
  - Link Aggregation Control Protocol (LACP)
  - Voice VLAN
- You can configure IEEE 802.1x port-based authentication on a private VLAN port, but do not configure 802.1x with port security, voice VLAN, or per-user ACL on private VLAN ports.
- IEEE 802.1q mapping works normally. Traffic is remapped to or from dot1Q ports as configured, as if received from the ISL VLANs.
- Do not configure a remote SPAN (RSPAN) VLAN as a private VLAN primary or secondary VLAN. For more information about SPAN, see Chapter 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.
- 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:
  - In all releases, 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.
  - The 12-port restriction does not apply to these Ethernet switching modules: WS-X6548-RJ-45, WS-X6548-RJ-21, WS-X6524-100FX-MM (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
  - 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
  – A port that has been configured with the **switchport mode dynamic auto** or **switchport mode dynamic desirable** command

  If one port within the group of 24 ports is one of these ports listed and has the above properties, any isolated or community VLAN configuration for other ports within the 24 ports is inactive. To reactivate the ports, remove the isolated or community VLAN port configuration and enter the **shutdown** and **no shutdown** commands.

Configuring Private VLANs

These sections contain configuration information:

Note If the VLAN is not defined already, the private VLAN configuration process defines it.

Configuring a VLAN as a Private VLAN

To configure a VLAN as a private VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# vlan vlan_ID`	Enters VLAN configuration submode.
Step 2	`Router(config-vlan)# private-vlan {community | isolated | primary}`	Configures a VLAN as a private VLAN.
	`Router(config-vlan)# no private-vlan {community | isolated | primary}`	Clears the private VLAN configuration. **Note** These commands do not take effect until you exit VLAN configuration submode.
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.

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 15-1 shows the privileged EXEC commands for monitoring private VLAN activity.

Table 15-1 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 router.
show interface switchport	Displays private VLAN configuration on interfaces.
show interface private-vlan mapping	Displays information about the private VLAN mapping for VLAN SVIs.

This is an example of the output from the **show vlan private-vlan** command:

```
Switch(config)# show vlan private-vlan
Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
10      501       isolated          Fa2/0/1, Gi3/0/1, Gi3/0/2
10      502       community         Fa2/0/11, Gi3/0/1, Gi3/0/4
10      503       non-operational
```

CHAPTER 16

Configuring Cisco IP Phone Support

This chapter describes how to configure support for Cisco IP phones on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding Cisco IP Phone Support

These sections describe Cisco IP phone support:

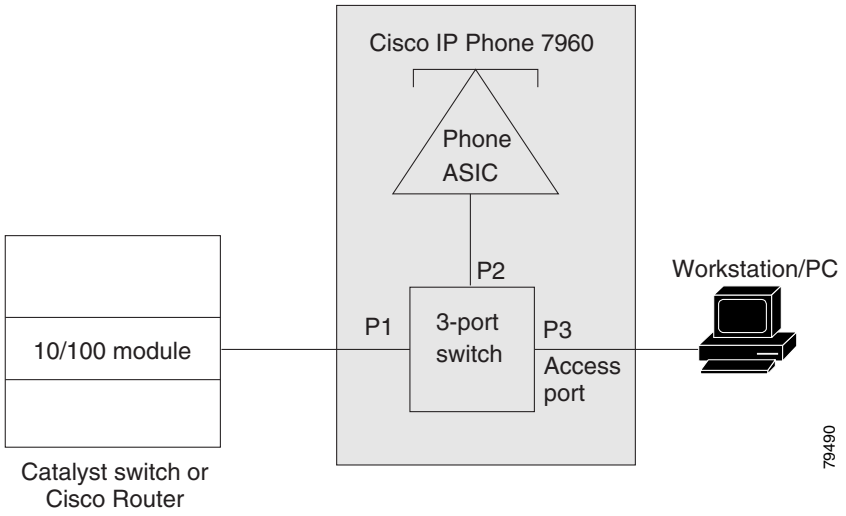
Cisco IP Phone Connections

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 router.
- 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 16-1 shows a Cisco IP phone connected between a router and a PC.

Figure 16-1 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 44, "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 router to send Cisco Discovery Protocol (CDP) packets that instruct an attached Cisco IP phone to transmit voice traffic to the router 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 16-1), you can configure Layer 2 access ports on the router to send CDP packets that instruct an attached Cisco IP phone to configure the access port on the Cisco IP phone to either of these two modes:

- Trusted mode—All traffic received through the access port on the Cisco IP phone passes through the Cisco IP phone unchanged.
- Untrusted mode—All traffic in 802.1Q or 802.1p frames received through the access port on the Cisco IP phone is marked with a configured Layer 2 CoS value. The default Layer 2 CoS value is 0. Untrusted mode is the default.

Cisco IP Phone Power Configurations

These sections describe Cisco IP phone power configurations:

- Locally Powered Cisco IP Phones, page 16-3
- Inline-Powered Cisco IP Phones, page 16-3

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

Inline power is from switching modules that support an inline power daughtercard. Inline power is sent over the twisted-pair Ethernet cable to the Cisco IP phone.

Note For information about switching modules that support inline power, refer to the *Release Notes for Cisco IOS Release Release Notes for Cisco IOS Release 12.2SX on the Supervisor Engine 720, Supervisor Engine 32, and Supervisor Engine 2* publication.

When a switching module port detects an unpowered Cisco IP phone, the switching module reports to the supervisor engine that an unpowered Cisco IP phone is present and on which module and port. If the port is configured in **auto** mode, the supervisor engine determines if there is enough system power available to power up the Cisco IP phone. If there is sufficient power available, the supervisor engine removes the default-allocated power required by a Cisco IP phone 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. 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.

Caution When a Cisco 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 Cisco 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.

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 Cisco 7600 series router 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 3.
- You cannot configure static secure MAC addresses in the voice VLAN.
- Ports configured with a voice VLAN can be secure ports (see Chapter 45, "Configuring Port Security").
- In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5 for voice traffic and 3 for voice control traffic).

Configuring Cisco IP Phone Support

These sections describe how to configure Cisco IP phone support:

Configuring Voice Traffic Support

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

	Command	Purpose
Step 1	`Router(config)# interface fastethernet slot/port`	Selects the port to configure.
Step 2	`Router(config-if)# switchport 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.

	Command	Purpose
Step 3	`Router(config)# end`	Exits configuration mode.
Step 4	`Router# show interfaces fastethernet slot/port switchport` `Router# show running-config interface fastethernet slot/port`	Verifies the configuration.

When configuring the way in which 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 router 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 router 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 router 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 router 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 44, "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 10-14 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 fastethernet slot/port`	Selects the port to configure.
Step 2	`Router(config-if)# mls qos trust extend [cos cos_value]`	Configures the way in which 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 interfaces fastethernet slot/port switchport` `Router# show running-config interface fastethernet slot/port`	Verifies the configuration.

When configuring the way in which the Cisco IP phone transmits 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 fastethernet slot/port`	Selects the port to configure.
Step 2	`Router(config-if)# power inline {auto | never}`	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 [fastethernet slot/port]`	Verifies the configuration.

When configuring inline power support, note the following information:

- To configure auto-detection of a Cisco IP phone, enter the **auto** keyword.
- To disable auto-detection of a Cisco IP phone, enter the **never** keyword.

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 17

Configuring IEEE 802.1Q Tunneling

This chapter describes how to configure IEEE 802.1Q tunneling on the Cisco 7600 series routers.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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 contains 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 routers.

The customer routers are trunk connected, but with 802.1Q tunneling, the service provider routers 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 router 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 17-1 on page 17-2 and Figure 17-2 on page 17-3.

Figure 17-1 IEEE 802.1Q Tunnel Ports in a Service-Provider Network

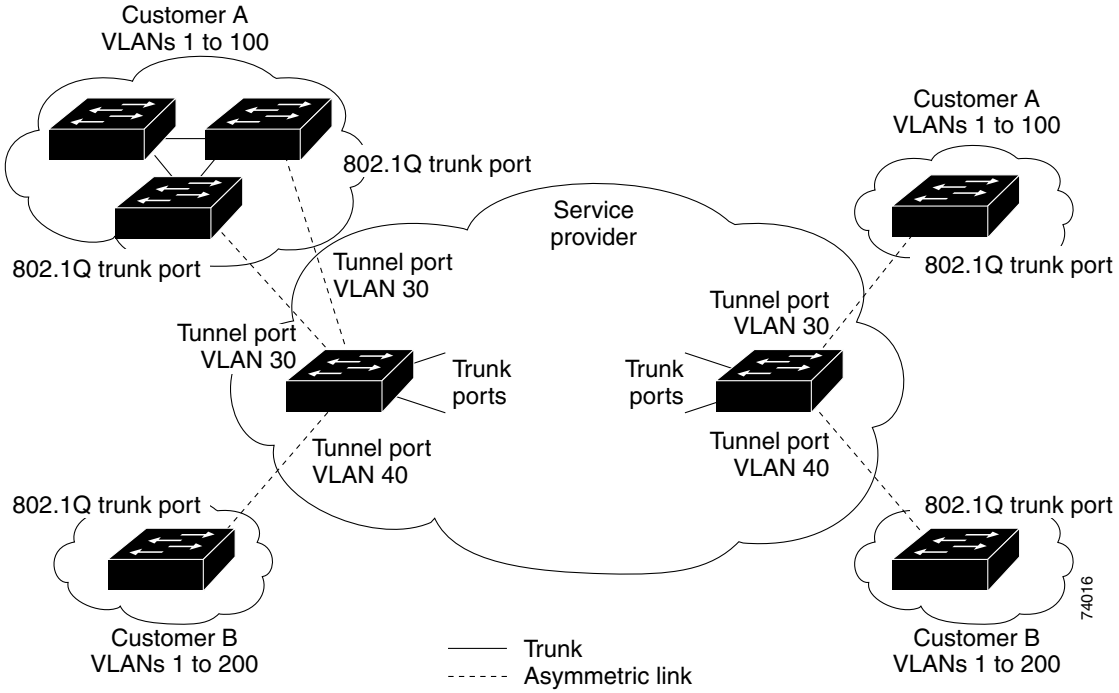
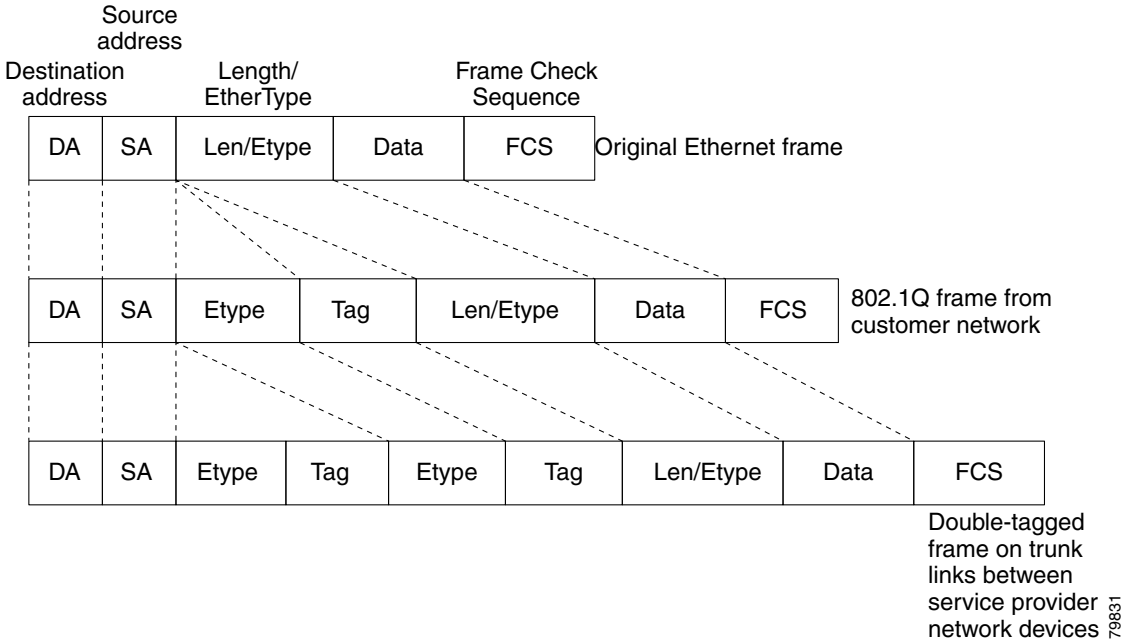


Figure 17-2 Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames



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.
- Ensure that the native VLAN of the 802.1Q trunk port in an asymmetrical link carries no traffic. Because traffic in the native VLAN is untagged, it cannot be tunneled correctly. Alternatively, you can enter the global **vlan dot1q tag native** command to tag native VLAN egress traffic and drop untagged native VLAN ingress traffic.
- Configure jumbo frame support on tunnel ports:
  - See the "Configuring Jumbo Frame Support" section on page 8-8.
  - 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 router, 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 router can provide only MAC-layer filtering for tunnel traffic (VLAN IDs and source and destination MAC addresses).
  - The router 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)
- PortFast 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 18, "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 routers, 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 PortFast 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 routers, 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 routers, *either* enable or disable the global **vlan dot1q tag native** option.

Note If this option is enabled on one router and disabled on another router, all traffic is dropped; all customer routers must have this option configured the same on each router.

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 routers, 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:

Caution Ensure that only the appropriate tunnel ports are in any VLAN used for tunneling and that one VLAN is used for each tunnel. Incorrect assignment of tunnel ports to VLANs can forward traffic inappropriately.

Configuring 802.1Q Tunnel Ports

To configure 802.1Q tunneling on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Selects the LAN port to configure.
Step 2	`Router(config-if)# switchport`	Configures the LAN port for Layer 2 switching: • You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional **switchport** commands with keywords. • Required only if you have not entered the **switchport** command already for the interface.
Step 3	`Router(config-if)# switchport mode dot1q-tunnel`	Configures the Layer 2 port as a tunnel port.
	`Router(config-if)# no switchport mode dot1q-tunnel`	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 Router to Tag Native VLAN Traffic

The **vlan dot1q tag native** command is a global command that configures the router 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 router to tag traffic in the native VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# vlan dot1q tag native`	Configures the router 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 router 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 18

Configuring Layer 2 Protocol Tunneling

This chapter describes how to configure Layer 2 protocol tunneling on the Cisco 7600 series routers.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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:

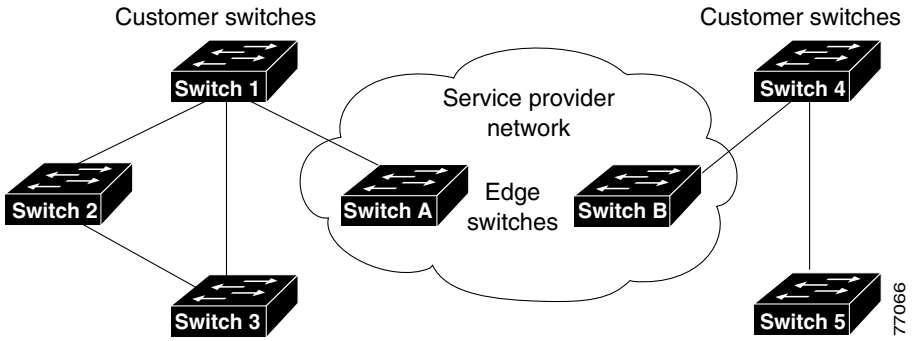
Understanding How Layer 2 Protocol Tunneling Works

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 router—The router connected to the customer router and placed on the boundary of the service provider network (see Figure 18-1).
- Layer 2 protocol tunnel port—A port on the edge router 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 router 1 (see Figure 18-1) 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 18-1 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 router 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 router 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 router.
- Configure jumbo frame support on Layer 2 protocol tunneling ports:
  - See the "Configuring Jumbo Frame Support" section on page 8-8.
  - 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 *Cisco 7600 Series Router Cisco IOS Command Reference* 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 19

Configuring STP and MST

This chapter describes how to configure the Spanning Tree Protocol (STP) and Multiple Spanning Tree (MST) protocol on Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter contains these sections:

Note For information on configuring the PortFast, UplinkFast, and BackboneFast STP enhancements, see Chapter 20, "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.

Cisco 7600 series routers 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 Cisco 7600 series routers, the extended system ID is always enabled.

The bridge priority is a 4-bit value when the extended system ID is enabled (see Table 19-1 on page 19-3 and the "Configuring the Bridge Priority of a VLAN" section on page 19-33).

Extended System ID

A 12-bit extended system ID field is part of the bridge ID (see Table 19-1 on page 19-3). Cisco 7600 series routers have 64 MAC addresses and always use the 12-bit extended system ID.

Table 19-1 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

Cisco 7600 series routers 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 router 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 router 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 19-2 describes the STP protocol timers that affect STP performance.

Table 19-2 STP Protocol Timers

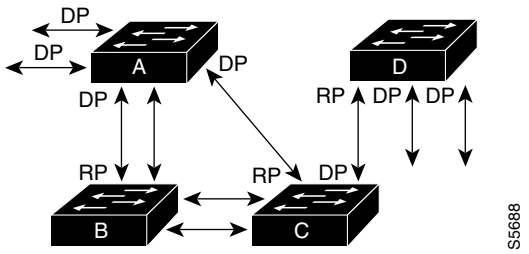
Variable	Description
Hello timer	Determines how often the network device broadcasts hello messages to other network devices.
Forward delay timer	Determines how long each of the listening and learning states last before the port begins forwarding.
Maximum age timer	Determines the amount of time protocol information received on an port is stored by the network device.

Creating the Spanning Tree Topology

In Figure 19-1, 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 19-1 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

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 Cisco 7600 series router using STP exists in one of the following five states:

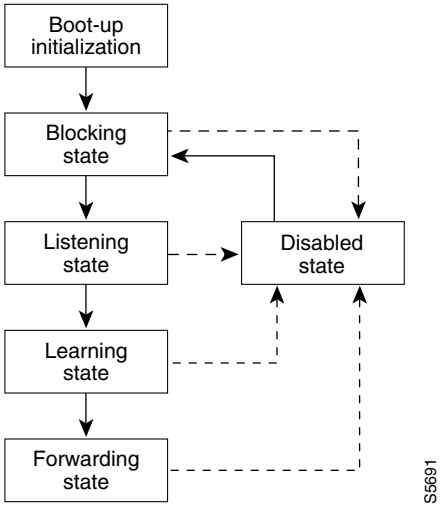
- Blocking—The Layer 2 LAN port does not participate in frame forwarding.
- Listening—First transitional state after the blocking state when STP determines that the Layer 2 LAN port should participate in frame forwarding.
- Learning—The Layer 2 LAN port prepares to participate in frame forwarding.
- 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 19-2 illustrates how a Layer 2 LAN port moves through the five states.

Figure 19-2 STP Layer 2 LAN Interface States



When you enable STP, every port in the Cisco 7600 series router, 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.

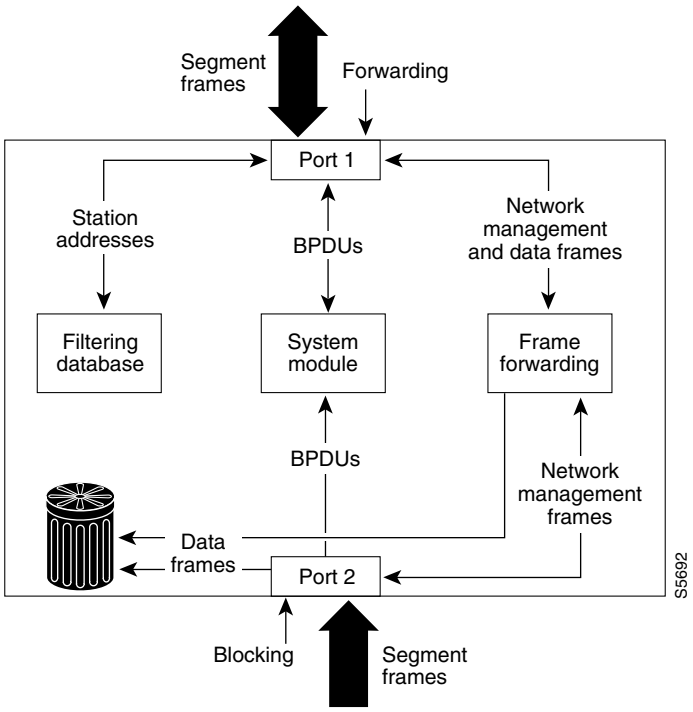
When the STP algorithm places a Layer 2 LAN port in the forwarding state, the following process occurs:

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 19-3. 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 19-3 Interface 2 in Blocking State



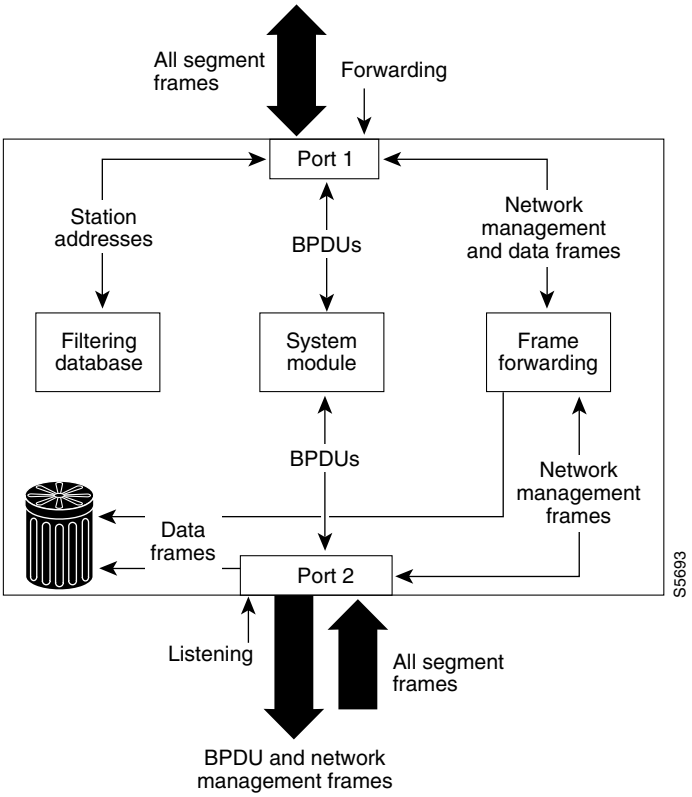
A Layer 2 LAN port in the blocking state performs as follows:

- 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 19-4 shows a Layer 2 LAN port in the listening state.

Figure 19-4 Interface 2 in Listening State



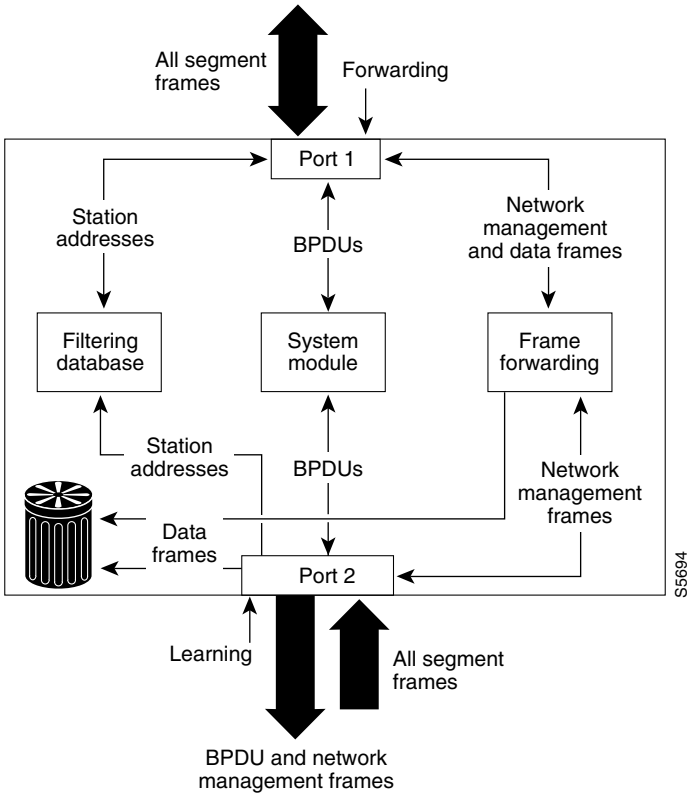
A Layer 2 LAN port in the listening state performs as follows:

- 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 19-5 shows a Layer 2 LAN port in the learning state.

Figure 19-5 Interface 2 in Learning State



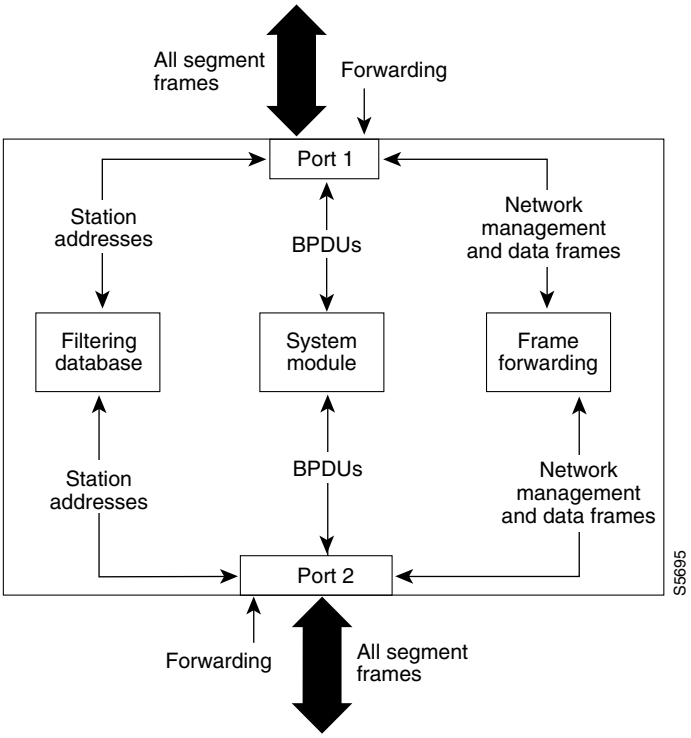
A Layer 2 LAN port in the learning state performs as follows:

- 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 19-6. The Layer 2 LAN port enters the forwarding state from the learning state.

Figure 19-6 Interface 2 in Forwarding State



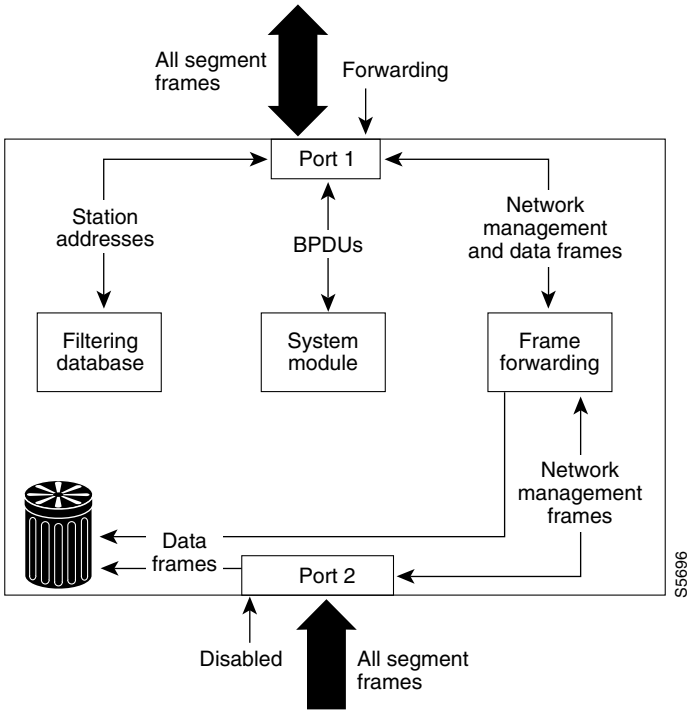
A Layer 2 LAN port in the forwarding state performs as follows:

- 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 19-7. A Layer 2 LAN port in the disabled state is virtually nonoperational.

Figure 19-7 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 10, "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 router 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 19-4. The RSTP then assigns one of these port roles to individual ports:

- Root port—Provides the best path (lowest cost) when the router forwards packets to the root bridge.
- Designated port—Connects to the designated router, which incurs the lowest path cost when forwarding packets from that LAN to the root bridge. The port through which the designated router 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 router 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 19-3 provides a comparison of 802.1D and RSTP port states.

Table 19-3 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
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 router, a router 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 router 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 19-8, router A is connected to router B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of router A is a smaller numerical value than the priority of router B. Router A sends a proposal message (a configuration BPDU with the proposal flag set) to router B, proposing itself as the designated router.

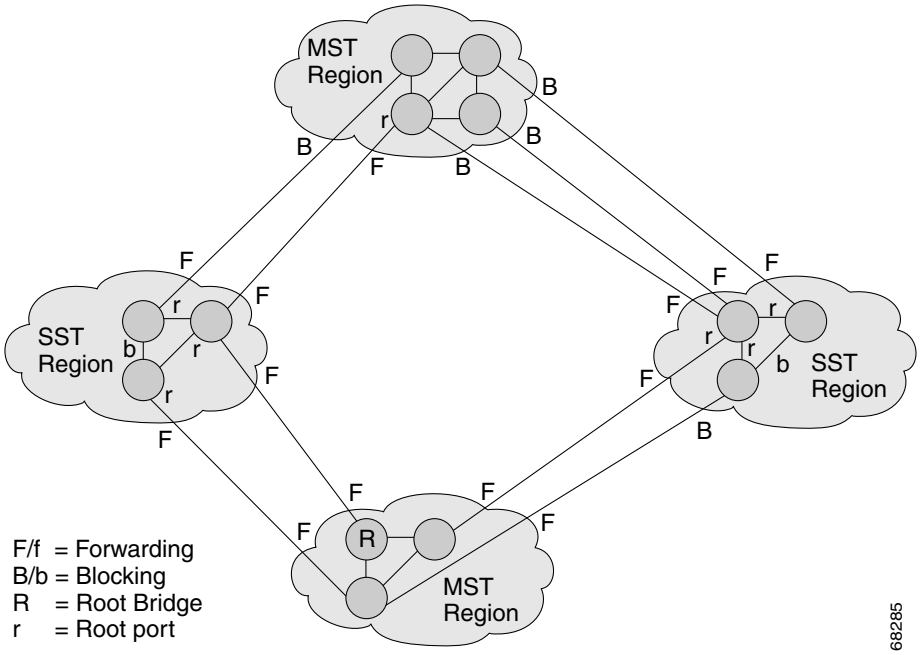
  After receiving the proposal message, router B selects as its new root port the port from which the proposal message was received, forces all nonedge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

  After receiving router B's agreement message, router A also immediately transitions its designated port to the forwarding state. No loops in the network are formed because router B blocked all of its nonedge ports and because there is a point-to-point link between routers A and B.

  When router C is connected to router B, a similar set of handshaking messages are exchanged. Router C selects the port connected to router B as its root port, and both ends immediately transition to the forwarding state. With each iteration of this handshaking process, one more router joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree.

The router 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 19-8 Proposal and Agreement Handshaking for Rapid Convergence



Synchronization of Port Roles

When the router 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.

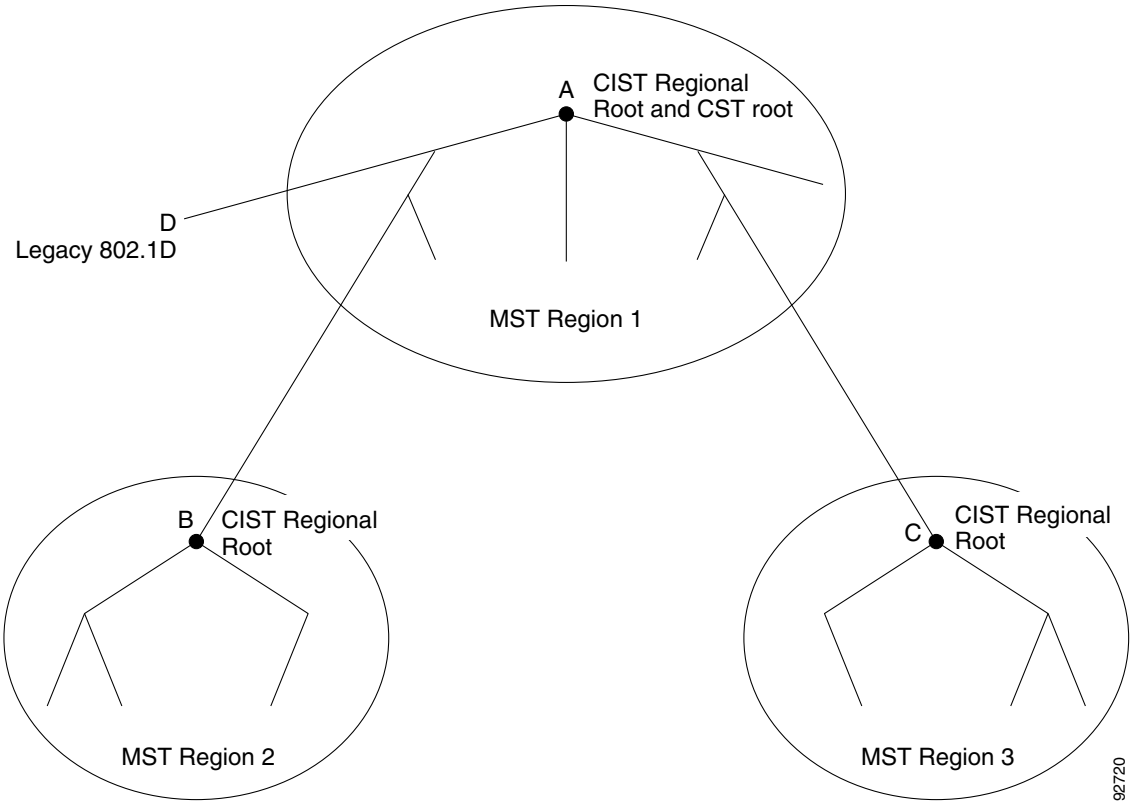
The router is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the router is synchronized if:

- 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 router sends an agreement message to the designated router corresponding to its root port. When the routers 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 19-9.

Figure 19-9 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 19-15
- Processing Superior BPDU Information, page 19-16
- Processing Inferior BPDU Information, page 19-16

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 19-4 describes the RSTP flag fields.

Table 19-4 RSTP BPDU Flags

Bit	Function
0	Topology change (TC)
1	Proposal

Table 19-4 RSTP BPDU Flags (continued)

Bit	Function
2–3:	Port role:
00	Unknown
01	Alternate port or backup port
10	Root port
11	Designated port
4	Learning
5	Forwarding
6	Agreement
7	Topology change acknowledgement (TCA)

The sending router sets the proposal flag in the RSTP BPDU to propose itself as the designated router on that LAN. The port role in the proposal message is always set to the designated port.

The sending router 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 routers, the RSTP router 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 router sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D BPDU, the router 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 router 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 router processes and generates TCN BPDUs.
- Acknowledgement—When an RSTP router receives a TCN message on a designated port from an 802.1D router, 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 router 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 routers. The RSTP BPDUs never have the TCA bit set.
- Propagation—When an RSTP router receives a TC message from another router 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 router starts the TC-while timer for all such ports and flushes the information learned on them.
- Protocol migration—For backward compatibility with 802.1D routers, 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 router processes all BPDUs received on that port and ignores the protocol type.

  If the router receives an 802.1D BPDU after the port migration-delay timer has expired, it assumes that it is connected to an 802.1D router and starts using only 802.1D BPDUs. However, if the RSTP router 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

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 other spanning tree features such as Port Fast, UplinkFast, root guard, and so forth, see Chapter 20, “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 routers to participate in MST instances, you must consistently configure the routers with the same MST configuration information. A collection of interconnected routers that have the same MST configuration comprises an MST region as shown in Figure 19-10 on page 19-21.

The MST configuration controls to which MST region each router 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 routers 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 19-20 and the "Spanning Tree Operations Between MST Regions" section on page 19-20.

Spanning Tree Operation Within an MST Region

The IST connects all the MST routers 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 19-10 on page 19-21. 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 routers at the boundary of the region is selected as the CIST regional root.

When an MST router 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 router also initializes all of its MST instances and claims to be the root for all of them. If the router 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 routers 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 routers in the MST region must agree on the same CIST regional root. Therefore, any two routers in the region only synchronize their port roles for an MST instance if they converge to a common CIST regional root.

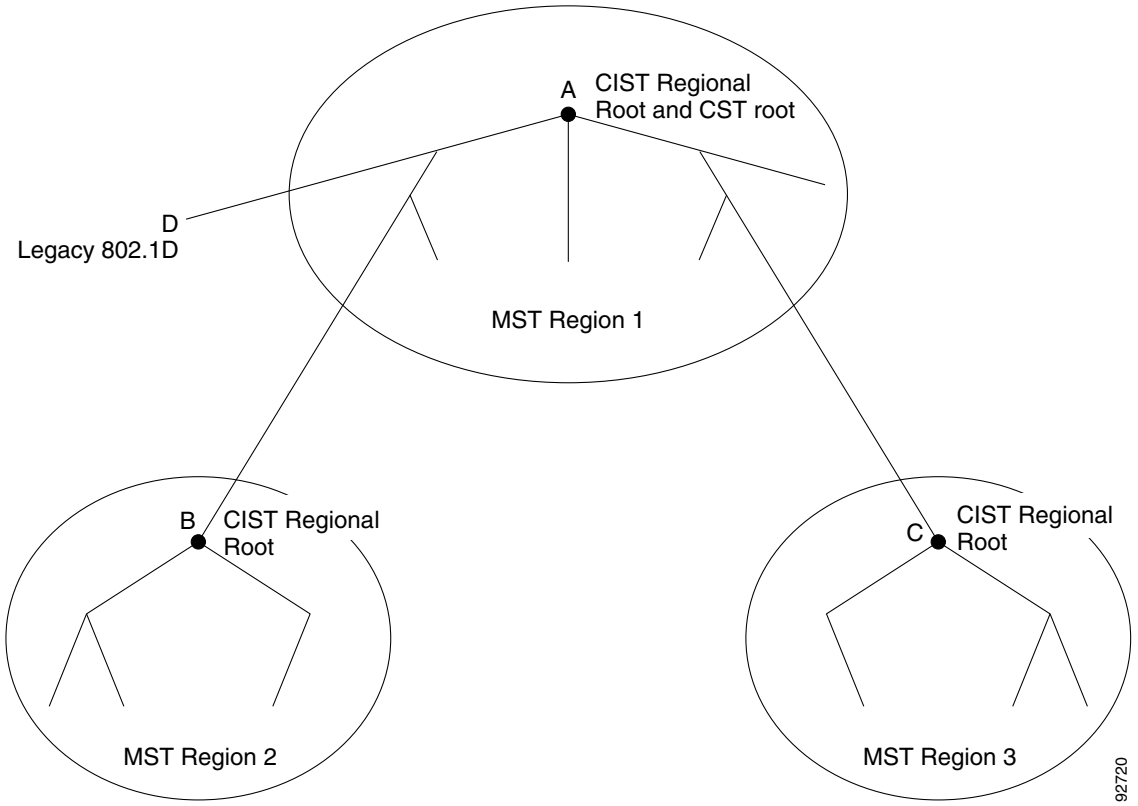
Spanning Tree Operations Between MST Regions

If there are multiple regions or 802.1D routers within the network, MST establishes and maintains the CST, which includes all MST regions and all 802.1D STP routers 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 routers 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 router to adjacent STP routers and MST regions.

Figure 19-10 shows a network with three MST regions and an 802.1D router (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 19-10 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 routers 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 routers use Version 3 BPDUs or 802.1D STP BPDUs to communicate with 802.1D routers. MST routers use MST BPDUs to communicate with MST routers.

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 router to the CIST. The CIST external root path cost is the root path cost calculated between these virtual routers and routers 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 router 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 19-5 compares the IEEE standard and the Cisco prestandard terminology.

Table 19-5 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 router 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 router 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 router of which is either a single spanning tree router or a router 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 routers 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 router 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 router has the sender switch ID. The whole region performs like a single virtual router by sending a consistent sender switch ID to neighboring routers. In this example, router 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 19-23
- Spanning Tree Interoperation Between Legacy and Standard-Compliant Routers, page 19-24
- Detecting Unidirectional Link Failure, page 19-24

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:

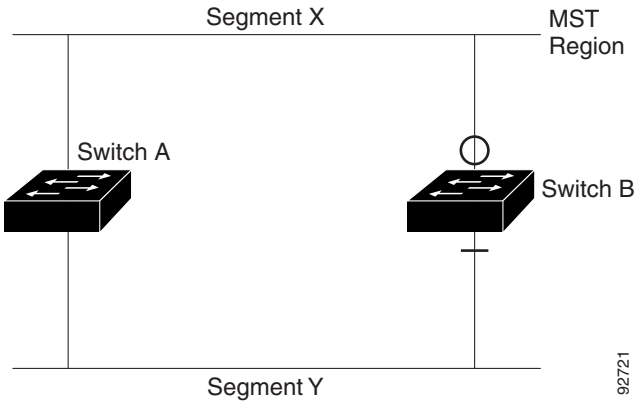
- The boundary port is the root port of the CIST regional root—When the CIST instance port is proposed and is synchronized, it can send back an agreement and move to the forwarding state only after all the corresponding MSTI ports are synchronized (and thus forwarding). The MSTI ports now have a special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

Because automatic detection of prestandard routers can fail, you can use an interface configuration command to identify prestandard ports. A region cannot be formed between a standard and a prestandard router, 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 router receives a prestandard BPDU on a port that has not been configured for prestandard BPDU transmission.

Figure 19-11 illustrates a standard-compliant router connected to a prestandard router. Assume that A is the standard-compliant router and B is a prestandard router, 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 router 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 19-11 Standard-Compliant and Prestandard Router Interoperation



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

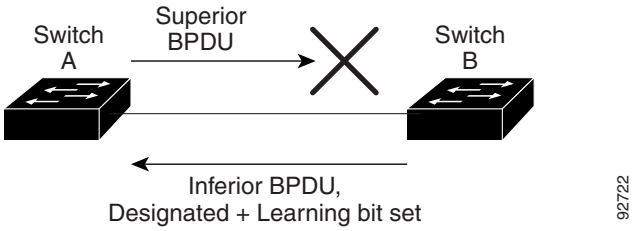
Detecting Unidirectional Link Failure

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 19-12 illustrates a unidirectional link failure that typically creates a bridging loop. Router A is the root bridge, and its BPDUs are lost on the link leading to router B. RSTP and MST BPDUs include the role and state of the sending port. With this information, router A can detect that router B does not react to the superior BPDUs it sends and that router B is the designated, not root bridge. As a result, router A blocks (or keeps blocking) its port, thus preventing the bridging loop.

Figure 19-12 Detecting Unidirectional Link Failure



Interoperability with IEEE 802.1D-1998 STP

A router running MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D routers. If this router 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 router 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 router does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D router has been removed from the link unless the 802.1D router is the designated router. A router might also continue to assign a boundary role to a port when the router to which this router is connected has joined the region. To restart the protocol migration process (force the renegotiation with neighboring routers), use the **clear spanning-tree detected-protocols** privileged EXEC command.

If all the 802.1D routers on the link are RSTP routers, they can process MST BPDUs as if they are RSTP BPDUs. Therefore, MST routers 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 router of which is either a single spanning tree router or a router 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 19-6 shows the default STP configuration.

Table 19-6 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 Cisco 7600 series router 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 19-6 on page 19-26).
	`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 Cisco 7600 series routers. This example shows how to verify the configuration:

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

Configuring the Root Bridge

Cisco 7600 series routers 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 router checks the bridge priority of the current root bridges for each VLAN. With the extended system ID enabled, the router sets the bridge priority for the specified VLANs to 24576 if this value will cause the router 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 router 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 19-1 on page 19-3.)

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 router sets the bridge priority to 24576, which causes the router to become the root bridge for VLAN 20.

Caution The root bridge for each instance of STP should be a backbone or distribution router. Do not configure an access router 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 Cisco 7600 series router 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 Cisco 7600 series router as the root bridge.

To configure a Cisco 7600 series router 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 Cisco 7600 series router as the root bridge. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 19-6 on page 19-26).
	`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 Cisco 7600 series router 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 Cisco 7600 series router as the secondary root, the STP bridge priority is modified from the default value (32768) so that the router 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 router 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 Cisco 7600 series router 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 Cisco 7600 series router as the secondary root bridge. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 14-1 on page 14-2).
	`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 Cisco 7600 series router 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 14-1 on page 14-2).
	`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 14-1 on page 14-2).
	`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 14-1 on page 14-2).
	`Router(config)# no spanning-tree vlan vlan_ID priority`	Reverts to the default bridge priority value.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show spanning-tree vlan vlan_ID bridge [detail]`	Verifies the configuration.

This example shows how to configure the bridge priority of VLAN 200 to 33792 when the extended system ID is disabled:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 priority 32768
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge
                                     Hello Max  Fwd
Vlan                 Bridge ID       Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          32768 0050.3e8d.64c8    2   20    15  ieee
Router#
```

Configuring the Hello Time

Note Be careful when using this command. For most situations, we recommend that you use the **spanning-tree vlan** *vlan_ID* **root primary** and **spanning-tree vlan** *vlan_ID* **root secondary** commands to modify the hello time.

To configure the STP hello time of a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree vlan vlan_ID hello-time hello_time`	Configures the hello time of a VLAN. The *hello_time* value can be from 1 to 10 seconds. The *vlan_ID* value can be 1 through 4094, except reserved VLANs (see Table 14-1 on page 14-2).
	`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 14-1 on page 14-2).
	`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 14-1 on page 14-2).
	`Router(config)# no spanning-tree vlan vlan_ID max-age`	Reverts to the default maximum aging time.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show spanning-tree vlan vlan_ID bridge [detail]`	Verifies the configuration.

This example shows how to configure the maximum aging time for VLAN 200 to 36 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 max-age 36
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge
                                     Hello Max  Fwd
Vlan                 Bridge ID       Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          49152 0050.3e8d.64c8    2   36    15  ieee
Router#
```

Enabling Rapid-PVST

Rapid-PVST uses the existing PVST+ framework for configuration and interaction with other features. It also supports some of the PVST+ extensions.

To enable Rapid-PVST mode on the router, enter the **spanning-tree mode rapid-pvst** command in privileged mode. To configure the router in Rapid-PVST mode, see the "Configuring STP" section on page 19-25.

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 router 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 router running both MSTP and RSTP supports a built-in protocol migration process that enables the router to interoperate with legacy 802.1D switches. If this router 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 router 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 router does not automatically revert to the MSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether the legacy router has been removed from the link unless the legacy router is the designated router. A router also might continue to assign a boundary role to a port when the router to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches) on the entire router, 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 19-7 shows the default MST configuration.

Table 19-7 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
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 router 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.

Specifying the MST Region Configuration and Enabling MST

For two or more routers to be in the same MST region, they must have the same VLAN-to-instance mapping, the same configuration revision number, and the same MST name.

A region can have one member or multiple members with the same MST configuration; each member must be capable of processing RSTP BPDUs. There is no limit to the number of MST regions in a network, but each region can only support up to 65 spanning tree instances. You can assign a VLAN to only one spanning tree instance at a time.

To specify the MST region configuration and enable MST, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# spanning-tree mst configuration`	Enters MST configuration mode.
Step 3	`Router(config-mst)# instance` *instance_id* **vlan** *vlan_range*	Maps VLANs to an MST instance. • For *instance_id*, the range is 0 to 4094. • For **vlan** *vlan_range*, the range is 1 to 4094. When you map VLANs to an MST instance, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped. To specify a VLAN range, use a hyphen; for example, **instance 1 vlan 1-63** maps VLANs 1 through 63 to MST instance 1. To specify a VLAN series, use a comma; for example, **instance 1 vlan 10, 20, 30** maps VLANs 10, 20, and 30 to MST instance 1.
Step 4	`Router(config-mst)# name` *instance_name*	Specifies the instance name. The *name* string has a maximum length of 32 characters and is case sensitive.
Step 5	`Router(config-mst)# revision` *version*	Specifies the configuration revision number. The range is 0 to 65535.
Step 6	`Router(config-mst)# show pending`	Verifies your configuration by displaying the pending configuration.
Step 7	`Router(config)# exit`	Applies all changes, and return to global configuration mode.
Step 8	`Router(config)# spanning-tree mode mst`	Enables MST and RSTP. **Caution** Changing the spanning tree mode can disrupt traffic because all spanning tree instances are stopped for the previous mode and restarted in the new mode. You cannot run both MST and PVST+ or both MST and rapid PVST+ at the same time.
Step 9	`Router(config)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 10	`Router# show running-config`	Verifies your entries.
Step 11	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return to defaults, do the following:

- To return to the default MST region configuration, use the **no spanning-tree mst configuration** global configuration command.
- To return to the default VLAN-to-instance map, use the **no instance** *instance_id* [**vlan** *vlan_range*] MST configuration command.
- To return to the default name, use the **no name** MST configuration command.
- To return to the default revision number, use the **no revision** MST configuration command.
- To reenable PVST+, use the **no spanning-tree mode** or the **spanning-tree mode pvst** global configuration command.

This example shows how to enter MST configuration mode, map VLANs 10 to 20 to MST instance 1, name the region *region1*, set the configuration revision to 1, display the pending configuration, apply the changes, and return to global configuration mode:

```
Router(config)# spanning-tree mst configuration
Router(config-mst)# instance 1 vlan 10-20
Router(config-mst)# name region1
Router(config-mst)# revision 1
Router(config-mst)# show pending
Pending MST configuration
Name      [region1]
Revision  1
Instances configured 2
Instance  Vlans Mapped
--------  ---------------------
0         1-9,21-4094
1         10-20
-------------------------------

Router(config-mst)# exit
Router(config)#
```

Configuring the Root Bridge

The router maintains a spanning tree instance for the group of VLANs mapped to it. A switch ID, consisting of the switch priority and the router MAC address, is associated with each instance. For a group of VLANs, the router with the lowest switch ID becomes the root bridge.

To configure a router 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 router becomes the root bridge for the specified spanning tree instance. When you enter this command, the router checks the switch priorities of the root bridges. Because of extended system ID support, the router sets its own priority for the specified instance to 24576 if this value will cause this router 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 router 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 19-1 on page 19-3.)

If your network consists of routers that both do and do not support the extended system ID, it is unlikely that the router 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 routers running older software.

The root bridge for each spanning tree instance should be a backbone or distribution router. Do not configure an access router 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 router 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 router 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 router 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 router 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 router 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 router with the extended system ID support as the secondary root, the switch priority is modified from the default value (32768) to 28672. The router 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 routers 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 router 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 router 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 router 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 routers 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 19-40.
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 router to its default setting, use the **no spanning-tree mst** *instance_id* **root** global configuration command.

Configuring Port Priority

If a loop occurs, MST uses the port priority when selecting an interface to put into the forwarding state. You can assign higher priority values (lower numerical values) to interfaces that you want selected first and lower priority values (higher numerical values) that you want selected last. If all interfaces have the same priority value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

To configure the MST port priority of an interface, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface {{gigabitethernet 1/port} | {port-channel number}}`	(Optional) Specifies an interface to configure, and enters interface configuration mode.
Step 3	`Router(config-if)# spanning-tree mst instance_id port-priority priority`	Configures the port priority. • For *instance_id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is 0 to 4094. • For *priority*, the range is 0 to 240 in increments of 16. The default is 128. The lower the number, the higher the priority. The priority values are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240. All other values are rejected.
Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.

Step 5	`Router# show spanning-tree mst interface interface_id`
	or
	`Router# show spanning-tree mst instance_id`
		Verifies your entries.

Step 6	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Note The **show spanning-tree mst interface** *interface_id* privileged EXEC command displays information only if the port is in a link-up operative state. Otherwise, you can use the **show running-config interface** privileged EXEC command to confirm the configuration.

To return the interface to its default setting, use the **no spanning-tree mst** *instance_id* **port-priority** interface configuration command.

Configuring Path Cost

The MST path cost default value is derived from the media speed of an interface. If a loop occurs, MST uses cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost values that you want selected last. If all interfaces have the same cost value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

To configure the MST cost of an interface, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface {{gigabitethernet 1/port} | {port-channel number}}`	(Optional) Specifies an interface to configure, and enters interface configuration mode.
Step 3	`Router(config-if)# spanning-tree mst instance_id cost cost`	Configures the cost. If a loop occurs, MST uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission. • For *instance_id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is 0 to 4094. • For *cost*, the range is 1 to 200000000; the default value is derived from the media speed of the interface.
Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.

Step 5	`Router# show spanning-tree mst interface interface_id`
	or
	`Router# show spanning-tree mst instance_id`
		Verifies your entries.

Step 6	`Router# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Note The **show spanning-tree mst interface** *interface_id* privileged EXEC command displays information only for ports that are in a link-up operative state. Otherwise, you can use the **show running-config** privileged EXEC command to confirm the configuration.

To return the interface to its default setting, use the **no spanning-tree mst** *instance_id* **cost** interface configuration command.

Configuring the Switch Priority

You can configure the switch priority so that it is more likely that a router 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 router 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 router 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 router 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 router 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 router 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 router 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 router 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 router 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 router 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 19-13.

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 router 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 router running MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D routers. If this router 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 router 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 router does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D router has been removed from the link unless the 802.1D router is the designated router. A router also might continue to assign a boundary role to a port when the router to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring routers) on the router, 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 19-8.

Table 19-8 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 20

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 Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Note For information on configuring the spanning tree protocol (STP), see Chapter 19, "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 router, 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 20-10), 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 20-1 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 20-1 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.

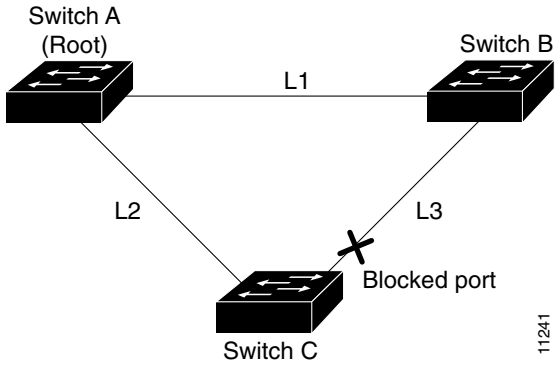
Understanding How UplinkFast Works

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 (for each 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.

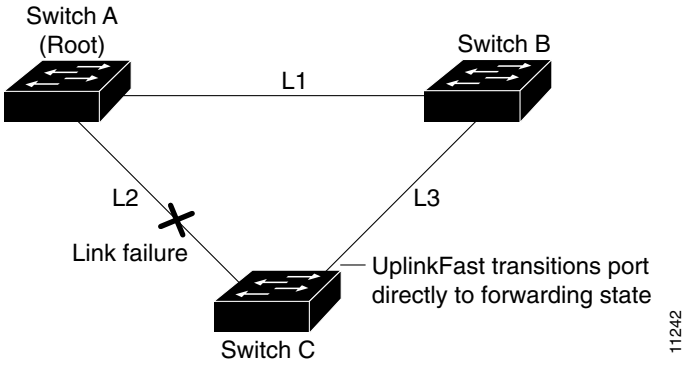
Figure 20-1 shows an example topology with no link failures. Switch A, the root bridge, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 20-1 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 20-2. This switchover takes approximately one to five seconds.

Figure 20-2 UplinkFast Example After Direct Link Failure



Understanding How BackboneFast Works

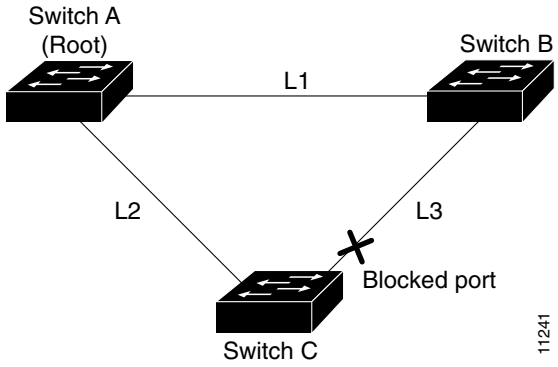
BackboneFast is initiated when a root port or blocked port on a network device receives inferior BPDUs from its designated bridge. An inferior BPDU identifies one network device as both the root bridge and the designated bridge. When a network device receives an inferior BPDU, it indicates that a link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

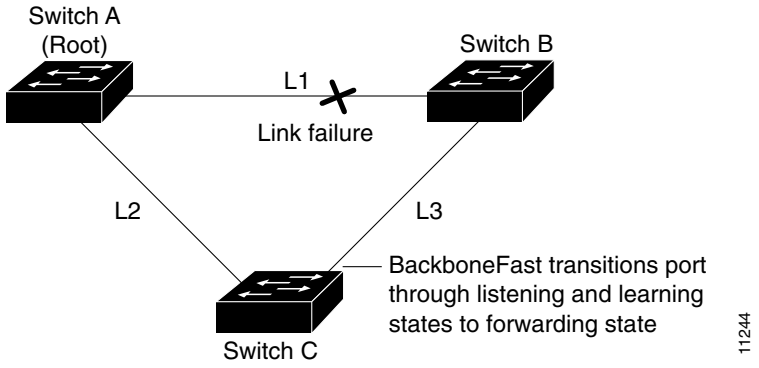
Figure 20-3 shows an example topology with no link failures. Switch A, the root bridge, connects directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that connects directly to Switch B is in the blocking state.

Figure 20-3 BackboneFast Example Before Indirect Link Failure



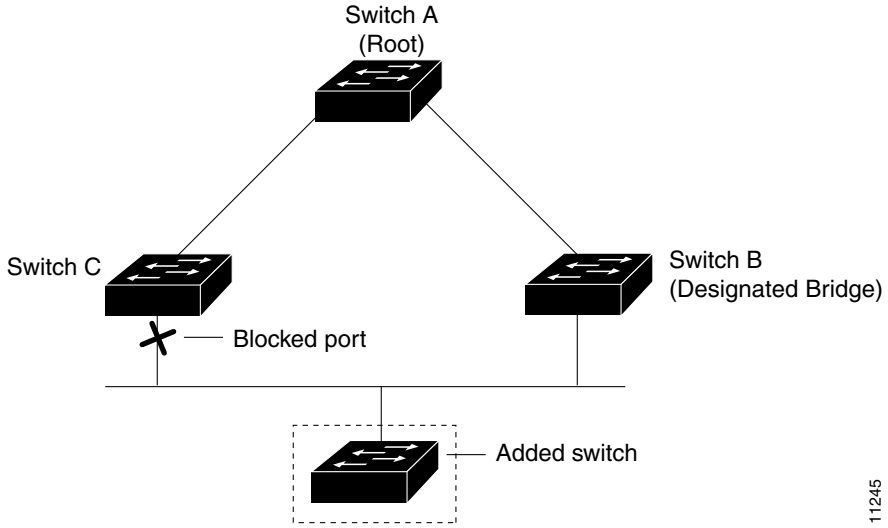
If link L1 fails, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root bridge over L1, it detects the failure and elects itself the root and begins sending BPDUs to Switch C indicating itself as the root. When 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 20-4 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 20-4 BackboneFast Example After Indirect Link Failure



If a new network device is introduced into a shared-medium topology as shown in Figure 20-5, 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 20-5 Adding a Network Device in a Shared-Medium Topology



Understanding How EtherChannel Guard Works

EtherChannel guard detects a misconfigured EtherChannel where interfaces on the Cisco 7600 series router 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 Cisco 7600 series router 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 20-6 shows loop guard in a triangle router configuration.

Figure 20-6 Triangle Switch Configuration with Loop Guard

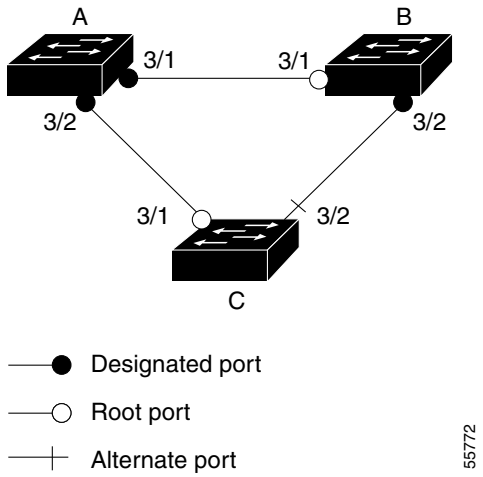


Figure 20-6 illustrates the following configuration:

- Switches A and B are distribution switches.
- Switch C is an access router.
- Loop guard is enabled on ports 3/1 and 3/2 on Switches A, B, and C.

Enabling loop guard on a root router has no effect but provides protection when a root router becomes a nonroot router.

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 router.
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 19, "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 Cisco 7600 series router, decreasing the probability that the router will become the root bridge. The *max_update_rate* value represents the number of multicast packets transmitted per second (the default is 150 packets per second). 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 Cisco 7600 series router. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree uplinkfast [max-update-rate max_update_rate]`	Enables UplinkFast.
	`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 with an update rate of 400 packets per second:

```
Router# configure terminal
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 router, perform this task:

	Command	Purpose
Step 1	`Router(config)# spanning-tree loopguard default`	Enables loop guard globally on the router.
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 21

Configuring Layer 3 Interfaces

This chapter contains information about how to configure Layer 3 interfaces on the Cisco 7600 series routers.

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

- The Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_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.
- The [**no**] **ip dhcp route** [**connected** | **static**] command is supported.
- To support VLAN interfaces, create and configure VLANs and assign VLAN membership to Layer 2 LAN ports. For more information, see Chapter 14, "Configuring VLANs" and Chapter 13, "Configuring VTP."
- Cisco 7600 series routers 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 MSFC.
- Cisco 7600 series routers do not support the IEEE bridging protocol for bridge groups. Configure bridge groups to use the VLAN-bridge or the DEC spanning-tree protocol.
- Do not configure an IP address on the physical interface if there is a subinterface configured with dot1q native encapsulation on the same physical interface.

Configuring Subinterfaces on Layer 3 Interfaces

When configuring Layer 3 subinterfaces, follow these guidelines and restrictions:

- The following 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 router, 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/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/index.htm

When configuring IPv4 routing and addresses, follow these guidelines and restrictions:

- For information about the **maximum paths** command, refer to the *Cisco 7600 Series Router Cisco IOS Command Reference* publication.
- The Policy Feature Card (PFC) and any Distributed Feature Cards (DFCs) provide hardware support for policy-based routing (PBR) for route-map sequences that use the **match ip address**, **set ip next-hop**, and **ip default next-hop** PBR keywords.

When configuring PBR, follow these guidelines and restrictions:

- The PFC provides hardware support for PBR configured on a tunnel interface.
- The PFC does not provide hardware support for PBR configured with the **set ip next-hop** keywords if the next hop is a tunnel interface.
- If the MSFC address falls within the range of a PBR ACL, traffic addressed to the MSFC is policy routed in hardware instead of being forwarded to the MSFC. To prevent policy routing of traffic addressed to the MSFC, configure PBR ACLs to deny traffic addressed to the MSFC.
- Any options in Cisco IOS ACLs that provide filtering in a PBR route-map that would cause flows to be sent to the MSFC 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/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt1/qcfpbr.htm

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 MSFC supports Internetwork Packet Exchange (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/univercd/cc/td/doc/product/software/ios122/122cgcr/fatipx_c/index.htm

To configure routing for IPX and to 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 22

IP Subscriber Awareness over Ethernet

This chapter provides information about how various Cisco 7600 features are scaled to support the IP Subscriber Awareness over Ethernet feature (sometimes referred to as *IP subscriber aggregation*), which was introduced for the Cisco 7600 series router in Cisco IOS Release 12.2SRB. From Cisco IOS Release 12.2(33)SRE onwards, the ISG functionality in distributed IP and PPPoE sessions on the Cisco 7600 series routers is supported on Ethernet Services Plus (ES+) access-facing line cards.

This chapter contains the following sections:

Overview

IP Subscriber Awareness over Ethernet is designed for an architecture in which the Cisco 7600 router is used as a DSLAM Gigabit Ethernet (GE) aggregator. In this scenario, the DSLAM is connected to the router through a physical port carrying data for multiple VLANs.

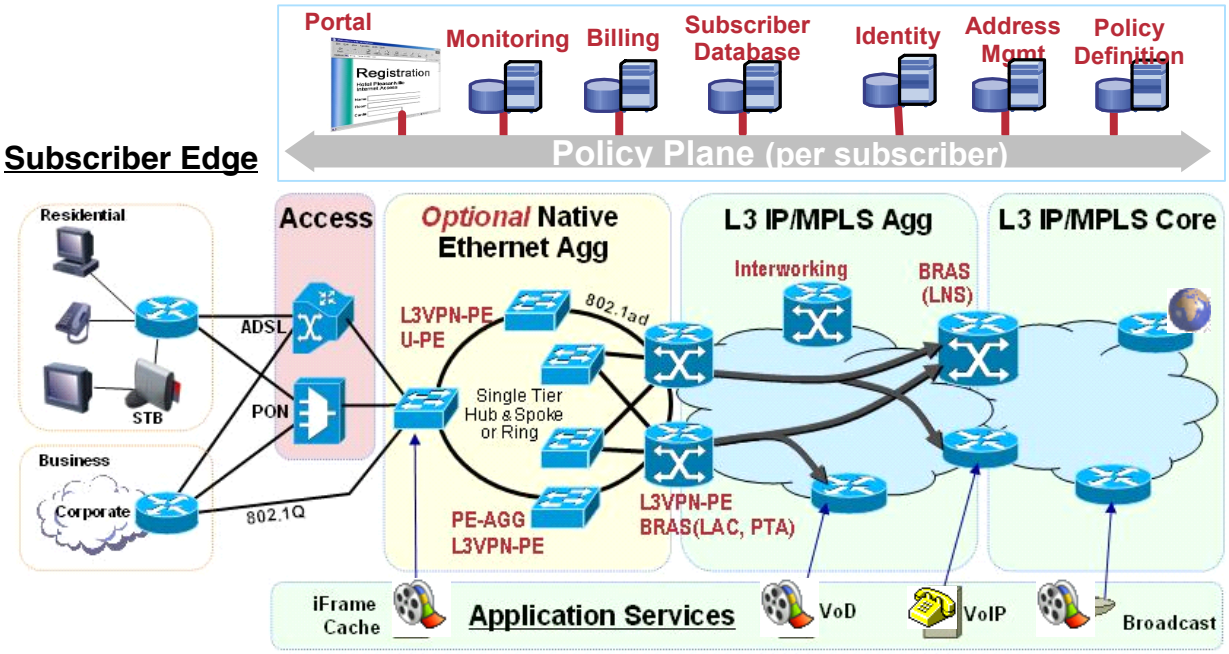
The IP Subscriber Awareness over Ethernet feature supports two models of carrying video, voice, and data services between the subscriber and the DSLAM:

Per-service VLAN model—One or more ATM VCs is used to carry each type of service between the subscriber and the VLAN.

- Per-subscriber VLAN model—A single ATM VC is used to carry all traffic (video, voice, and data) between the subscriber and the DSLAM.

Figure 22-1 shows an example of a wireline Ethernet architecture where the IP Subscriber Awareness over Ethernet might be used.

Figure 22-1 Wireline Ethernet Architecture



These following sections provide more details about the IP Subscriber Awareness over Ethernet feature:

- Benefits, page 22-2
- IP Subscriber Interfaces, page 22-3
- IP Subscriber Session, page 22-3
- IP Subscriber Session Features, page 22-4

Benefits

The IP Subscriber Awareness over Ethernet feature provides these benefits:

- IP session termination and IP session aggregation on the Cisco 7600 router.
- Support for up to 32000 IP subscribers on a router with ES+ line cards:
  - Supports a maximum of 4000 subscribers on a Gigabit Ethernet port
  - Supports a maximum of 8000 subscribers on a Ten Gigabit Ethernet port
- Interface scalability to support up to 32000 interfaces on the router having a Cisco 7600 SIP-400.
  - Support for up to 1000 subinterfaces on each physical port.
  - Support for up to 8000 subinterfaces on each Cisco 7600 SIP-400.
- DHCP and Radius accounting for IP subscribers. Support for 256 DHCP pools. A DHCP can handle up to 150 calls per second for IP subscriber sessions.

- QoS support for individual IP subscribers (up to 32000 subscribers), including: classification (IP prec and DSCP), policing, shaping, marking, priority queues, and weighted random early detection (WRED).
- Per-subscriber statistics and accounting information.
- Support for up to 96000 ARP entries.
- RPR, RPR+, stateful switchover (SSO), and non-stop forwarding (NSF) are provided for the IP subscribers.
- Control plane protection (CoPP) protects against denial of service (DOS) and other attacks.

IP Subscriber Interfaces

Starting Cisco IOS Release 12.2SRB, a new type of interface is used to represent IP subscribers:

- Access—A subinterface that represents an individual IP subscriber. The access subinterface can be configured for .1Q or Q-in-Q encapsulation.

  Apply traffic shaping and policing policies (including HQoS) to the access interface to define the amount of bandwidth for different types of subscriber traffic (such as, voice and data).

Note Configure the access interface as a subinterface of the physical interface that the IP subscriber is connected to.

Example

The following example shows an access subinterface on the interface :

```
interface GigabitEthernet 1/0/0.100 access
    ip vrf forwarding vrf0
    encapsulation dot1q 100
```

  - On an ES+ line card, this feature is supported on the access interfaces and non-access interfaces (limited to 500 subinterfaces).

IP Subscriber Session

An IP subscriber session exists while an IP subscriber is using its shared VLAN to access the network. To begin an IP subscriber session, the router must assign an IP address to the subscriber's access subinterface. You can either assign a static IP address to the subinterface, or you can allow DHCP to assign an address. Following are some notes about both methods of assigning an IP address:

- Static IP address—If you assign a static IP address to the access subinterface, the IP subscriber session is considered to always be Up. We recommend that you do not configure many IP subscribers with static IP addresses.
- DHCP-assigned IP address—You can allow DHCP to assign an IP address for the subscriber session. An IP subscriber session begins when the router receives a DHCP discover packet for the subscriber and an IP address is assigned for the subscriber. The session is terminated when the subscriber receives a DHCP release message and its IP address is released. If the subscriber session is VRF aware (that is, if the subscriber belongs to a VRF), the VRF-aware DHCP pool must be used.

Note

- The router can be operating as a DHCP server or DHCP relay device.

- To configure an IP subscriber as part of a VRF (to make the subscriber session VRF aware), configure the VRF under the access subinterface.

This feature supports the following sessions in a ES+ line card:

- IP sessions (routed and L2-connected)
- DHCP integration with IP sessions
- Static IP subnet sessions
- Source IP address and MAC address sessions (IP sessions)
- PPPoE supported in the PPP Termination and Aggregation (PTA) mode
- PPPoEoVLAN supported in the PTA mode
- PPPoEoQinQ supported in the PTA mode
- PPPoEoDot1Q supported in the PTA mode
- For information on IP and PPPoE Session Coexistence with Multicast, see Configuring Multicast Features chapter in the Cisco 7600 Series Ethernet Services Plus (ES+) and Ethernet Services Plus T (ES+T) Line Card Configuration Guide.

IP Subscriber Session Features

The following features are provided for IP subscriber sessions:

- Per-subscriber Radius accounting—Enables system administrators to track IP session activity for individual subscribers, and to extract subscriber accounting records periodically. Per-subscriber Radius accounting works with DHCP IP address assignment, and improves the authentication, authorization, and accounting (AAA) of broadband service delivery. For more information, see its feature description at:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/newft/122srb33/ipradacc.htm

- Lawful intercept—Enables a Law Enforcement Agency (LEA) to perform electronic surveillance on a subscriber as authorized by a court order. To assist in the surveillance, the service provider intercepts the subscriber's traffic as it passes through one of their routers, and sends a copy of the intercepted traffic to the LEA without the subscriber's knowledge. For more information, see the documents at the following URLs:

  http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/76licfg/index.htm

- Quality of Service—Standard QoS features are supported for individual subscribers (access subinterfaces), including classification, marking, policing, shaping, priority queuing, and weighted random early detection (WRED). For information about recommended QoS settings for IP Subscriber Awareness over Ethernet, see the following section ("QoS Recommendations"). For information about QoS features on the Cisco 7600 SIP-400, see the information about QoS features in the "Cisco 7600 SIP-400 Features" section of the document at this URL:

  http://www.cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/7600series/sipspasw.html

In addition to standard QoS features, the following new Cisco 7600 SIP-400 QoS features are being introduced to support the deployment of broadband services:

- Dual-priority queues—Provide two priority queues for voice and video traffic for 4000 to 8000 subscribers. You can assign a different priority level to each traffic class to configure the router to treat both types of traffic as priority traffic but to handle them differently (for example, by giving voice traffic precedence over video traffic).
- Bandwidth-remaining ratio (BRR)—Allows service providers to prioritize subscriber traffic during periods of congestion. You can use the Distribution of Remaining Bandwidth Using Ratio feature to specify the relative weight of a subinterface or class queue with respect to other subinterfaces or queues. For information about this feature, see the "Bandwidth-Remaining Ratio Recommendations" section on page 22-21.
- Priority-rate propagation—Takes the priority level and traffic rate assigned to priority traffic in a low-level queue and applies that level and rate to priority traffic at all higher-level queues in the queue hierarchy, even if those queues are not specifically configured for minimum rates or priority. For more information, see the "Priority-Rate Propagation Recommendations" section on page 22-25.

IP Address Assignment

- DHCP Based IP address assignment: If DHCP is used to assign IP addresses, and the IP address assigned by DHCP is correct for the service domain, ISG is not involved in the assignment of an IP address for the subscriber. If the IP address that is assigned by DHCP incorrect or if the domain changes due to a VRF transfer, ISG can be configured to influence the DHCP IP address assignment.
- Static IP address assignment: If a subscriber's static IP address is configured correctly for the service domain, ISG is not involved in the assignment of an IP address for the subscriber.
- IP subnet: For IP subnet sessions, the IP subnet is specified in the user profile.
- IP interface: ISG is not involved in the assignment of subscriber IP addresses.

IP Subnet (IP Range) Sessions

A client subnet identifies an IP Subnet session and applies uniform edge processing to packets associated with a particular IP subnet. IP Subnet sessions are hosted for clients directly connected or over multiple hops. The following functionalities are not supported on IP Subnet Sessions, but are supported on IP Sessions:

- DHCP session initiation not supported
- No Source MAC address session support
- No Dynamic VPN selection support

IP Interface Sessions

In an IP Interface session, all the traffic received on a particular physical or logical interface is collated. However, dynamic VRF transfer is not supported in an IP interface session, and the VRF transfer can only be used with static VRF configuration. Irrespective of the subsriber logged in, a session is created by default.

PPPoE and IPoE Session Support on Port Channel (1:1 Redundancy)

The 1:1 redundancy on a port channel, coupled with Link Aggregation Control Protocol (LACP), dynamically handles the member links in a port channel bundle. A port channel has two members, of which one member is active and the other is in standby or redundant mode. The member ports can be located on any line cards, but must originate from Ethernet Services Plus (ES+) line card. At any given point of time, one link is on the physical mode.

The following sessions support 1:1 redundancy on an ES+ line card:

- IP Subnet sessions
- IP Interface sessions
- PPPoEoX sessions.

PPPoE and IPoE Session Support on QinQ Subinterfaces with IEEE 802.1AH Customer Ethertype

This feature enables you to implement PPPoE and IPoE session (ISG functions) on QinQ subinterfaces that are configured with custom ethertype. The custom ethertype implemented on the main interface is inherited by all the subinterfaces. To implement this feature, use the **dot1q tunnel ethertype** command on the main interface for the respective QinQ subinterfaces.

Packets are accepted if the outer VLAN tag on a PPPoE or IPoE session packet matches the custom ethertype VLAN settings on the QinQ subinterface, else dropped. You can set the outer VLAN tag to the following values:

- 0x9100
- 0x9200
- ox8100
- 0x88a8

The PPPoE or IPoE session does not come up if tthere is mismatch in the ethertype between ISG and the client. For example, if the outer VLAN tag on a packet is set to 0x9100, and the interface is configured using custom ethertype to accept only packets with 0x88a8 VLAN tag, the packet is dropped in the QinQ subinterface.

You can create a QinQ subinterface using the access keyword, while defining an interface. The following code shows how to define an interface with the access keyword, create a VLAN QinQ subinterface, and enable PPPoE session:

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitethernet 1/0/0
Router(config-if)# dot1q tunneling ethertype 0x9100
Router(config-if)# interface gigabitethernet 1/0/0.100 access
Router(config-subif)# encapsulation dot1q 100 second-dot1q 200
Router(config-subif)# ip subscriber interface
```

Restictions and Usage Guidelines

Follow these restrictions and usage guidelines when you configure an IP or a PPPoE sessions on an ES+ linecard:

- IP Sessions are not supported on ambiguous VLANs.

- Radius proxy is not supported for the IP Sessions.
- IP and MAC address spoof Prevention is not supported on subinterfaces on a ES+ linecard unlike on a SIP400 line card.
- IP sessions are supported on Link Aggregation (Ether-Channel) interfaces. LAG etherchannel interfaces are supported for links on the same and across line cards.
- PPPoE sessions are supported on ambiguous VLAN interfaces and VLAN ranges.
- There are no drop counters to identify the number of packets dropped due to custom ethertype mismatch.
- VLANs, Source MAC Address, and Ports are matched against session IDs to extend security for PPPoE sessions.

Follow these restrictions and usage guidelines when you configure 1:1 redundancy on an ES+ linecard:

- Subscriber redundancy is available only on a 1:1 access standby model.
- Supports access interfaces in port channels to scale the number of port channel subinterfaces to greater than 4000.
- Link Aggregation Control Protocol (LACP) allows dynamic handling of member links in a GEC bundle.
- Supports a maximum of 64 GEC bundles with 8 links.
- Member links in a single GEC bundle reside across NPs or the linecard.
- LAG is supported with members across linecards.
- Supports LAG across linecards and membership of the LAG does not change after new sessions are initiated.
- Feature supports 32000 access subinterfaces and 8K access interfaces.
- Supports per session load balancing across member links where all the traffic for a session is relayed over a single port.
- To reduce the downtime during member link addition or deletion, QOS queues are allocated for all member links belonging to the port channel. Though the ingress and egress traffic could be on different member links, the peer relays all the traffic for a session through a single member link.
- LAG supports sessions on non access subinterfaces to support coexistence of multicast streams.

Verification

This section lists the commands to display configuration information.

- Use the following commands to configure the PPPoE:

```
Router-DJ4-dfc9#sh debug
CWAN iEdge LC:
  CWAN iEdge LC session event debug debugging is on
X40G XLIF Client:
  XLIF NP events debugging is on

Router-DJ4-dfc9# sh log
Syslog logging: enabled (0 messages dropped, 4 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.
No Inactive Message Discriminator.
```

```
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 308 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

Log Buffer (1000000 bytes):

Nov 19 16:08:48.247 IST: DFC9: provision_pppoe_routed_ac: switch_info 2CDEC4A4
seghandle 2CD93474 uid 40 if_number 80
Nov 19 16:08:48.247 IST: DFC9:  type 1 2 0opaque handle = 0x186DAB48
Nov 19 16:08:48.247 IST: DFC9: inserting 186DAB48 105 40
Nov 19 16:08:48.247 IST: DFC9: cwan_iedge_session_pending_timer started
Nov 19 16:08:48.247 IST: DFC9: no dbus vlan session pending on int 105
Nov 19 16:08:48.251 IST: DFC9:  cwan_iedge_update_dbus_vlan: Session 40 gets hidden
vlan 1020 through update for Virtual-Access2.1
Nov 19 16:08:50.247 IST: DFC9: cwan_iedge_common_session_notify: cfg_type 2 va_if_num
105 phy_if_num 80 uid 0action 0
Nov 19 16:08:50.247 IST: DFC9: cwan_iedge_get_session_config: sess_type 2 if_num 105
pid 0
Nov 19 16:08:50.247 IST: DFC9: cwan_iedge_get_pppoe_config: if_num 80 va_if_num 105
vlan 1020 sess-id 40 cond_debug off
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_create Cfn[965F2BC] Creating Xlif:
GigabitEthernet9/5 Xid[0] Typ[4] Ch[0] Ifn[105] Xreg[0] Xidx[205352] efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_create_internal successfully created
xlif: GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0] Xidx[205352] efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_eg_xlif_update_port Cfn[92D1658] Xlif Update
Port 4 : GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0] Xidx[205352]
efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_update_tag_rewrite Cfn[965F334] Tag(i-0,
o-2) Dir[2]: GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0] Xidx[205352]
efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_update_dbus_vlan Cfn[965F36C] Updatng
Dbus Vlan 1020: GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0]
Xidx[205352] efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_update_stats_id Cfn[965D780] Updatng
StatId 599056 Dir[0]: GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0]
Xidx[205352] efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_update_stats_id Cfn[965D8A8] Updatng
StatId 599064 Dir[1]: GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0]
Xidx[205352] efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_fwd_feat_enable Cfn[965F3BC] Xlif Fwd
Feat 0x1 Enable 1 : GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0]
Xidx[205352] efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_enable Cfn[965F3F0] Xlif Enable 1:
GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0] Xidx[205352] efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_update_feat_info Cfn[965F604] Xlif update
feature Dir[0]: GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0]
Xidx[205352] efp[0]
Nov 19 16:08:50.247 IST: DFC9: x40g_npc_xlif_update_feat_info Cfn[965F700] Xlif update
feature Dir[1]: GigabitEthernet9/5 Xid[205352] Typ[4] Ch[0] Ifn[105] Xreg[0]
Xidx[205352] efp[0]

Router-DJ4#sh debug
PPP:
  PPP protocol negotiation debugging is on
PPPoE:
  PPPoE protocol events debugging is on
  PPPoE control packets debugging is on
```

```
Router-DJ4#sh log
Syslog logging: enabled (3340 messages dropped, 2 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 5280 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 203 message lines logged

Log Buffer (1000000 bytes):

Nov 19 16:08:48.231 IST: PPPoE 0: I PADI  R:bb00.1912.0001 L:ffff.ffff.ffff 2 Gi9/5.1
contiguous pak, size 60
 FF FF FF FF FF FF BB 00 19 12 00 01 81 00 00 02
 88 63 11 09 00 00 00 04 01 01 00 00 00 0A 03 06
 B6 00 00 01 00 00 00 01 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 06 F8 00 00 9C 88
Nov 19 16:08:48.231 IST:  Service tag: NULL Tag
Nov 19 16:08:48.231 IST: PPPoE 0: O PADO, R:a110.0050.0006 L:bb00.1912.0001 1019
Gi9/5.1
Nov 19 16:08:48.231 IST:  Service tag: NULL Tag
contiguous pak, size 100
 06 02 00 10 03 FB 28 00 03 80 00 00 44 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 02 04 00 00
 BB 00 19 12 00 01 A1 10 00 50 00 06 81 00 00 02
 88 63 11 07 00 00 00 24 01 01 00 00 01 02 00 08
 52 69 61 7A 2D 44 4A 34 ...
Nov 19 16:08:48.231 IST: PPPoE 0: I PADR  R:bb00.1912.0001 L:000c.31c9.7000 2 Gi9/5.1
contiguous pak, size 60
 00 0C 31 C9 70 00 BB 00 19 12 00 01 81 00 00 02
 88 63 11 19 00 00 00 18 01 01 00 00 01 04 00 10
 E2 DB 75 8D E5 9C 95 C1 83 35 DC 91 B2 14 32 89
 63 63 65 73 73 2D 70 70 6C 63 70 30
Nov 19 16:08:48.231 IST:  Service tag: NULL Tag
Nov 19 16:08:48.231 IST: PPPoE : encap string prepared
Nov 19 16:08:48.231 IST: [40]PPPoE 40: Access IE handle allocated
Nov 19 16:08:48.231 IST: [40]PPPoE 40: AAA get retrieved attrs
Nov 19 16:08:48.231 IST: [40]PPPoE 40: AAA get nas port details
Nov 19 16:08:48.231 IST: [40]PPPoE 40: AAA get dynamic attrs
Nov 19 16:08:48.231 IST: [40]PPPoE 40: AAA unique ID allocated
Nov 19 16:08:48.231 IST: [40]PPPoE 40: No AAA accounting method list
Nov 19 16:08:48.231 IST: [40]PPPoE 40: Service request sent to SSS
Nov 19 16:08:48.231 IST: [40]PPPoE 40: Created, Service: None R:000c.31c9.7000
L:bb00.1912.0001 2 Gi9/5.1
Nov 19 16:08:48.231 IST: [40]PPPoE 40: State NAS_PORT_POLICY_INQUIRY    Event SSS MORE
KEYS
Nov 19 16:08:48.231 IST: PPP: Alloc Context [19C03860]
Nov 19 16:08:48.231 IST: ppp40 PPP: Phase is ESTABLISHING
Nov 19 16:08:48.231 IST: [40]PPPoE 40: data path set to PPP
Nov 19 16:08:48.231 IST: [40]PPPoE 40: Segment (SSS class): PROVISION
Nov 19 16:08:48.231 IST: [40]PPPoE 40: State PROVISION_PPP    Event SSM PROVISIONED
```

```
Nov 19 16:08:48.231 IST: [40]PPPoE 40: O PADS  R:bb00.1912.0001 L:000c.31c9.7000 1019
Gi9/5.1
contiguous pak, size 100
 00 02 00 10 03 FB 28 00 03 80 00 00 44 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 02 04 00 00
 BB 00 19 12 00 01 A1 10 00 50 00 06 81 00 00 02
 88 63 11 65 00 28 00 18 01 01 00 00 01 04 00 10
 E2 DB 75 8D E5 9C 95 C1 ...
Nov 19 16:08:48.231 IST: ppp40 PPP: Using vpn set call direction
Nov 19 16:08:48.231 IST: ppp40 PPP: Treating connection as a callin
Nov 19 16:08:48.231 IST: ppp40 PPP: Session handle[28] Session id[40]
Nov 19 16:08:48.231 IST: ppp40 LCP: Event[OPEN] State[Initial to Starting]
Nov 19 16:08:48.231 IST: ppp40 PPP LCP: Enter passive mode, state[Stopped]
Nov 19 16:08:48.231 IST: ppp40 LCP: I CONFREQ [Stopped] id 0 len 14
Nov 19 16:08:48.231 IST: ppp40 LCP:    MagicNumber 0xA4E30BAF (0x0506A4E30BAF)
Nov 19 16:08:48.231 IST: ppp40 LCP:    MRU 1492 (0x010405D4)
Nov 19 16:08:48.231 IST: ppp40 LCP: O CONFREQ [Stopped] id 1 len 19
Nov 19 16:08:48.231 IST: ppp40 LCP:    MRU 1492 (0x010405D4)
Nov 19 16:08:48.231 IST: ppp40 LCP:    AuthProto CHAP (0x0305C22305)
Nov 19 16:08:48.235 IST: ppp40 LCP:    MagicNumber 0x0F501712 (0x05060F501712)
Nov 19 16:08:48.235 IST: ppp40 LCP: O CONFACK [Stopped] id 0 len 14
Nov 19 16:08:48.235 IST: ppp40 LCP:    MagicNumber 0xA4E30BAF (0x0506A4E30BAF)
Nov 19 16:08:48.235 IST: ppp40 LCP:    MRU 1492 (0x010405D4)
Nov 19 16:08:48.235 IST: ppp40 LCP: Event[Receive ConfReq+] State[Stopped to ACKsent]
Nov 19 16:08:48.235 IST: ppp40 LCP: I CONFACK [ACKsent] id 1 len 19
Nov 19 16:08:48.235 IST: ppp40 LCP:    MRU 1492 (0x010405D4)
Nov 19 16:08:48.235 IST: ppp40 LCP:    AuthProto CHAP (0x0305C22305)
Nov 19 16:08:48.235 IST: ppp40 LCP:    MagicNumber 0x0F501712 (0x05060F501712)
Nov 19 16:08:48.235 IST: ppp40 LCP: Event[Receive ConfAck] State[ACKsent to Open]
Nov 19 16:08:48.243 IST: ppp40 PPP: Phase is AUTHENTICATING, by this end
Nov 19 16:08:48.243 IST: ppp40 CHAP: O CHALLENGE id 1 len 29 from "Router-DJ4"
Nov 19 16:08:48.243 IST: ppp40 LCP: State is Open
Nov 19 16:08:48.243 IST: ppp40 CHAP: I RESPONSE id 1 len 29 from "PPP_USER"
Nov 19 16:08:48.243 IST: ppp40 PPP: Phase is FORWARDING, Attempting Forward
Nov 19 16:08:48.243 IST: ppp40 PPP: Phase is AUTHENTICATING, Unauthenticated User
Nov 19 16:08:48.243 IST: ppp40 IPCP: Authorizing CP
Nov 19 16:08:48.243 IST: ppp40 IPCP: CP stalled on event[Authorize CP]
Nov 19 16:08:48.243 IST: ppp40 IPCP: CP unstall
Nov 19 16:08:48.243 IST: ppp40 PPP: Phase is FORWARDING, Attempting Forward
Nov 19 16:08:48.243 IST: [40]PPPoE 40: State LCP_NEGOTIATION    Event SSS CONNECT
LOCAL
Nov 19 16:08:48.247 IST: [40]PPPoE 40: Segment (SSS class): UPDATED
Nov 19 16:08:48.247 IST: [40]PPPoE 40: Segment (SSS class): BOUND
Nov 19 16:08:48.247 IST: [40]PPPoE 40: data path set to Virtual Acess
Nov 19 16:08:48.247 IST: [40]PPPoE 40: State LCP_NEGOTIATION    Event SSM UPDATED
Nov 19 16:08:48.247 IST: Vi2.1 PPP: Phase is AUTHENTICATING, Authenticated User
Nov 19 16:08:48.247 IST: Vi2.1 CHAP: O SUCCESS id 1 len 4
Nov 19 16:08:48.247 IST: [40]PPPoE 40: AAA get dynamic attrs
Nov 19 16:08:48.247 IST: Vi2.1 PPP: Phase is UP
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: Protocol configured, start CP. state[Initial]
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: Event[OPEN] State[Initial to Starting]
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: O CONFREQ [Starting] id 1 len 10
Nov 19 16:08:48.247 IST: Vi2.1 IPCP:    Address 100.0.0.1 (0x030664000001)
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: Event[UP] State[Starting to REQsent]
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: I CONFREQ [REQsent] id 0 len 10
Nov 19 16:08:48.247 IST: Vi2.1 IPCP:    Address 0.0.0.0 (0x030600000000)
Nov 19 16:08:48.247 IST: Vi2.1 IPCP AUTHOR: Start.  Her address 0.0.0.0, we want
0.0.0.0
Nov 19 16:08:48.247 IST: Vi2.1 IPCP AUTHOR: Done.  Her address 0.0.0.0, we want
0.0.0.0
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: Pool returned 182.0.0.1
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: O CONFNAK [REQsent] id 0 len 10
Nov 19 16:08:48.247 IST: Vi2.1 IPCP:    Address 182.0.0.1 (0x0306B6000001)
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: Event[Receive ConfReq-] State[REQsent to REQsent]
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: I CONFACK [REQsent] id 1 len 10
```

```
Nov 19 16:08:48.247 IST: Vi2.1 IPCP:    Address 100.0.0.1 (0x030664000001)
Nov 19 16:08:48.247 IST: Vi2.1 IPCP: Event[Receive ConfAck] State[REQsent to ACKrcvd]
Nov 19 16:08:48.251 IST: [40]PPPoE 40: State PTA_BINDING    Event STATIC BIND RESPONSE
Nov 19 16:08:48.251 IST: [40]PPPoE 40: Connected PTA
Nov 19 16:08:48.251 IST: Vi2.1 IPCP: I CONFREQ [ACKrcvd] id 1 len 10
Nov 19 16:08:48.251 IST: Vi2.1 IPCP:    Address 182.0.0.1 (0x0306B6000001)
Nov 19 16:08:48.251 IST: Vi2.1 IPCP: O CONFACK [ACKrcvd] id 1 len 10
Nov 19 16:08:48.251 IST: Vi2.1 IPCP:    Address 182.0.0.1 (0x0306B6000001)
Nov 19 16:08:48.251 IST: Vi2.1 IPCP: Event[Receive ConfReq+] State[ACKrcvd to Open]
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: Event[DOWN] State[Open to Starting]
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: Event[CLOSE] State[Starting to Initial]
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: Event[OPEN] State[Initial to Starting]
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: O CONFREQ [Starting] id 2 len 10
Nov 19 16:08:48.255 IST: Vi2.1 IPCP:    Address 100.0.0.1 (0x030664000001)
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: Event[UP] State[Starting to REQsent]
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: I CONFREQ [REQsent] id 2 len 10
Nov 19 16:08:48.255 IST: Vi2.1 IPCP:    Address 182.0.0.1 (0x0306B6000001)
Nov 19 16:08:48.255 IST: Vi2.1 IPCP AUTHOR: Start.  Her address 182.0.0.1, we want
182.0.0.1
Nov 19 16:08:48.255 IST: Vi2.1 IPCP AUTHOR: Reject 182.0.0.1, using 182.0.0.1
Nov 19 16:08:48.255 IST: Vi2.1 IPCP AUTHOR: Done.  Her address 182.0.0.1, we want
182.0.0.1
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: O CONFACK [REQsent] id 2 len 10
Nov 19 16:08:48.255 IST: Vi2.1 IPCP:    Address 182.0.0.1 (0x0306B6000001)
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: Event[Receive ConfReq+] State[REQsent to ACKsent]
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: I CONFACK [ACKsent] id 2 len 10
Nov 19 16:08:48.255 IST: Vi2.1 IPCP:    Address 100.0.0.1 (0x030664000001)
Nov 19 16:08:48.255 IST: Vi2.1 IPCP: Event[Receive ConfAck] State[ACKsent to Open]
Nov 19 16:08:48.275 IST: Vi2.1 IPCP: State is Open (Indicates that the PPPoE session
is up)
Nov 19 16:08:48.275 IST: Vi2.1 Added to neighbor route AVL tree: topoid 0, address
182.0.0.1
Nov 19 16:08:48.275 IST: Vi2.1 IPCP: Install route to 182.0.0.1
Router-DJ4#

interface GigabitEthernet9/17.1
 encapsulation dot1Q 2000
 ip address 180.0.0.1 255.255.255.0

interface GigabitEthernet9/5.1
 encapsulation dot1Q 2
 ip address 192.0.0.1 255.255.255.0
 pppoe enable group dj4_bba_group1

aaa new-model
aaa authentication login default group radius local
aaa authentication ppp default local
aaa authorization network default local
aaa authorization subscriber-service default group radius
aaa session-id common

bba-group pppoe dj4_bba_group1
 virtual-template 1
 sessions per-vc limit 16000
 sessions per-mac limit 16000
 sessions per-vlan limit 8000

interface Loopback1
 ip address 100.0.0.1 255.255.255.255

interface Virtual-Template1
 ip unnumbered Loopback1
 no logging event link-status
 peer default ip address pool PPPPool_1
```

```
 no snmp trap link-status
 keepalive 300
 ppp authentication chap
```

Use the following commands to verify the PPPoE session:

```
Router-DJ4#sh pppoe summary
    PTA  : Locally terminated sessions
    FWDED: Forwarded sessions
    TRANS: All other sessions (in transient state)

                                TOTAL     PTA   FWDED   TRANS
TOTAL                               1       1       0       0
GigabitEthernet9/5                  1       1       0       0
Router-DJ4#sh pppoe ses
Router-DJ4#sh pppoe session
     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID  PPPoE  RemMAC          Port                    VT  VA         State
           SID  LocMAC                                      VA-st      Type
     42     42  bb00.1912.0001  Gi9/5.1                  1  Vi2.1      PTA
                000c.31c9.7000  VLAN:   2                   UP
Router-DJ4#sh sss session uid 42 detailed
Unique Session ID: 42
Identifier: PPP_USER
SIP subscriber access type(s): PPPoE/PPP
Current SIP options: Req Fwding/Req Fwded
Session Up-time: 00:19:04, Last Changed: 00:19:04
Interface: Virtual-Access2.1

Policy information:
  Context 137426FC: Handle 2400002A
  AAA_id 00000038: Flow_handle 0
  Authentication status: authen
  Downloaded User profile, excluding services:
    Framed-Protocol      1 [PPP]
    username             "PPP_USER"
  Downloaded User profile, including services:
    Framed-Protocol      1 [PPP]
    username             "PPP_USER"
  Config history for session (recent to oldest):
    Access-type: PPP Client: SM
     Policy event: Process Config Connecting
      Profile name: apply-config-only, 2 references
        Framed-Protocol      1 [PPP]
        username             "PPP_USER"
  Rules, actions and conditions executed:
    subscriber rule-map PPPoE-SUB
      condition always event session-start
        1 service local

Configuration sources associated with this session:
Interface: Virtual-Template1, Active Time = 00:19:04

Router-DJ4# sh pppoe session packets
Total PPPoE sessions 1


SID     Pkts-In         Pkts-Out        Bytes-In        Bytes-Out
42      12              13              184             190
Router-DJ4#
```

```
Router-DJ4#sh cef int gig 9/5.1
GigabitEthernet9/5.1 is up (if_number 80)
  Corresponding hwidb fast_if_number 80
  Corresponding hwidb firstsw->if_number 25
  Internet address is 192.0.0.1/24
  ICMP redirects are always sent
  IP unicast RPF check is disabled
  Output features: MFIB Adjacency, HW Shortcut Installation
  IP policy routing is disabled
  BGP based policy accounting on input is disabled
  BGP based policy accounting on output is disabled
  Hardware idb is GigabitEthernet9/5
  Fast switching type 28, interface type 146
  IP CEF switching enabled
  IP CEF switching turbo vector
  IP Null turbo vector
  IP prefix lookup IPv4 mtrie generic
  Input fast flags 0x40000000, Output fast flags 0x0
  ifindex 24(24)
  Slot 9/0 (9) Slot unit 5 VC -1
  IP MTU 1500
```

- Use the following commands to configure IP session:

```
aaa new-model
!
aaa session-id common
!
interface GigabitEthernet2/9
 no ip address
 load-interval 30
!
interface GigabitEthernet2/9.1 access
 encapsulation dot1Q 2 second-dot1q 2
 ip address 182.0.0.1 255.255.255.0
 ip subscriber routed
  initiator unclassified ip-address
!
interface GigabitEthernet2/10
 no ip address
 load-interval 30
!
interface GigabitEthernet2/10.1
 encapsulation dot1Q 2000 second-dot1q 2001
 ip address 180.0.0.1 255.255.255.0
!
no ip http server
no ip http secure-server
!
arp 182.0.0.2 aa00.0000.0001 ARPA
arp 180.0.0.2 0000.0000.0001 ARPA
!
```

Use the following commands to debug IP session:

```
ISG_NMB#sh deb
CWAN iEdge RP:
  CWAN iEdge RP debug debugging is on

IP Subscriber:
  all IP subscriber debugs debugging is on
ISG_NMB#
Nov 19 16:02:46.087 IST: IPSUB_DP: [Gi2/9.1:I:CEF:DFL:21.0.0.1] Packet triggers
session initiation
```

```
Nov 19 16:02:46.087 IST: IPSUB_DP: [Gi2/9.1:I:CEF:DFL:21.0.0.1] Packet classified,
results = 0x1
Nov 19 16:02:46.087 IST: IPSUB_DP: [uid:0] Insert new entry for mac 0000.1500.0001
Nov 19 16:02:46.087 IST: IPSUB_DP: [uid:0] Processing new in-band session request
Nov 19 16:02:46.087 IST: IPSUB_DP: [uid:0] Delete mac entry 0000.1500.0001
Nov 19 16:02:46.087 IST: IPSUB_DP: [uid:0] In-band session request event for session
Nov 19 16:02:46.087 IST: IPSUB_DP: [uid:0] Added upstream entry into the classifier
Nov 19 16:02:46.087 IST: IPSUB_DP: [uid:0] VRF = DFL, IP = 21.0.0.1, MASK =
255.255.255.255
Nov 19 16:02:46.087 IST: IPSUB: Try to create a new session
Nov 19 16:02:46.087 IST: IPSUB: IPSUB: Check IP DHCP session recovery: 21.0.0.1
Gi2/9.1 mac aa00.0000.0001
Nov 19 16:02:46.087 IST: IPSUB: IPSUB: No DHCP binding found
Nov 19 16:02:46.087 IST: IPSUB: [uid:0] IPSUB: Proceed to create the IP inband session
Nov 19 16:02:46.087 IST: IPSUB: [uid:0] Request to create a new session
Nov 19 16:02:46.087 IST: IPSUB: [uid:0] Session start event for session
Nov 19 16:02:46.087 IST: IPSUB: [uid:0] Event session start, state changed from idle
to requesting
Nov 19 16:02:46.087 IST: IPSUB: HA[uid:32]: Session init-notification on Active
Nov 19 16:02:46.087 IST: IPSUB: HA[uid:32]: Allocated SHDB handle (0xF1000020)
Nov 19 16:02:46.087 IST: IPSUB: HA[uid:32]: Successfully initialized for HA
Nov 19 16:02:46.087 IST: IPSUB: [uid:32] AAA unique ID allocated
Nov 19 16:02:46.087 IST: IPSUB: [uid:32] Added session 21.0.0.1 to L3 session table
Nov 19 16:02:46.087 IST: IPSUB: [uid:32] Added session to session table with access
session keys
Nov 19 16:02:46.087 IST: IPSUB: [uid:32] IP session(0x63000020) to be associated to
Gi2/9.1
Nov 19 16:02:46.087 IST: IPSUB: [uid:32] Inserted IP session(0x63000020) to
sessions-per-interface db with interface Gi2/9.1
Nov 19 16:02:46.087 IST: IPSUB_DP: [uid:0] Sent message to control plane for in-band
session creation
Nov 19 16:02:46.087 IST: IPSUB_DP: [uid:0] Event inband-session, state changed from
idle to intiated
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Recieved Message = connect local
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Connect Local event for session
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Event connect local, state changed from
requesting to waiting
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Inside processing IPSIP info
Nov 19 16:02:46.091 IST: IPSUB-ROUTE: [uid:32] Checking whether routes to be
inserted/removed
Nov 19 16:02:46.091 IST: IPSUB-ROUTE: [uid:32] Context not present, creating context
Nov 19 16:02:46.091 IST: IPSUB-ROUTE: [uid:32] Entered the sg subrte context alloc
Nov 19 16:02:46.091 IST: IPSUB-ROUTE: [uid:32] Returning the sg subrte context
0x1348DD20
Nov 19 16:02:46.091 IST: IPSUB-ROUTE: [uid:32] Added Fib Prefix [DFL]:
21.0.0.1/255.255.255.255
Nov 19 16:02:46.091 IST: IPSUB-ROUTE: [uid:32] Both IP addresses and VRF are same, no
need to add route
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Keys not changed, seg needn't be updated
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Key list to be created to update SM
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Created key list to update SM
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Session Keys Available event for session
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Event session keys available, state changed
from waiting to provisioning
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Access and service keys same, no need to add
session with service keys
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Data plane prov successful event for session
Nov 19 16:02:46.091 IST: IPSUB: [uid:32] Event dataplane prov successful, state
changed from provisioning to connected
Nov 19 16:02:46.091 IST: IPSUB: HA[uid:32]: Session up notification
Nov 19 16:02:46.091 IST: IPSUB: HA[uid:32]: Session ready to sync data (0xF1000020)
Nov 19 16:02:46.091 IST: IPSUB_DP: [uid:0] Setup event for session (session hdl
3858759691)
Nov 19 16:02:46.091 IST: IPSUB_DP: [uid:32] Added downstream entry into the classifier
```

```
Nov 19 16:02:46.091 IST: IPSUB_DP: [uid:32] VRF = DFL, IP = 21.0.0.1, MASK =
255.255.255.255
Nov 19 16:02:46.091 IST: IPSUB_DP: [uid:32] Session setup successful
Nov 19 16:02:46.091 IST: IPSUB_DP: [uid:32] Event setup-session, state changed from
intiated to established
Nov 19 16:02:46.091 IST: IPSUB_DP: [uid:32] Activate event for session
Nov 19 16:02:46.091 IST: IPSUB_DP: [uid:32] Event activate-session, state changed from
established to connected
```

Use the following commands to verify IP session:

```
ISG_NMB#sh ip sub
Displaying subscribers in the default service vrf:

Type        Subscriber Identifier    Display UID    Status
---------   ----------------------   ------------   ------
routed      21.0.0.1/32              [32]           up
ISG_NMB#
ISG_NMB#sh sss sess
Current Subscriber Information: Total sessions 1

Uniq ID Interface    State         Service      Identifier           Up-time
32      IP           unauthen      Local Term   21.0.0.1             00:02:40

ISG_NMB#sh sss sess uid 32
Unique Session ID: 32
Identifier: 21.0.0.1
SIP subscriber access type(s): IP
Current SIP options: Req Fwding/Req Fwded
Session Up-time: 00:02:46, Last Changed: 00:02:46

Policy information:
  Authentication status: unauthen

Configuration sources associated with this session:
Interface: GigabitEthernet2/9.1, Active Time = 00:02:46

ISG_NMB#sh sss sess uid 32 de
ISG_NMB#sh sss sess uid 32 detailed
Unique Session ID: 32
Identifier: 21.0.0.1
SIP subscriber access type(s): IP
Current SIP options: Req Fwding/Req Fwded
Session Up-time: 00:02:49, Last Changed: 00:02:49

Policy information:
  Context 133B22FC: Handle DF000020
  AAA_id 00000030: Flow_handle 0
  Authentication status: unauthen

Configuration sources associated with this session:
Interface: GigabitEthernet2/9.1, Active Time = 00:02:49


Following details is for a L2-connected DHCP session on Dot1Q interface:-
================================================================================
```

Use the following commands to configure L2-connected DHCP session:

```
aaa new-model
!
!
aaa session-id common
```

```
!
!
!
clock timezone IST 5
ip source-route
!
!
ip dhcp excluded-address 182.0.0.11 182.0.0.15
no ip dhcp ping packets
!
ip dhcp pool pool_global1
   network 182.0.0.0 255.255.255.240
   lease 0 0 3
   update arp
!
!
!
interface Loopback10
 ip address 182.0.0.11 255.255.255.255
!
!
interface GigabitEthernet2/9
 no ip address
 load-interval 30
!
interface GigabitEthernet2/9.1 access
 encapsulation dot1Q 2
 ip unnumbered Loopback10
 ip subscriber l2-connected
  initiator dhcp class-aware
!
interface GigabitEthernet2/10
 no ip address
 load-interval 30
!
interface GigabitEthernet2/10.1
 encapsulation dot1Q 2000
 ip address 180.0.0.1 255.255.255.0
!
!
no ip http server
no ip http secure-server
ip route 7.0.0.0 255.0.0.0 7.38.0.1
ip route 202.153.0.0 255.255.0.0 7.38.0.1
!
!
```

Use the following commands to debug L2-connected DHCP session:

```
ISG_NMB#sh deb
DHCP server packet debugging is on.
DHCP server event debugging is on.

IP Subscriber:
  IP subscriber events debugging is on
  IP subscriber errors debugging is on
  IP subscriber packets debugging is on


ISG_NMB#
Nov 19 15:40:33.595 IST: IPSUB_DP: [Gi2/9.1:I:PROC:aa00.1314.0001] Packet classified,
results = 0x40
```

```
Nov 19 15:40:33.595 IST: IPSUB_DP: [Gi2/9.1:I:PROC:aa00.1314.0001] Rx driver allowing
IP routing
Nov 19 15:40:33.595 IST: DHCPD: Reload workspace interface GigabitEthernet2/9.1
tableid 0.
Nov 19 15:40:33.595 IST: DHCPD: tableid for 182.0.0.11 on GigabitEthernet2/9.1 is 0
Nov 19 15:40:33.595 IST: DHCPD: client's VPN is .
Nov 19 15:40:33.595 IST: DHCPD: Sending notification of DISCOVER:
Nov 19 15:40:33.595 IST:   DHCPD: htype 1 chaddr aa00.1314.0001
Nov 19 15:40:33.595 IST:   DHCPD: remote id 020a0000b600000b21010002
Nov 19 15:40:33.595 IST:   DHCPD: interface = GigabitEthernet2/9.1
Nov 19 15:40:33.595 IST:   DHCPD: class id 49786961
Nov 19 15:40:33.595 IST: IPSUB: Create session keys from SSS key list
Nov 19 15:40:33.595 IST: IPSUB: Mac_addr = aa00.1314.0001, Recvd Macaddr =
aa00.1314.0001
Nov 19 15:40:33.599 IST: IPSUB: Session input interface(0x13348754) =
GigabitEthernet2/9.1
Nov 19 15:40:33.599 IST: IPSUB: SHDB Handle = 5A00000B
Nov 19 15:40:33.599 IST: IPSUB: Remote_id = 020a0000b600000b21010002
Nov 19 15:40:33.599 IST: IPSUB: Vendor_Class_id = Ixia
Nov 19 15:40:33.599 IST: DHCPD: DHCPDISCOVER received from client 01aa.0013.1400.01 on
interface GigabitEthernet2/9.1.
Nov 19 15:40:33.599 IST: DHCPD: Sending notification of DISCOVER:
Nov 19 15:40:33.599 IST:   DHCPD: htype 1 chaddr aa00.1314.0001
Nov 19 15:40:33.599 IST:   DHCPD: remote id 020a0000b600000b21010002
Nov 19 15:40:33.599 IST:   DHCPD: interface = GigabitEthernet2/9.1
Nov 19 15:40:33.599 IST:   DHCPD: class id 49786961
Nov 19 15:40:33.599 IST: DHCPD: Saving workspace (ID=0x8900000B)
Nov 19 15:40:33.599 IST: DHCPD: New packet workspace 0x1333D0D8 (ID=0x2700000C)
Nov 19 15:40:33.599 IST: IPSUB: Try to create a new session
Nov 19 15:40:33.599 IST: IPSUB: [uid:0] Request to create a new session
Nov 19 15:40:33.599 IST: IPSUB: [uid:0] Session start event for session
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] AAA unique ID allocated
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Added session aa00.1314.0001 to L2 session
table
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Added session to session table with access
session keys
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] IP session(0xC500000B) to be associated to
Gi2/9.1
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Inserted IP session(0xC500000B) to
sessions-per-interface db with interface Gi2/9.1
Nov 19 15:40:33.599 IST: DHCPD: Callback for workspace (ID=0x8900000B)
Nov 19 15:40:33.599 IST: DHCPD: No authentication required. Continue
Nov 19 15:40:33.599 IST: DHCPD: Callback: class '' now specified for client
01aa.0013.1400.01
Nov 19 15:40:33.599 IST: DHCPD: Reprocessing saved workspace (ID=0x8900000B)
Nov 19 15:40:33.599 IST: DHCPD: Reload workspace interface GigabitEthernet2/9.1
tableid 0.
Nov 19 15:40:33.599 IST: DHCPD: tableid for 182.0.0.11 on GigabitEthernet2/9.1 is 0
Nov 19 15:40:33.599 IST: DHCPD: client's VPN is .
Nov 19 15:40:33.599 IST: DHCPD: Sending notification of DISCOVER:
Nov 19 15:40:33.599 IST:   DHCPD: htype 1 chaddr aa00.1314.0001
Nov 19 15:40:33.599 IST:   DHCPD: remote id 020a0000b600000b21010002
Nov 19 15:40:33.599 IST:   DHCPD: interface = GigabitEthernet2/9.1
Nov 19 15:40:33.599 IST:   DHCPD: class id 49786961
Nov 19 15:40:33.599 IST: DHCPD: DHCPDISCOVER received from client 01aa.0013.1400.01 on
interface GigabitEthernet2/9.1.
Nov 19 15:40:33.599 IST: DHCPD: Adding binding to radix tree (182.0.0.1)
Nov 19 15:40:33.599 IST: DHCPD: Adding binding to hash tree
Nov 19 15:40:33.599 IST: DHCPD: assigned IP address 182.0.0.1 to client
01aa.0013.1400.01. (13 1)
Nov 19 15:40:33.599 IST: DHCPD: DHCPOFFER notify setup address 182.0.0.1 mask
255.255.255.240
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] IP session context 0x133D28C8 available to
authorize
```

```
Nov 19 15:40:33.599 IST: IPSUB-VRFSET: [uid:11] Entered allocate feature info
Nov 19 15:40:33.599 IST: IPSUB-VRFSET: [uid:11] Allocated sg vrfset info 0x13488EE0
Nov 19 15:40:33.599 IST: IPSUB-VRFSET: [uid:11] Freeing the sg vrfset info 0x13488EE0
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] IPSIP Parsing HostIP: 182.0.0.1 SubnetMask=
255.255.255.255
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Recieved Message = connect local
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Connect Local event for session
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Inside processing IPSIP info
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Processing IPSIP info: 0x1330208C (APPLY)
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Got IP address- IP:-182.0.0.1
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Set IP address- IP:-182.0.0.1
Nov 19 15:40:33.599 IST: IPSUB-VRFSET: [uid:11] Applying SG VRFSET info
Nov 19 15:40:33.599 IST: IPSUB-VRFSET: [uid:11] DHCP Initiated session, no config,
ignore
Nov 19 15:40:33.599 IST: IPSUB-ROUTE: [uid:11] Checking whether routes to be
inserted/removed
Nov 19 15:40:33.599 IST: IPSUB-ROUTE: [uid:11] Context not present, creating context
Nov 19 15:40:33.599 IST: IPSUB-ROUTE: [uid:11] Entered the sg subrte context alloc
Nov 19 15:40:33.599 IST: IPSUB-ROUTE: [uid:11] Returning the sg subrte context
0x1348DD04
Nov 19 15:40:33.599 IST: IPSUB-ROUTE: [uid:11] Installed ARP entry [DFL]: 182.0.0.1
Nov 19 15:40:33.599 IST: IPSUB-ROUTE: [uid:11] Added Fib Prefix [DFL]:
182.0.0.1/255.255.255.255
Nov 19 15:40:33.599 IST: IPSUB-ROUTE: [uid:11] Route insert not required for DHCP
hosts with IP unnumbered config on: GigabitEthernet2/9.1
Nov 19 15:40:33.599 IST: IPSUB-ROUTE: [uid:11] Both IP addresses and VRF are same, no
need to add route
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Found that seg to be updated with new session
keys
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Key list to be created to update SM
Nov 19 15:40:33.599 IST: IPSUB: [uid:11]   Update IP-Address-VRF key: 182.0.0.1:0
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Created key list to update SM
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Found address change to be notified
Nov 19 15:40:33.599 IST: IPSUB: [uid:11] Session Keys Available event for session
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Added session 182.0.0.1 to L3 session table
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Added session to session table with service
session keys
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Recieved Message = update SIP config
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Config Update event for session
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Inside processing IPSIP info
Nov 19 15:40:33.603 IST: IPSUB-ROUTE: [uid:11] Checking whether routes to be
inserted/removed
Nov 19 15:40:33.603 IST: IPSUB-ROUTE: [uid:11] Ctx present, No config change, Nothing
to be done
Nov 19 15:40:33.603 IST: IPSUB-ROUTE: [uid:11] Both IP addresses and VRF are same, no
need to add route
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Keys not changed, seg needn't be updated
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Key list to be created to update SM
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Created key list to update SM
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Data plane prov successful event for session
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Notifying about address change: 182.0.0.1
Nov 19 15:40:33.603 IST: DHCPD: Callback for workspace (ID=0x8900000B)
Nov 19 15:40:33.603 IST: DHCPD: Callback: switching path now setup for client
01aa.0013.1400.01
Nov 19 15:40:33.603 IST: DHCPD: Reprocessing saved workspace (ID=0x8900000B)
Nov 19 15:40:33.603 IST: DHCPD: Sending notification of DISCOVER:
Nov 19 15:40:33.603 IST:   DHCPD: htype 1 chaddr aa00.1314.0001
Nov 19 15:40:33.603 IST:   DHCPD: remote id 020a0000b600000b21010002
Nov 19 15:40:33.603 IST:   DHCPD: interface = GigabitEthernet2/9.1
Nov 19 15:40:33.603 IST:   DHCPD: class id 49786961
Nov 19 15:40:33.603 IST: DHCPD: DHCPDISCOVER received from client 01aa.0013.1400.01 on
interface GigabitEthernet2/9.1.
Nov 19 15:40:33.603 IST: DHCPD: Found previous server binding
```

```
Nov 19 15:40:33.603 IST: DHCPD: Sending DHCPOFFER to client 01aa.0013.1400.01
(182.0.0.1).
Nov 19 15:40:33.603 IST: DHCPD: ARP entry exists (182.0.0.1, aa00.1314.0001).
Nov 19 15:40:33.603 IST: DHCPD: unicasting BOOTREPLY to client aa00.1314.0001
(182.0.0.1).
Nov 19 15:40:33.603 IST: DHCPD: unicast BOOTREPLY output i/f override
GigabitEthernet2/9.1
Nov 19 15:40:33.603 IST: IPSUB_DP: [Gi2/9.1:O:PROC:DFL:182.0.0.1] Packet classified,
results = 0x0
Nov 19 15:40:33.603 IST: DHCPD: removing ARP entry (182.0.0.1 vrf default).
Nov 19 15:40:33.603 IST: DHCPD: Freeing saved workspace (ID=0x8900000B)
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:0] Setup event for session (session hdl 0)
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:0] Insert new entry for mac aa00.1314.0001
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:11] Added upstream entry into the classifier
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:11] MAC = aa00.1314.0001
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:11] Added downstream entry into the classifier
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:11] VRF = DFL, IP = 182.0.0.1, MASK =
255.255.255.255
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:11] Session setup successful
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:11] Sent update msg to the control plane
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:11] Activate event for session
Nov 19 15:40:33.603 IST: IPSUB: [uid:11] Data plane prov successful event for session
Nov 19 15:40:33.603 IST: IPSUB_DP: [uid:0] Found mac entry aa00.1314.0001
Nov 19 15:40:33.603 IST: IPSUB_DP: [Gi2/9.1:I:PROC:aa00.1314.0001] Packet classified,
results = 0x40
Nov 19 15:40:33.603 IST: IPSUB_DP: [Gi2/9.1:I:PROC:aa00.1314.0001] Rx driver allowing
IP routing
Nov 19 15:40:33.603 IST: DHCPD: input i/f override GigabitEthernet2/9.1 for client
Nov 19 15:40:33.603 IST: DHCPD: Reload workspace interface GigabitEthernet2/9.1
tableid 0.
Nov 19 15:40:33.603 IST: DHCPD: tableid for 182.0.0.11 on GigabitEthernet2/9.1 is 0
Nov 19 15:40:33.603 IST: DHCPD: client's VPN is .
Nov 19 15:40:33.603 IST: DHCPD: DHCPREQUEST received from client 01aa.0013.1400.01.
Nov 19 15:40:33.603 IST: DHCPD: Sending notification of ASSIGNMENT:
Nov 19 15:40:33.603 IST:  DHCPD: address 182.0.0.1 mask 255.255.255.240
Nov 19 15:40:33.603 IST:   DHCPD: htype 1 chaddr aa00.1314.0001
Nov 19 15:40:33.603 IST:   DHCPD: lease time remaining (secs) = 180
Nov 19 15:40:33.603 IST:   DHCPD: interface = GigabitEthernet2/9.1
Nov 19 15:40:33.603 IST: DHCPD: Sending DHCPACK to client 01aa.0013.1400.01
(182.0.0.1).
Nov 19 15:40:33.603 IST:  DHCPD: lease time = 180
Nov 19 15:40:33.603 IST: DHCPD: dhcpd_lookup_route: host = 182.0.0.1
Nov 19 15:40:33.603 IST: DHCPD: dhcpd_lookup_route: index = 183
Nov 19 15:40:33.603 IST: DHCPD: dhcpd_create_and_hash_route: host = 182.0.0.1
Nov 19 15:40:33.603 IST: DHCPD: dhcpd_create_and_hash_route index = 183
Nov 19 15:40:33.603 IST: DHCPD: dhcpd_add_route: lease = 180
Nov 19 15:40:33.607 IST: DHCPD: ARP entry exists (182.0.0.1, aa00.1314.0001).
Nov 19 15:40:33.607 IST: DHCPD: Changing arp entry 182.0.0.1 to secure arp entry
Nov 19 15:40:33.607 IST: DHCPD: Failed to secure arp entry 182.0.0.1
Nov 19 15:40:33.607 IST: DHCPD: unicasting BOOTREPLY to client aa00.1314.0001
(182.0.0.1).
Nov 19 15:40:33.607 IST: DHCPD: unicast BOOTREPLY output i/f override
GigabitEthernet2/9.1
Nov 19 15:40:33.607 IST: IPSUB_DP: [Gi2/9.1:O:PROC:DFL:182.0.0.1] Packet classified,
results = 0x10
```

Use the following commands to verify L2-connected DHCP session:

```
ISG_NMB#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
```

```
182.0.0.1           01aa.0013.1400.01       Nov 19 2009 03:45 PM    Automatic

ISG_NMB#sh sss session
Current Subscriber Information: Total sessions 1

Uniq ID Interface    State         Service      Identifier           Up-time
11      IP           unauthen      Local Term   aa00.1314.0001       00:00:58

ISG_NMB#sh sss session uid 11
Unique Session ID: 11
Identifier: aa00.1314.0001
SIP subscriber access type(s): IP
Current SIP options: Req Fwding/Req Fwded
Session Up-time: 00:01:04, Last Changed: 00:01:04

Policy information:
  Authentication status: unauthen

Configuration sources associated with this session:
Interface: GigabitEthernet2/9.1, Active Time = 00:01:04

ISG_NMB#sh sss session uid 11 de
Unique Session ID: 11
Identifier: aa00.1314.0001
SIP subscriber access type(s): IP
Current SIP options: Req Fwding/Req Fwded
Session Up-time: 00:01:07, Last Changed: 00:01:07

Policy information:
  Context 133B2154: Handle 9000000B
  AAA_id 00000017: Flow_handle 0
  Authentication status: unauthen

Configuration sources associated with this session:
Interface: GigabitEthernet2/9.1, Active Time = 00:01:07
```

QoS Recommendations

When you configure QoS features on the Cisco 7600 SIP-400 for use with the IP Subscriber Awareness over Ethernet feature, note the following configuration guidelines and recommendations:

- The Cisco 7600 SIP-400 is capable of throughput of 5.1 to 5.6 gigabits per second (Gbps). We recommend that you do not oversubscribe the card beyond 8 Gbps. Beyond this limit, the card's behavior is unpredictable.
- Oversubscription is supported only on the 5-Port Gigabit Ethernet SPA (SPA-5X1GE-V2).

Note For information on configuring QoS features on ES+ and ES+T Line Cards, see Configuring QoS Features chapter in Cisco 7600 Series Ethernet Services Plus (ES+) and Ethernet Services Plus T (ES+T) Line Card Configuration Guide.

Egress Oversubscription

- High-priority traffic (typically voice and video) must have an IP precedence value of 5, 6, or 7.
- IP precedence values of 0, 1, 2, 3, or 4 will result in drops if oversubscription occurs, even if the traffic is classified as priority traffic in a QoS policy.

Note We recommend that you match the IP precedence and VLAN user priority values of the packets. If ingress oversubscription occurs, priority traffic with non-matching IP precedence and VLAN user priority values might be dropped at the SPA level.

Ingress Oversubscription

- High-priority traffic (typically voice) must have VLAN user priority values of 5, 6, or 7. Priority values of 0, 1, 2, 3, or 4 will result in drops if oversubscription occurs, even if the traffic is classified as priority traffic by a QoS policy.

QoS Counter Updates

- To obtain statistics for an individual IP subscriber session, issue the **show policy-map interface** command two or three times. This is necessary because the counters retain their existing values the first time you issue the command.
- If you issue the **show policy-map interface** command and do not specify an interface, the router must update all of the session counters. With 32000 subscribers, this can take up to 30 minutes.

Bandwidth-Remaining Ratio Recommendations

The Bandwidth-Remaining Ratio (BRR) feature (also called Distribution of Remaining Bandwidth Using Ratio) allows service providers to prioritize subscriber traffic during periods of congestion. You can use the feature to specify the relative weight of a subinterface or class queue with respect to other subinterfaces or queues. During congestion, the router uses the bandwidth-remaining ratio to optimize the scheduling of uncommitted bandwidth on subinterfaces and class queues. Without BRR, the unassigned bandwidth on a physical interface is equally distributed among all queues. For an overview, see the feature description at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sb/newft/122sb31/bwratio.htm

This section provides recommendations and guidelines for configuring BRR on the Cisco 7600 SIP-400 to support IP Subscriber Awareness over Ethernet. It contains the following sections:

BRR Configuration Guidelines

Follow these guidelines to configure BRR:

- Supported only on the Cisco 7600 SIP-400 with 2-port and 5-port Gigabit Ethernet (GE) SPAs.
- Requires RSP720, Sup720, or Sup32.
- If two subinterfaces have bandwidth remaining ratios that vary greatly (for example, 1000 to 1), you must configure a low queue limit (between 2 and 50) for the child default class of the subinterface with the lower ratio. Without a low queue limit, the packets that are buffered due to the default queue-limit value are allowed to pass after traffic is stopped, which affects bandwidth remaining ratios significantly. Configuring a low queue limit ensures that the ratios are maintained even after the traffic is stopped.

Note We recommend that you use BRR with priority-rate propagation. See the "Priority-Rate Propagation Recommendations" section on page 22-25 for more information.

BRR Configuration Instructions

Following is a summary of the steps required to configure a QoS policy that defines BRR for a subscriber (access) interface on the Cisco 7600 SIP-400. The table provides detailed instructions.

Note The command lines include only those arguments and keywords required to configure BRR.

1. **enable**
2. **configure terminal**
3. **qos scheduler priority-rate-propagation platform sip-400** (optional but recommended)
4. **policy-map** *child-policy-name*
5. **class** *class-map-name*
6. **priority level** *level* (optional but recommended)
7. **police** *bps*
8. **exit**
9. **exit**
10. **policy-map** *parent-policy-name*
11. **class class-default**
12. **bandwidth remaining ratio** *ratio*
13. **shape average** *cir* [*bc*] [*be*]
14. **service-policy** *child-policy-name*
15. **exit**
16. **exit**
17. **interface** *type slot/module/port.subinterface* **access**
18. **service-policy output** *parent-policy-name*

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**qos scheduler priority-rate-propagation platform sip-400** **Example:** `Router(config)# qos scheduler priority-rate-propagation platform sip-400`	Enables the priority-rate propagation feature on the Cisco 7600 SIP-400. This feature applies a priority level and traffic rate for priority traffic to all higher-level queues in the queue hierarchy, even if the queues are not specifically configured for minimum rates or priority. **Note** This step is optional; however, if you are using BRR, we recommend that you perform this step.
Step 4	**policy-map** *child-policy-name* **Example:** `Router(config)# policy-map child`	Creates or modifies a child policy map and enters policy-map configuration mode. • *child-policy-name* is the name of the child policy map.
Step 5	**class** *class-map-name* **Example:** `Router(config-pmap)# class precedence_0`	Configures the class map that you specify. Enters policy-map class configuration mode. • *class-map-name* is the name of a previously created class map.
Step 6	**priority level** *level* **Example:** `Router(config-pmap-c)# priority level 1`	Assigns a priority level to this traffic class. • *level* is the priority level to assign. Valid values are: 1 (high) and 2 (low). **Note** Do not specify the same priority level for two different classes in the same policy map.
Step 7	police *bps* **Example:** `Router(config-pmap-c)# police 200000000`	(Optional) Specifies the rate at which to police traffic belonging to this traffic class. • *bps* specifies the average rate in bits per second (bps). Valid values are from 8,000 to 2,488,320,000 bps.
Step 8	**exit**	Exits policy-map class configuration mode.
Step 9	**exit**	Exits policy-map configuration mode.
Step 10	**policy-map** *parent-policy-name* **Example:** `Router(config)# policy-map Parent`	Creates or modifies a parent policy map. Enters policy-map configuration mode. • *parent-policy-name* is the name of the parent policy map.
Step 11	**class class-default** **Example:** `Router(config-pmap)# class class-default`	Configures the class-default class. Enters policy-map class configuration mode. **Note** The router interprets any features configured under the class-default class as aggregate features on the subinterface.

	Command or Action	Purpose
Step 12	**bandwidth remaining ratio** *ratio* **Example:** `Router(config-pmap-c)# bandwidth remaining ratio 10`	Specifies the bandwidth-remaining ratio for the subinterface. The scheduler allocates the excess bandwidth relative to other subinterfaces. • *ratio* is the value that is used to determine the amount of unused bandwidth to allocate to each queue on the subinterface during periods of congestion. Valid values are 1 to 1000. The default and minimum values are 1. **Note** The CLI supports a *ratio* value of 1 to 65535 but you cannot apply a policy with a BRR value above 1000 to a Cisco 7600 SIP-400 interface.
Step 13	**shape average** *cir* [*bc*] [*be*] **Example:** `Router(config-pmap-c)# shape average 100000000`	(Optional) Shapes the average rate to the rate you specify. • **average** specifies average rate shaping. • *cir* specifies the committed information rate (CIR), in bits per second (bps). • (Optional) *bc* specifies the committed burst size, in bits. • (Optional) *be* specifies the excess burst size, in bits.
Step 14	**service-policy** *child-policy-name* **Example:** `Router(config-pmap-c)# service-policy child`	Applies the specified child policy map to the default traffic class of the parent policy. The router applies the QoS actions specified in the child policy to the traffic class. • *child-policy-name* is the name of the child policy. **Note** Do not include **input** or **output** keyword when applying a child policy to a parent policy. **Note** On a subinterface, the child policy can be applied only to the parent's default traffic class.
Step 15	**exit**	Exits policy-map class configuration mode.
Step 16	**exit**	Exits policy-map configuration mode.

	Command or Action	Purpose
Step 17	**interface** *type slot/module/port.subinterface* **access** **Example:** `Router(config)# interface GigabitEthernet 1/0/0.1 access`	Creates or modifies the access subinterface you specify. Enters subinterface configuration mode. • *type* is the interface type (for example, Gigabit Ethernet). • *slot/module/port.subinterface* identifies the subinterface (for example, 1/0/0.1). • **access** identifies this as an IP subscriber interface.
Step 18	**service-policy output** *parent-policy-name* **Example:** `Router(config-subif)# service-policy output parent`	Applies the parent policy to the subinterface. • **output** applies the service policy to outbound traffic. • *parent-policy-name* is the name of the parent policy. **Note** A policy map with BRR can be used only in the egress direction. The router shapes the subinterface traffic to the shaping rate specified in the parent class-default class and applies the QoS actions specified in the child policy to traffic matching the traffic classes. During periods of congestion, the router uses the bandwidth-remaining ratio specified in the parent policy map to allocate unused bandwidth on this subinterface relative to other subinterfaces.

Priority-Rate Propagation Recommendations

Priority-rate propagation is a process where you apply (propagate) a priority level and traffic rate from a lower-level queue to all of the upper-layer queues in the queue hierarchy, even if the upper-layer queues are not specifically configured for minimum rates or priority. For example, if you configure a priority level and traffic rate for a traffic class (such as video) in a child policy, you can use priority-rate propagation to apply that rate to video traffic at all queue levels (parent queue, subinterface queue, and interface queue).

Dual-priority queues enable you to define two classes of high-priority traffic in a single policy map. You can also use the **priority level** command to assign a priority (high or low) to each priority queue. The **priority level** command specifies that a class of traffic has latency requirements with respect to other classes. Currently, the router supports two priority levels: level 1 (high) and level 2 (low). The router places traffic with a high priority level on the outbound link ahead of traffic with a low priority level. High priority packets, therefore, are not delayed behind low priority packets.

The router associates a single priority queue with each priority level and services the high level priority queues until empty before servicing the next level priority queues and non-priority queues. While the router services a queue, the service rate is as fast as possible and is constrained only by the rate of the underlying link or parent node in a hierarchy. If a rate is configured and the router determines that a traffic stream has exceeded the configured rate, the router drops the exceeding packets during periods of congestion. If the link is currently not congested, the router places the exceeding packets onto the outbound link.

If bandwidth remaining ratio (BRR) has also been configured, the router services priority traffic first. After servicing the priority traffic bandwidth, the router allocates unused bandwidth to the logical queues based on the configured bandwidth-remaining ratio. In this default case, the three-level scheduler allocates an equal share of the unused bandwidth to each logical queue.

If high priority traffic is not policed appropriately, the low priority traffic could be deprived of adequate bandwidth. Therefore, we recommend you use the **police** command to configure a policer for high priority traffic. If you configure the **police** command for priority queues, the traffic rate is policed to the police rate for each of the priority queues.

Note For information on how Cisco 7600 Series routers handle dual priority queues on SIP-600 and ES20, see Dual Priority Queues on SIP-600 and ES20.

Priority-Rate Propagation Configuration Guidelines

As you configure priority-rate propagation for use with BRR, consider the following guidelines:

- Use the [**no**] **qos scheduler priority-rate-propagation platform sip400** command in global configuration mode to enable and disable the priority-rate propagation feature.
- The [**no**] **qos scheduler priority-rate-propagation platform sip400** command has no effect on QoS policies that are already attached to interfaces. Therefore, we recommend that you issue the command before attaching QoS policies.

  Note If you issue the [**no**] **qos scheduler priority-rate-propagation platform sip400** command after attaching QoS policies to Cisco 7600 SIP-400 interfaces, you must save the configuration and reload the router for the command to take effect.

- Priority-rate propagation and BRR work together as follows:
  - When priority-rate propagation is enabled, the router services the priority bandwidth for all subinterface policies. The remaining bandwidth is then distributed according to the bandwidth remaining ratios. In this scenario, the priority rate is propagated from the child level to the interface queue.
  - When priority-rate propagation is disabled, the aggregate subinterface bandwidth (priority and best effort) is shared according to the bandwidth remaining ratios. In this scenario, the priority bandwidth is not propagated from the child queue to the interface queue.

Priority-Rate Propagation and BRR Configuration Example

This is an example of a priority level (2) assigned to video traffic in a child policy map and used with BRR, which is configured in the parent policy map:

```
policy-map parent
 class class-default
   bandwidth remaining ratio 1
   service-policy child

policy-map child
 class video
   priority level 2
   police 200 Mbps
```

Unsupported IP Subscriber Session Features

Due to the way that internal VLANs are allocated for sharing among IP subscribers, the following features are not available for individual subscribers:

- Policy-based routing (PBR), Network Address Translation (NAT), or unicast Reverse Path Forwarding (uRPF)
- IPv4 and IPv6 multicast
- Encoded address resolution logic (EARL) features, such as reflexive ACL, Generic Route Encapsulation (GRE) tunneling, Context-Based Access Control (CBAC), and server load balancing (SLB)

IP Subscriber Awareness over Ethernet Configuration Guidelines

Note The IP Subscriber Awareness over Ethernet feature is not available in the IP services software image (*xxx*-ipservices_wan-mz). Although the image shows the **access** keyword as being available for the **interface** command, the subscriber awareness functionality is not available.

Observe the following guidelines and limitations to configure IP Subscriber Awareness over Ethernet on Cisco 7600 routers:

- Software and hardware requirements:
  - Cisco IOS Release 12.2SRB or later
  - RSP720 with PFC3C or PFC3CXL (other supervisor engines are not supported)
  - Cisco 7600 SIP-400 and 5-Port Gigabit Ethernet SPA (SPA-5X1GE-V2)
  - Support for ES+ linecards from 12.2(33)SRE onwards.
- Oversubscription is supported only on the 5-Port Gigabit Ethernet SPA.
- A maximum of 32000 interfaces are supported on the router. To support 32000 interfaces:
  - The RSP720 must have 2 GB of RP memory and 1 GB of SP memory.
  - The Cisco 7600 SIP-400 must have 1 GB of memory.
- The Cisco 7600 SIP-400 supports a maximum of 8000 IP subscribers.
- The 5-Port Gigabit Ethernet SPA (SPA-5X1GE-V2) supports up to 8000 VLANs.
- The access subinterface that represents an IP subscriber must be configured for .1Q or Q-in-Q encapsulation.
- The MTU of the access subinterface is 1500 and this value cannot be changed.
- You can convert a regular GE subinterface to an access interface, but you cannot convert an access interface to a regular GE subinterface. Instead, you must delete the access subinterface.
- EARL-based features are not supported. This includes Network Address Translation (NAT), Reflexive ACL, Generic Route Encapsulation (GRE) tunneling, Context-Based Access Control (CBAC), and server load balancing (SLB).
- We recommend that you do not configure Hot Standby Routing Protocol (HSRP) for link redundancy.
- See the "QoS Recommendations" section on page 22-20 for QoS guidelines.

Interaction with Other Features

This section list describes the interaction between IP Subscriber Awareness over Ethernet and other features that are configured on the router.

Multicast traffic is not affected by the feature. The router can participate in IGMP functions and replication without being affected by IP Subscriber Awareness over Ethernet. In addition, the router supports multicast traffic without the authentication of data service. This allows basic video service to be provided without data service.

The DSLAM (not the router) is responsible for replicating multicast traffic and delivering it to IP subscribers. Therefore, it is not necessary for the IP Subscriber Awareness over Ethernet feature to support multicast traffic on IP subscriber interfaces (access interfaces).

Configuring IP Subscriber Awareness over Ethernet

This section provides information about configuring the IP Subscriber Awareness over Ethernet feature on a Cisco 7600 series router:

Configuration Summary

This is a summary of steps required to configure IP Subscriber Awareness over Ethernet. Detailed configuration instructions are provided in the next section.

Before Starting

- Determine which VPN routing and forwarding (VRF) table each IP subscriber should be part of. All of the subscribers in a VRF share a single internal VLAN for data services. Use the **ip vrf** and **rd** commands to create each of the VRF tables that you need.

  To use the same VRF, all the subscribers must belong to the same network service provider (NSP), Internet service provider (ISP), or access service provider (ASP). If you do not assign a subscriber to a VRF, the subscriber is added to the default VRF, which the router creates during system bootup.

- Make sure that the router is configured as a DHCP server or a DHCP relay device in order to allow IP addresses to be dynamically assigned for IP subscriber sessions. Otherwise, you would have to assign a static IP address to each IP subscriber access subinterface (which is not recommended).

  For information about configuring DHCP, see "Configuring DHCP" in the *Cisco IOS IP Configuration Guide* at:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcprt1/1cfdhcp.htm

- Determine which physical interfaces are used by IP subscribers. For each IP subscriber, configure an access subinterface on the physical interface that the subscriber is connected to.

Configure QoS and HQoS Policies for IP Subscribers

- Define QoS policies (class maps and policy maps) to define traffic bandwidth and shaping policies for subscriber traffic. You can use a hierarchical QoS (HQoS) policy to shape traffic at different levels. For example, the parent policy could define the total bandwidth for the subscriber, and the child policy could define the bandwidth for different types of subscriber traffic (such as video). On a subinterface, the child policy can be attached only for the default class of the parent.
- (Optional) You can create dual-priority queues to handle the subscriber's voice and video traffic.
- You can also define a class-based weighted fair queue (CBWFQ) or priority queue (PQ) for different types of subscriber traffic.

Configure Access Lists and Security ACLs

- Determine the security policies needed for IP subscribers. Create access lists and security ACLs to define these policies.

  Here is an example of two access lists (2 and 3) that will be applied to IP subscribers:

```
access-list 2 permit 18.18.18.18
access-list 3 permit 23.23.23.23
access-list 101 deny   ip 44.1.1.0 0.0.0.255 any
access-list 101 permit icmp any any
```

  This example configures an input and output security ACL for the IP subscriber session that is represented by the access subinterface gig0/1/1.100:

```
interface gig0/1/1.100 access
  encapsulation dot1q 100
  ip address 10.10.10.1 255.255.255.0
  ip access-group 101 in
  ip access-group 102 out
```

Configure IP Subscriber Interfaces

- Create an access interface for each IP subscriber. Create the access interface as a subinterface of the subscriber's physical interface. For example, if the subscriber is connected to Gig1/0/0, you could configure the access interface as Gig1/0/0.100.
- Configure the access interface as follows:
  - (Optional) Assign a static IP address to the interface. We recommend that you do not configure many access interfaces with a static IP address. Instead, you should allow DHCP to dynamically assign IP addresses for IP subscriber sessions.
  - If the IP subscriber belongs to a particular VRF table, include the **ip vrf forwarding** *vrf-name* command in the configuration to associate the interface with the table. If you do not specify a VRF table, the subscriber is added to the default VRF.
  - Set the encapsulation type (.1Q or Q-in-Q) and specify which VLAN the interface is part of.
  - Attach QoS policies to the interface to define traffic bandwidth and shaping policies for the subscriber traffic.

  This example shows two IP subscriber access interfaces (gig1/0/0.100 and gig1/0/0.300). Since the subscribers connect through Gig1/0/0, the access interfaces are created as subinterfaces of Gig1/0/0. Notice that gig1/0/0.100 is assigned a static IP address and gig1/0/0.300 uses DHCP to obtain an IP address. In addition, notice that gig1/0/0.300 is VRF aware.

```
interface gig1/0/0.100 access
 ip address 10.10.10.10 255.255.255.255
 encapsulation dot1q 100
 service-policy input bband-in1
 service-policy output bband-out1
```

```
interface gig1/0/0.300 access
 ip vrf forwarding vrf1
 encapsulation dot1q 300
 service-policy input bband-in1
 service-policy output bband-out1
```

Verify the IP Subscriber Awareness over Ethernet Feature

Use the **show running-config interface** command to verify the status of each access interface that represents an IP subscriber. An access subinterface should exist for each subscriber and the interfaces should be in the Up state.

- Issue the **show running-config interface** *interface***.***subinterface* command to verify the configuration of each access subinterface (where *interface* is the physical interface and **.***subinterface* is the access subinterface). For example, **show running-config interface Gig1/0/2.1** displays the access subinterface (.1) that exists on the physical interface Gig1/0/2.

Configuration Examples

This example shows a configuration with three subscribers (Gig3/2/0.10, Gig3/2/0.11, and Gig3/2/0.12), each receiving a different type of service: gold (30 Mbps), silver (15 Mbps), and bronze (5 Mbps). Each subscriber has per-subscriber accounting and per-subscriber ACL configured.

The QoS policy maps are configured so that video traffic is never dropped, and default traffic is shared in the ratio of 30:15:5 (which results in a bandwidth remaining ratio of 6:3:1).

```
aaa new-model
aaa accounting network defaul start group radius
radius-server key cisco
radius-server host 2.2.2.2
int loopback 1
ip address 13.0.7.254 255.255.248.0

ip dhcp pool Loopback1
  network 13.0.0.0 255.255.248.0

Class-map voip
 match ip precedence 5

Class-map video
 match ip precedence 6

policy-map data_gold_child_out
 class video
   priority level 2
   police 27000000
   set cos 5
 class class-default
   police 30000000
   set cos 3

policy-map data_gold_parent_out
 class class-default
   shape average 29900000
   bandwidth remaining ratio 6
   service-policy data_gold_child_out

policy-map data_silver_child_out
 class video
   priority level 2
```

```
   police 27000000
   set cos 5
 class class-default
   police 15000000
   set cos 2

policy-map data_silver_parent_out
 class class-default
   shape average 29900000
   bandwidth remaining ratio 3
   service-policy data_silver_child_out

policy-map data_bronze_child_out
 class video
   priority level 2
   police 27000000
   set cos 5
 class class-default
   police 5000000
   set cos 1

access-list 102 permit ip any any precedence 5
access-list 102 permit ip any any precedence 2
access-list 102 permit ip any any precedence 0

policy-map data_bronze_parent_out
 class class-default
   shape average 29900000
   bandwidth remaining ratio 1
   service-policy data_bronze_child_out

policy-map data_gold_in
  class class-default
    police 5000000
policy-map data_silver_in
  class class-default
    police 2000000
policy-map data_bronze_in
  class class-default
    police 2000000


interface gig 3/2/0.10 access
    ip unnumbered Loopback 1
    encapsulation dot1q 10
    service-policy output data_gold_parent_out
    service-policy input data_gold_in
    accounting dhcp source-ip aaa list default
    ip access-group 103 in

interface gig 3/2/0.11 access
    ip unnumbered Loopback 1
    encapsulation dot1q 11
    service-policy output data_silver_parent_out
    service-policy input data_silver_in
    accounting dhcp source-ip aaa list default
    ip access-group 103 in

interface gig 3/2/0.12 access
    ip unnumbered Loopback 1
    encapsulation dot1q 12
    service-policy output data_bronze_parent_out
    service-policy input data_bronze_in
    accounting dhcp source-ip aaa list default
```

```
ip access-group 103 in
```

Command Reference

This section describes the new commands for IP Subscriber Awareness over Ethernet. This new command is introduced as part of this feature:

- interface access

interface access

To create an access interface for an IP subscriber, use the **interface access** command in global configuration mode. Use the **no** form of the command to delete an IP subscriber access interface.

interface *interface***.***subinterface* **access**

no interface *interface***.***subinterface* **access**

Syntax Description

interface	Identifies the physical interface that this IP subscriber is connected to.
.*subinterface*	A subinterface number to assign to the access interface.

Defaults

None

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Usage Guidelines

This command creates an access interface for an IP subscriber. Create the access interface as a subinterface of the physical interface that the subscriber is connected to. For example, if the subscriber is connected to Gig1/0/0, configure the access interface as Gig1/0/0.1, Gig1/0/0.2, Gig1/0/0.3, and so on.

Include the **ip vrf forwarding** *vrf-name* command in the configuration to associate the IP subscriber with the specified VRF table. If you do not specify a VRF table, the subscriber is added to the default VRF table (which is created during router bootup).

Examples

This command example creates an access interface for an IP subscriber and assigns the subscriber to the VRF table named vrf1. The access interface is created as subinterface .300 on the physical interface Gig2/0/1. Use additional commands to complete the configuration (to specify encapsulation type, and to assign QoS policies).

```
Router(config)# interface Gig2/0/1.300 access
Router(config-if)# ip vrf forwarding vrf1
Router(config-if)#
```

CHAPTER 23

Configuring UDE and UDLR

This chapter describes how to configure unidirectional Ethernet (UDE) and unidirectional link routing (UDLR) on the Cisco 7600 series router.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter contains the following sections:

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 Cisco 7600 series routers, 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

On Cisco 7600 series routers, 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 23-2 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, which are less expensive than bidirectional transceivers. Cisco 7600 series routers support the following 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 23-3
- Configuring UDLR, page 23-6

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 23-4
- Configuring Hardware-Based UDE, page 23-5
- Configuring Software-Based UDE, page 23-5

UDE Configuration Guidelines

When configuring UDE, follow these guidelines:

- UDE is supported on the Supervisor Engine 720.
- STP cannot prevent Layer 2 loops in topologies that include unidirectional links.
- Send-only ports always transition to the STP forwarding state, because send-only ports never receive BPDUs.
- Receive-only ports cannot send BPDUs.
- Unidirectional ports do not support any features or protocols that require negotiation with the port at the other end of the link, including these:
  - Speed and duplex mode autonegotiation
  - Link negotiation
  - IEEE 802.3Z flow control
  - Dynamic trunking protocol (DTP)

  You must manually configure the parameters that are typically controlled by Layer 2 protocols.
- A topology that includes unidirectional links only supports the VLAN Trunking Protocol (VTP) when the VTP server can send VTP frames to all routers in the VTP domain.
- Disable VTP pruning on routers 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 router 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.
- Prior to 12.2(33) SRD4 release, when you configure SPAN and UDLD combination on a port where the interface is the span destination port, the current operational state of the UDLD peer is disabled as if the UDLD is disabled at the local end. Post 12.2(33) SRD4 release, if the interface is set as the SPAN destination, the current operational state of the UDLD peer is displayed as **Advertisement** instead of **Disabled**.
- IGMP snooping does not support topologies where there are unidirectional links between the router 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.

- During OIR, unless the line card comes online, you should not attempt to delete the UDE configuration or you will lose UDLD capability on the port.

  In case you have deleted the UDE configuration while the card is still offline and you want to enable UDLD again, you should:

  – Reload the router, or

  – Configure UDE on that port again and unconfigure UDE only when the line card is online.

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

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

- The PFC3 does not provide hardware support for UDLR back-channel tunnels. The MSFC3 and MSFC4 (RSP720) support 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 24

Configuring Multiprotocol Label Switching on the PFC

This chapter describes how to configure Multiprotocol Label Switching (MPLS) on the Cisco 7600 PFC card. The information in this chapter describes MPLS operation on the PFC3B, PFC3BXL, PFC3C, and PFC3CXL cards. Unless otherwise noted, MPLS operation is the same on all of these PFC cards.

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

- The Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter contains these sections:

PFC MPLS Label Switching

These sections describe MPLS label switching:

Understanding MPLS

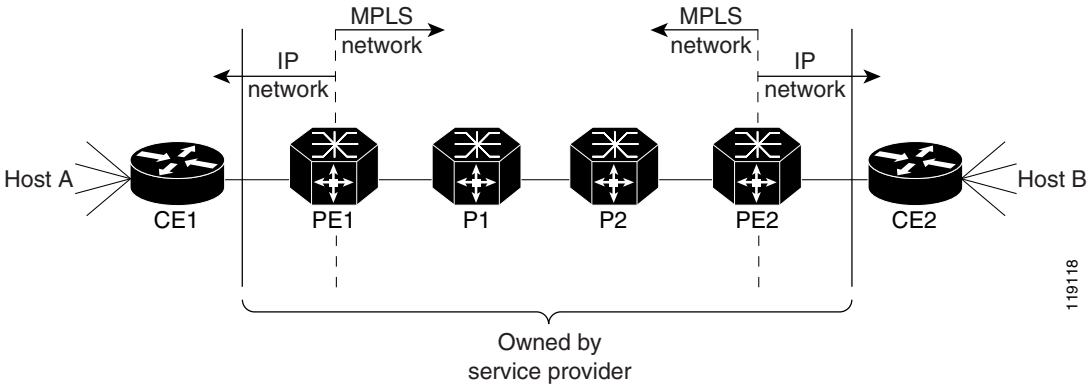
MPLS uses label switching to forward packets over various link-level technologies such as Packet-over-SONET (POS), Frame Relay, ATM, and Ethernet. Labels are assigned to packets based on groupings or forwarding equivalence classes (FECs). The label is added between the Layer 2 and the Layer 3 header.

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 24-1 shows an MPLS network of a service provider that connects two sites of a customer network.

Figure 24-1 MPLS Network



For additional information on MPLS, see this publication:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/xcftagov.htm

Understanding MPLS Label Switching

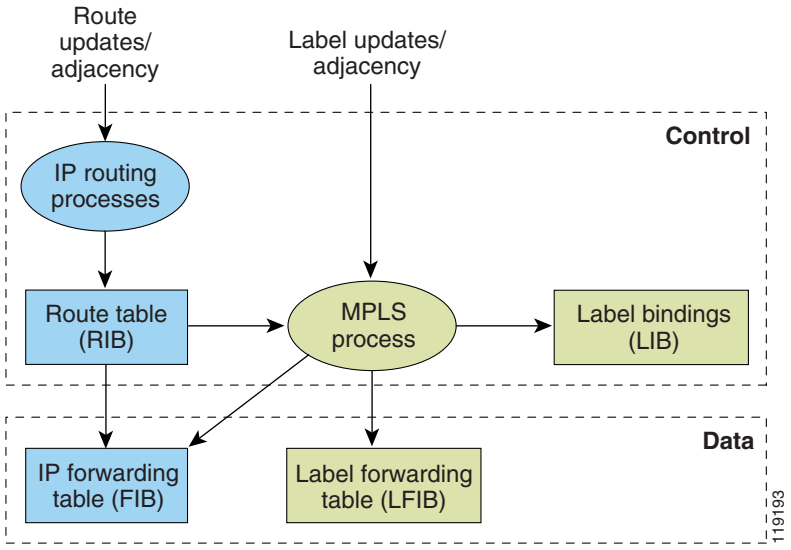
The PFC supports Layer 3 Multiprotocol Label Switching (MPLS) virtual private networks (VPNs), and Layer 2 Ethernet over MPLS (EoMPLS), with quality of service (QoS) and security.

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

The PFC receives the information and creates its own copies of the FIB and LFIB tables. Together, these tables comprise the FIB TCAM. The DFC looks up incoming IP packets and labeled packets against the FIB TCAM table. The lookup result is the pointer to a particular adjacency entry. It is the adjacency entry that contains appropriate information for label pushing (for IP to MPLS path), label swapping (for MPLS to MPLS path), label popping (for MPLS to IP path), and encapsulation.

Figure 24-2 shows the various functional blocks on the PFC 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 24-2 MPLS Forwarding, Control and Data Planes



IP to MPLS

At the ingress to the MPLS network, the PFC examines the IP packets and performs a route lookup in the FIB TCAM. The lookup result is the pointer to a particular adjacency entry. The adjacency entry contains the appropriate information for label pushing (for IP to MPLS path) and encapsulation. The PFC generates a result containing the imposition label(s) needed to switch the MPLS packet.

Note If MPLS load sharing is configured, the adjacency may point to a load-balanced path. See "Basic MPLS Load Balancing" section on page 24-8.

MPLS to MPLS

At the core of an MPLS network, the PFC uses the topmost label to perform a lookup in the FIB TCAM. The successful lookup points to an adjacency that swaps the top label in the packet with a new label as advertised by the downstream label switch router (LSR). If the router is the penultimate hop LSR router (the upstream LSR next to the egress LER), the adjacency instructs the PFCBXL or PFC3CXL to pop the topmost label, resulting in either an MPLS packet with the remaining label for any VPN or AToM use or a native IP packet.

MPLS to IP

At the egress of the MPLS network there are several possibilities.

For a native IP packet (when the penultimate router has popped the label), the PFC performs a route lookup in the FIB TCAM.

For a MPLS VPN packet, after the Interior Gateway Protocol (IGP) label is popped at penultimate router, the VPN label remains. The operation that the PFC performs depends on the VPN label type. Packets carrying aggregate labels require a second lookup based on the IP header after popping the aggregate label. For a nonaggregate label, the PFC performs a route lookup in the FIB TCAM to obtain the IP next hop information.

For the case of a packet with an IGP label and a VPN label, when there is no penultimate hop popping (PHP), the packet carries the explicit-null label on top of the VPN label. The PFC looks up the top label in the FIB TCAM and recirculates the packet. Then the PFC handles the remaining label as described in the preceding paragraph, depending on whether it is an aggregate or nonaggregate label.

Packets with the explicit-null label for the cases of EoMPLS, MPLS, and MPLS VPN an MPLS are handled the same way.

MPLS VPN Forwarding

There are two types of VPN labels: aggregate labels for directly connected network or aggregate routes, and nonaggregate labels. Packets carrying aggregate labels require a second lookup based on the IP header after popping the aggregate label. The VPN information (VPN-IPv4 address, extended community, and label) is distributed through the Multiprotocol-Border Gateway Protocol (MP-BGP).

Recirculation

In certain cases, the PFC provides the capability to recirculate the packets. Recirculation can be used to perform additional lookups in the ACL or QoS TCAMs, the Netflow table, or the FIB TCAM table. Recirculation is necessary in these situations:

- To push more than three labels on imposition
- To pop more than two labels on disposition
- To pop an explicit null top label
- When the VPN Routing and Forwarding (VRF) number is more than 511
- For IP ACL on the egress interface (for nonaggregate (per-prefix) labels only)

Packet recirculation occurs only on a particular packet flow; other packet flows are not affected.The rewrite of the packet occurs on the modules; the packets are then forwarded back to the PFC for additional processing.

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 PFC provides the capability to recirculate the packets. See the "Recirculation" section on page 24-4.
- 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 on the PFC:

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.

- 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/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120st/120st21/fs_rtr.htm
- 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/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/xcftagc.htm

  http://www.cisco.com/warp/public/105/mplsteisis.html

  http://www.cisco.com/warp/public/105/mpls_te_ospf.html

- 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/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s18/fsdserv3.htm

- 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/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s18/fstefa_3.htm

- 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/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s18/fsiarea3.htm

- 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/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120st/120st21/fs_vpn.htm

- 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/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/ftcsc8.htm

- 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/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftcscl13.htm

- 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/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s24/fsias24.htm

- 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/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftiasl13.htm

- 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/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dt_hsmp.htm

- 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/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/ospfshmk.htm

- Any Transport over MPLS (AToM)—Transports Layer 2 packets over an MPLS backbone. See the "Any Transport over MPLS" section on page 24-13.

MPLS Guidelines and Restrictions

When configuring MPLS on the PFC follow these guidelines and restrictions:

- The PFC supports up to 15 load-shared paths. Cisco IOS releases for other platforms support only 8 load-shared paths.
- The PFC supports MTU checking and fragmentation.
- Fragmentation is supported with software (for IP to MPLS path). See the **mtu** command description in the *Cisco 7600 Series Router Cisco IOS Command Reference*.
- Observe the following maximum transmission unit (MTU) guidelines when you configure MPLS:
  - Both ends of the MPLS link must have the same MTU size; otherwise, MPLS detects a mismatch between the interfaces and it never becomes operational.

    Note that MPLS over RBE allows different MTU sizes (for example, default Gigabit Ethernet and ATM). However, when running OSPF over RBE, you must include the **ip ospf mtu-ignore** command on the ATM interface; otherwise, OSPF detects a mismatch and never becomes active.
  - The MPLS MTU size must be less than the MTU size of the physical interface that the MPLS link uses. Otherwise, problems can occur and MPLS packets might be dropped.

    Although not recommended, you can use the **mpls mtu override** *bytes* command to set the MPLS MTU size to a value greater than the interface MTU size (where *bytes* specifies MPLS MTU size).

    The **mpls mtu override** *bytes* command is available only on interfaces with a default MTU size of 1580 bytes or less (for example, Ethernet). It is not available on ATM bridged interfaces.
- For information on other restrictions, see the "MPLS VPN Guidelines and Restrictions" section on page 24-11 and the "EoMPLS Guidelines and Restrictions" section on page 24-14.

MPLS Supported Commands

MPLS on the PFC 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/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_r/index.htm

Configuring MPLS

For information about configuring MPLS, see the *Multiprotocol Label Switching on Cisco Routers* publication at the following URL:

http://preview.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcftagc_external_docbase_0900e4b180753c33_4container_external_docbase_0900e4b180754411.html

MPLS Per-Label Load Balancing

The following sections provide information on basic MPLS, MLPS Layer 2 VPN, and MPLS Layer 3 VPN load balancing.

Basic MPLS Load Balancing

The maximum number of load balancing paths is 8. The PFC forwards MPLS labeled packets without explicit configuration. If the packet has three labels or less and the underlying packet is IPv4, then the PFC uses the source and destination IPv4 address. If the underlying packet is not IPv4 or more than three labels are present, the PFC parses down as deep as the fifth or lowest label and uses it for hashing.

MPLS Layer 2 VPN Load Balancing

Load balancing is based on the VC label in the MPLS core if the first nibble of the MAC address in the customer Ethernet frame is not 4.

Note Load balancing is not supported at the ingress PE for Layer 2 VPNs. Load balancing is done based on the VC label and it is pre-selected.

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 24-8.

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 on the PFC

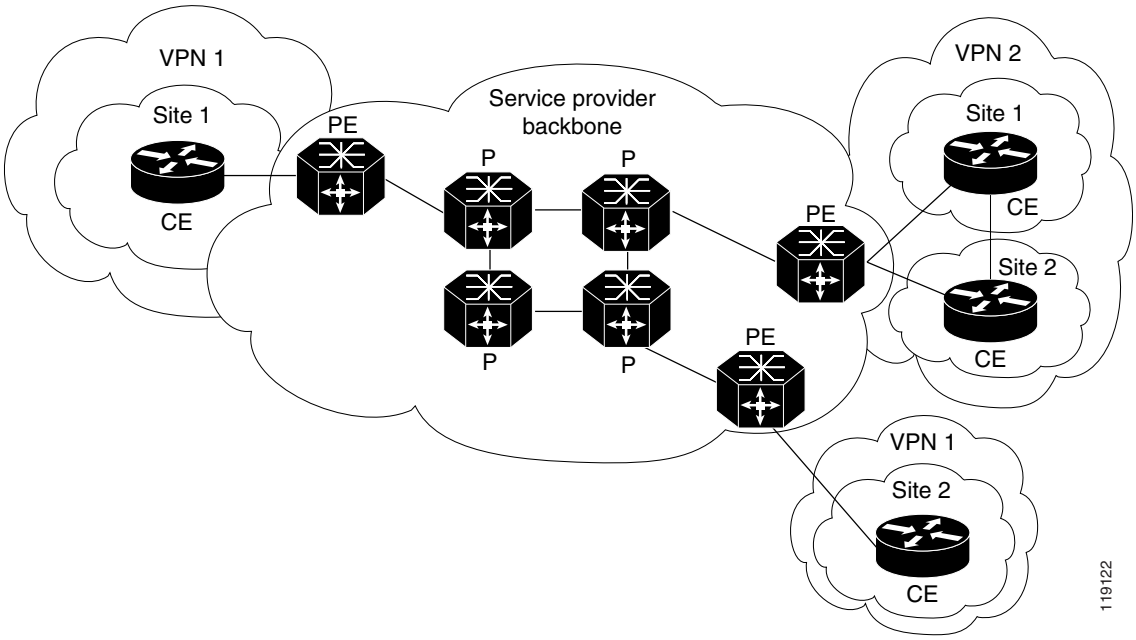
These sections describe VPN switching on the PFC:

VPN Switching Operation on the PFC

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.

A typical MPLS VPN network topology is shown in Figure 24-3.

Figure 24-3 VPNs with Service Provider Backbone



At the ingress PE, the PFC makes a forwarding decision based on the packet headers. The PFC contains a table that maps VLANs to VPNs. In the Cisco 7600 series router architecture, all physical ingress interfaces in the system are associated with a specific VPN. The PFC looks up the IP destination address in the CEF table but only against prefixes that are in the specific VPN. (The table entry points to a specific set of adjacencies and one is chosen as part of the load-balancing decision if multiple parallel paths exist.)

The table entry contains the information on the Layer 2 header that the packet needs, as well as the specific MPLS labels to be pushed onto the frame. The information to rewrite the packet goes back to the ingress 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 PFC allocates only one aggregate label per VRF.

If aggregate labels are used for disposition in an egress PE, many prefixes on the multiple interfaces may be associated with the label. In this case, the PFC must perform an IP lookup to determine the final destination. The IP lookup may require recirculation.

MPLS VPN Guidelines and Restrictions

When configuring MPLS VPN, follow these guidelines and restrictions:

- The PFC supports a total of 1024 VRFs per chassis a with enhanced OSMs. Using a nonenhanced OSM causes the system to default to 511 VRFs.
- The PFC recirculates VPNs when the number of VPNs is over 511.

MPLS VPN Supported Commands

The PFC supports these MPLS VPN commands:

- **address-family**
- **exit-address-family**
- **import map**
- **ip route vrf**
- **ip route forwarding**
- **ip vrf**
- **neighbor activate**
- **rd**
- **route-target**

Configuring MPLS VPN

For information on configuring MPLS VPN, refer to the *MPLS Virtual Private Networks* feature module at this URL:

http://preview.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcftagc_external_docbase_0900e4b180753c33_4container_external_docbase_0900e4b180754411.html#63744

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 Enhanced FlexWAN CE-facing interfaces. The PFC 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 on the PFC:

- 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
- PPP over MPLS
- HDLC over MPLS
- Circuit Emulation (TDM) over MPLS

Note Additional AToM types are planned in future releases.

The PFC supports hardware-based EoMPLS and OSM- or Enhanced FlexWAN-based EoMPLS. (Note that Release 12.2SR does not support FlexWAN-based EoMPLS). For more information, see:

http://preview.cisco.com/en/US/docs/general/TD_Trash/lczaplys_trash/mpls.html#wp1128955

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

http://preview.cisco.com/en/US/docs/general/TD_Trash/lczaplys_trash/mpls.html#wp1279824

These sections describe AToM:

AToM Load Balancing

EoMPLS on the PFC does not support load balancing at the tunnel ingress; only one Interior Gateway Protocol (IGP) path is pre-selected based on the VC label.

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 or Enhanced FlexWAN-based EoMPLS when you want local Layer 2 switching and EoMPLS on the same VLAN. You must configure EoMPLS on the SVI, and the core-facing card must be an OSM or an Enhanced FlexWAN module. When local Layer 2 switching is not required, use PFC-based EoMPLS configured on the subinterface or physical interface.

EoMPLS Guidelines and Restrictions

When configuring EoMPLS, consider 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.
- For VLAN-based EoMPLS, the MTU size on the VLAN subinterface must be greater than 1500 (the default) if a larger MTU size is specified on the physical interface.

Note Port-channel and xconnect combinations are supported on Port-based EoMPLS. However, all the restrictions for normal PFC based EoMPLS are applicable to port-channel and xconnect as well.

- 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. A new command allows you to configure the PFC 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 44, "Configuring PFC QoS."

Note Do not use the **no mls qos rewrite ip dscp** command if you have PFC-based EoMPLS and PXF-based EoMPLS services in the same system.

- EoMPLS is not supported with private VLANs.
- The following restrictions apply to using trunks with EoMPLS:
  - To support Ethernet spanning tree bridge protocol data units (BPDUs) across an EoMPLS cloud, you must disable 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 router. 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. For more information on Scalable EoMPLS (SVI-based EoMPLS) and Port-mode EoMPLS and its sample configuration, see Scalable EoMPLS and Port-mode EoMPLS, page 24-16 and Sample Configuration for SwEoMPLS and VPLS, page 24-16.
- Cisco 7600 provides three different flavors of the Ethernet over MPLS (EoMPLS) solutions.
  - PFC-based EoMPLS, also known as Hardware-based EoMPLS where the Earl imposes on the Supervisor or DFC based line card
  - LAN-based EoMPLS, also known as Software-based EoMPLS, where Earl imposes on the MPLS Core-facing line card
  - Scalable EoMPLS, where the Earl imposes on customer device facing line card. The feature is supported in the SIP400, ES20, and ES40 as customer-facing line cards. Further, in ES20 and ES40 the solution is supported only in EVC-based configuration.
- On the PFC, all protocols (for example, CDP, VTP, BPDUs) are tunneled across the MPLS cloud without conditions.
- ISL encapsulation is not supported for the interface that receives EoMPLS packets.
- Unique VLANs are required across interfaces. You cannot use the same VLAN ID on different interfaces.
- EoMPLS tunnel destination route in the routing table and the CEF table must be a /32 address (host address where the mask is 255.255.255.255) to ensure that there is a label-switched path (LSP) from PE to PE.
- For a particular EoMPLS connection, both the ingress EoMPLS interface on the ingress PE and the egress EoMPLS interface on the egress PE have to be subinterfaces with dot1Q encapsulation or neither is a subinterface.
- 802.1Q in 802.1Q over EoMPLS is supported if the outgoing interface connecting to MPLS network is a port on an Layer 2 card.

- Shaping EoMPLS traffic is not supported if the egress interface connecting to an MPLS network is a Layer 2 LAN port (a mode known as PFC-based EoMPLS).
- EoMPLS based on a PFC does not perform any Layer 2 lookup to determine if the destination MAC address resides on the local or remote segment and does not perform any Layer 2 address learning (as traditional LAN bridging does). This functionality (local switching) is available only when using 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, subinterfaces and EVC.
- Some of the SPA-based Ethernet line cards like the ES20 support matching the outer VLAN for QinQ traffic. See the documentation for the line card you are interested in for more information.

Scalable EoMPLS and Port-mode EoMPLS

In a scalable EoMPLS scenario, you can configure cross-connect directly on the EVC on the PE routers. In a port-mode EoMPLS, you can configure a cross-connect on the physical interface or subinterface. In case of scalable and port-mode EoMPLS, it is not required to disable spanning tree, since the access facing interfaces do not participate in spanning tree protocol (STP). For more information on configuring the Spanning Tree Protocol (STP) and Multiple Spanning Tree (MST) protocol on Cisco 7600 series routers, refer Chapter 19, “Configuring STP and MST”.

The next section describes sample configurations and scenarios to handle STP and allow the BPDUs to relay through the pseudowire.

Note Cisco 7600 series routers do not support multiple backup PWs.

Sample Configuration for SwEoMPLS and VPLS

Also termed as SVI-based EoMPLS, the following example outlines a sample topology for SwEoMPLS:

CE1-----PE1-----P------PE2----CE2

In a SwEoMPLS, you configure cross-connect on a SVI interface (interface VLAN). The following is a sample configuration on the CE facing interface:

```
interface FastEthernet1/13
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 110
switchport mode trunk
end
```

The following is a sample configuration for the SVI interface with cross-connect.

```
interface Vlan110
no ip address
```

```
xconnect 6.6.6.6 200 encapsulation mpls
end
```

Following sample shows a configuration on a core facing line card towards a P router:

```
interface GigabitEthernet2/2/0
ip address 53.53.53.1 255.255.255.0
mpls ip
end
```

Following sample shows a configuration on a CE facing line card for VPLS:

```
interface FastEthernet1/13
switchport switchport trunk encapsulation dot1q
switchport trunk allowed vlan 110
switchport mode trunk
end
```

Following sample shows a configuration when a cross-connect is configured within SVI:

```
interface Vlan110
description VPLS
no ip address
xconnect vfi PE1-VPLS
end
```

Based on the previous sample configurations, the VFI definitions are defined:

```
l2 vfi PE1-VPLS manual
vpn id 110
neighbor 6.6.6.6 encapsulation mpls
```

Following sample shows a configuration on a CE facing line card for VFI:

```
interface GigabitEthernet2/2/0
ip address 53.53.53.1 255.255.255.0
mpls ip
end
```

In the topologies and configurations listed previously:

- The customer routers CE1 and CE2 possess ethernet connectivity.
- Relays traffic tagged with any VLANS
- A EoMPLS pseudo wire is created between routers PE1 and PE2 to allow CE1-CE2 traffic transparently through PE1-PE2.

Managing Spanning Tree Protocol to allow Bridge Protocol Data Units

The Customer facing interfaces on the PE routers participate in the STP. To support Ethernet spanning tree bridge protocol data units (BPDUs) across an EoMPLS cloud (PE1-P-PE2), modify the methods listed below to disable the supervisor engine spanning tree:

1. If spanning tree mode is MST, then STP BPDUs are untagged.

```
spanning-tree mode mst
```

On the CE facing interface on a PE router:

```
Int Gig 1/1
switchport
switchport trunk allowed vlan 110
switchport mode trunk
```

Configure a VFI (mst-1 here) to relay the STP BPDUs.

```
l2 vfi mst-1 manual
vpn id 1
forward permit l2protocol all
```

Attach the VFI configured in the previous step to SVI.

```
interface Vlan1
no ip address
xconnect vfi mst-1
```

2. If spanning tree mode is PVST, STP BPDUs are tagged. For example, if the customer router's traffic is expected on VLAN110, then the BPDU's are tagged with VLAN 110.

```
spanning-tree mode pvst
```

On the access facing interface:

```
Int Gig1/1
switchport
switchport trunk allowed vlan 110
switchport mode trunk
no spanning-tree vlan 110
```

In the above scenario, **no spanning-tree vlan 110** is sufficient and a special VFI is not needed to relay BPDUs.

Configuring EoMPLS

These sections describe how to configure EoMPLS:

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 on the PFC:

- 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 on the PFC does not allow local switching of packets between interfaces unless you use loopback ports.
- A system can have both an OSM or Enhanced FlexWAN configuration and PFC-mode 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 Enhanced FlexWAN-enabled interfaces, OSM or Enhanced FlexWAN-based EoMPLS connections will not be active; this causes packets for OSM or Enhanced FlexWAN-based EoMPLS arriving on non-WAN interfaces to be dropped.

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

With a PFC, you can configure an OSM or an Enhanced FlexWAN module to face the core of MPLS network and use either the OSM configuration, the Enhanced FlexWAN configuration, or the PFC-mode configuration.

For more information on EoMPLS over WAN (Enhanced FlexWAN and OSM), see the following publication. (Note that Release 12.2SR does not support FlexWAN-based EoMPLS).

http://preview.cisco.com/en/US/docs/general/TD_Trash/lczaplys_trash/mpls.html#wp1128955

Configuring PFC-Mode VLAN-Based EoMPLS

When configuring VLAN-based EoMPLS on the PFC, 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. In addition, the MTU size on the VLAN subinterface must be greater than 1500 (the default) if a larger MTU size is specified on the physical interface.

To configure VLAN-based EoMPLS on the PFC, 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.

Here is a sample of a VLAN-based EoMPLS configuration on the PFC:

```
!
interface GigabitEthernet6/4
xconnect 13.13.13.13 4 encapsulation mpls
no shut
!
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 on the PFC

When configuring port-based EoMPLS on the PFC, 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 on the PFC, configure port-based EoMPLS by performing this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface gigabitethernet`*slot/interface*	Specifies the Gigabit Ethernet interface. Make sure that the interface on the adjoining CE router is on the same VLAN as this PE router.
Step 3	`Router(config-if)# xconnect` *peer_router_id vcid* `encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

The following is an example of a port-based configuration:

```
!
EoMPLS:

router# show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Fa8/48         Ethernet             75.0.78.1       1          UP
Gi7/11.2000    Eth VLAN 2000        75.0.78.1       2000       UP


Port-Based EoMPLS Config:

router# show run interface f8/48
Building configuration...

Current configuration : 86 bytes
!
interface FastEthernet8/48
 no ip address
 xconnect 75.0.78.1 1 encapsulation mpls
end

Sub-Interface Based Mode:
router# show run interface g7/11
Building configuration...

Current configuration : 118 bytes
!
interface GigabitEthernet7/11
 description Traffic-Generator
 no ip address
 logging event link-status
 speed nonegotiate
end

router# show run int g7/11.2000
Building configuration...

Current configuration : 112 bytes
!
interface GigabitEthernet7/11.2000
 encapsulation dot1Q 2000
 xconnect 75.0.78.1 2000 encapsulation mpls
end

kb7606# show mpls l2transport vc 1 detail
Local interface: Gi7/47 up, line protocol up, Ethernet up
  Destination address: 75.0.80.1, VC ID: 1, VC status: up
    Tunnel label: 5704, next hop 75.0.83.1
    Output interface: Te8/3, imposed label stack {5704 10038}
  Create time: 00:30:33, last status change time: 00:00:43
  Signaling protocol: LDP, peer 75.0.80.1:0 up
    MPLS VC labels: local 10579, remote 10038
    Group ID: local 155, remote 116
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  VC statistics:
    packet totals: receive 26, send 0
    byte totals:   receive 13546, send 0
    packet drops:  receive 0, send 0
```

To obtain the VC type:

```
kb7606# remote command switch show mpls l2transport vc 1 de

Local interface: GigabitEthernet7/47, Ethernet
  Destination address: 75.0.80.1, VC ID: 1
  VC status: receive UP, send DOWN
  VC type: receive 5, send 5
    Tunnel label: not ready, destination not in LFIB
    Output interface: unknown, imposed label stack {}
    MPLS VC label: local 10579, remote 10038
  Linecard VC statistics:
    packet totals:  receive: 0  send: 0
    byte totals:  receive: 0  send: 0
    packet drops:  receive: 0  send: 0
  Control flags:
    receive 1,  send: 31
!
```

Verifying the Configuration

To verify and display the configuration of Layer 2 VLAN transport over MPLS tunnels, perform the following:

- To display a single line for each VLAN, naming the VLAN, status, and ports, enter the **show vlan brief** command.

```
Router# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active
2    VLAN0002                         active    Gi1/4
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

- To make sure the PE router endpoints have discovered each other, enter the **show mpls ldp discovery** command. When an PE router receives an LDP Hello message from another PE router, it considers that router and the specified label space to be “discovered.”

```
Router# show mpls ldp discovery
 Local LDP Identifier:
    13.13.13.13:0
    Discovery Sources:
    Interfaces:
        GE-WAN3/3 (ldp): xmit/recv
            LDP Id: 12.12.12.12:0
    Targeted Hellos:
        13.13.13.13 -> 11.11.11.11 (ldp): active/passive, xmit/recv
            LDP Id: 11.11.11.11:0
```

- To make sure the label distribution session has been established, enter the **show mpls ldp neighbor** command. The third line of the output shows that the state of the LDP session is operational and shows that messages are being sent and received.

```
Router# show mpls ldp neighbor
    Peer LDP Ident: 12.12.12.12:0; Local LDP Ident 13.13.13.13:0
        TCP connection: 12.12.12.12.646 - 13.13.13.13.11010
        State: Oper; Msgs sent/rcvd: 1715/1706; Downstream
        Up time: 1d00h
        LDP discovery sources:
```

```
      GE-WAN3/3, Src IP addr: 34.0.0.2
    Addresses bound to peer LDP Ident:
      23.2.1.14       37.0.0.2        12.12.12.12     34.0.0.2
      99.0.0.1
Peer LDP Ident: 11.11.11.11:0; Local LDP Ident 13.13.13.13:0
    TCP connection: 11.11.11.11.646 - 13.13.13.13.11013
    State: Oper; Msgs sent/rcvd: 1724/1730; Downstream
    Up time: 1d00h
    LDP discovery sources:
      Targeted Hello 13.13.13.13 -> 11.11.11.11, active, passive
    Addresses bound to peer LDP Ident:
      11.11.11.11     37.0.0.1        23.2.1.13
```

- To make sure the label forwarding table is built correctly, enter the **show mpls forwarding-table** command.

```
Router# show mpls forwarding-table
Local  Outgoing    Prefix              Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id        switched   interface
16     Untagged    223.255.254.254/32   \
                                   0          Gi2/1      23.2.0.1
20     Untagged    l2ckt(2)           55146580   Vl2        point2point
24     Pop tag     37.0.0.0/8         0          GE3/3      34.0.0.2
25     17          11.11.11.11/32     0          GE3/3      34.0.0.2
26     Pop tag     12.12.12.12/32     0          GE3/3      34.0.0.2
```

- The output shows the following data:
  - Local tag—Label assigned by this router.
  - Outgoing tag or VC—Label assigned by next hop.
  - Prefix or Tunnel Id—Address or tunnel to which packets with this label are going.
  - Bytes tag switched— Number of bytes switched out with this incoming label.
  - Outgoing interface—Interface through which packets with this label are sent.
  - Next Hop—IP address of neighbor that assigned the outgoing label.
- To view the state of the currently routed VCs, enter the **show mpls l2transport vc** command:

```
Router# show mpls l2transport vc

Local intf     Local circuit        Dest address    VC ID      Status
-------------  -------------------- --------------- ---------- ----------
Vl2            Eth VLAN 2           11.11.11.11     2          UP
```

Configuring 7600-MUX-UNI Support on LAN Cards

A User Network Interface (UNI) is the point where the customer edge (CE) eqipment connects to the ingress PE and an attachment VLAN is a VLAN on a UNI port.

The 7600-MUX-UNI Support on LAN Cards feature provides the ability to partition a physical port on an attachment VLAN to provide multiple Layer 2 and Layer 3 services over a single UNI.

When configuring 7600-MUX-UNI Support on LAN Cards, follow these guidelines and restrictions:

- Encapsulation on main interface has to be dot1Q and not ISL
- With dot1q encapsulation on the main interface, you cannot configure ISL on the subinterfaces; Layer 3 interfaces are unaffected

To configure 7600-MUX-UNI Support on LAN Cards, perform this task on the provider edge (PE) routers.

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface type number`	Selects an interface to configure and enters interface configuration mode; valid only for Ethernet ports.
Step 3	`Router(config-if)# switchport`	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 4	`Router(config-if)# switchport trunk encapsulation {isl | dot1q}`	Configure the port to support 802.1Q encapsulation. You must configure each end of the link with the same encapsulation type. **Note** The valid choice for MUX-UNI Support is dot1Q.
Step 5	`Router(config-if)# switchport mode trunk`	Configure the port as a VLAN trunk
Step 6	`Router(config-if)# switchport trunk allowed vlan vlan-list`	By default, all VLANs are allowed. Use this command to explicitly allow VLANs; valid values for vlan-list are from 1 to 4094. **Note** Avoid overlapping VLAN assigments between main and subinterfaces. VLAN assigments between the main interface and subinterfaces must be mutually exclusive.
Step 7	`Router(config)# interface type slot/port.subinterface-number`	Selects a subinterface to configure and enters interface configuration mode; valid only for Ethernet ports.
Step 8	`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 9	`Router(config-if)# xconnect peer_router_id vcid encapsulation mpls`	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

This example for the 7600-MUX-UNI Support on LAN Cards feature shows a physical trunk port used as UNI:

```
interface Fastethernet3/1
switchport
switchport encapsulation dot1q
switchport mode trunk
switchport trunk allowed VLAN 200-250

interface Fastethernet3/1.10
encap dot1q 3000
xconnect 10.0.0.1 3000 encapsulation mpls
```

This example for the 7600-MUX-UNI Support on LAN Cards feature shows a Layer 2 port channel used as UNI:

```
interface Port-channel100
    switchport
    switchport trunk encapsulation dot1q
    switchport trunk allowed VLAN 100-200
    switchport mode trunk
    no ip address

interface Port-channel100.1
    encapsulation dot1Q 3100
    xconnect 10.0.0.30 100 encapsulation mpls
```

This example for the 7600-MUX-UNI Support on LAN Cards feature shows Layer 3 termination and VRF for Muxed UNI ports:

```
Vlan 200, 300, 400
interface Fastethernet3/1
switchport
switchport encapsulation dot1q
switchport mode trunk
switchport trunk allowed VLAN 200-500

interface Fastethernet3/1.10
encap dot1q 3000
xconnect 10.0.0.1 3000 encapsulation mpls

interface Vlan 200
ip address 1.1.1.3

interface Vlan 300
ip vpn VRF A
ip address 3.3.3.1

interface Vlan 400
ip address 4.4.4.1
ip ospf network broadcast
mpls label protocol ldp
mpls ip
```

Troubleshooting

This section describes how to troubleshoot common AToMPLS, EoMPLS and MPLS VPN issues.

Scenarios/Problems	Solution

How do I verify whether MPLS is enabled on an interface?	Use the **show mpls interfaces** command. This is a sample output:
	`PE1#show mpls interfaces`
	`Interface IP Tunnel BGP Static Operational`
	`GigabitEthernet1/1 Yes (ldp) Yes No No Yes`
	`GigabitEthernet1/1 Yes (ldp) Yes No No Yes`
	`Tunnel2 No Yes No No Yes`
	`Tunnel1 No Yes No No Yes`

How do I verify whether LDP neighborship is established between the PE routers?	Use the **show mpls ldp neighbor** command. This is a sample output:
	`PE1#show mpls ldp neighbor`
	`Peer LDP Ident: 11.11.11.11:0; Local LDP Ident 10.10.10.10:0`
	`TCP connection: 11.11.11.11.32784 - 10.10.10.10.646`
	`State: Oper; Msgs sent/rcvd: 1073/1061; UPstream`
	`Up time: 14:53:49`
	`LDP discovery sources:`
	`GigabitEthernet1/1, Src IP addr: 110.110.110.1`
	`Targeted Hello 10.10.10.10 -> 11.11.11.11, active <<-- This should be 'active'.`
	`Addresses bound to peer LDP Ident:`
	`11.11.11.11 7.23.8.20 120.120.120.2 110.110.110.1`

Scenarios/Problems	Solution

How do I verify whether the VC statuses are UP?	Use the **show mpls l2transport vc** command. This is a sample output:
	`PE1#show mpls l2transport vc`
	`Local intf Local circuit Dest address VC ID Status`
	`------------ ------------------------- --------------- ----------`
	`ATM3/1/1 or Gi3/2/1.1004 ATM AAL5 100/100 11.11.11.11 200 UP <<---- Shows VC status UP`
	
	To check the detailed VC status, use the **show mpls l2transport vc detail command**. This example shows a sample output of the command. The important points that needs to be checked are highlighted:
	`PE1#show mpls l2transport vc 200 detail`
	`Local interface: ATM3/1/1 or Gi3/2/1.1004 up, line protocol up, ATM AAL5 100/100 up <<-- Everything here should be up, else check AC-side interface status`
	`Destination address: 11.11.11.11, VC ID: 200, VC status: up <<-- VC status should be UP`
	`Output interface: GigabitEthernet1/1, imposed label stack {17} <<-- Outgoing interface & label stack should NEVER be blank.`
	`Preferred path: not configured`
	`Default path: active`
	`Next hop: point2point`
	`Create time: 1d02h, last status change time: 00:00:11`
	`Signaling protocol: LDP, peer 11.11.11.11:0 up`
	`Targeted Hello: 10.10.10.1(LDP Id) -> 11.11.11.11, LDP is UP <<-- LDP should be UP`
	`Status TLV support (local/remote) : enabled/supported`
	`LDP route watch : enabled`
	`Label/status state machine : established, LruRru <<-- 'Lru' indicates Local-Ready-Up, 'Rru' indicates Remote-Ready-Up`
	`Last local dataplane status rcvd: No fault <<-- Should not show faults, else check the fault shown.`
	`Last local SSS circuit status rcvd: No fault <<-- Should not show faults, else check the fault shown.`
	`Last local SSS circuit status sent: No fault <<-- Should not show faults, else check the fault shown.`
	`Last local LDP TLV status sent: No fault <<-- Should not show faults, else check the fault shown.`
	`Last remote LDP TLV status rcvd: No fault <<-- Should not show faults, else check the fault shown.`
	`Last remote LDP ADJ status rcvd: No fault <<-- Should not show faults, else check the fault shown.`
	`MPLS VC labels: local 41, remote 17 <<-- (Important) Shows the local and remote LABELS negotiated by LDP.`
	`Group ID: local 0, remote 0`
	`MTU: local 4470, remote 4470 <<-- Check if MTUs are correct`
	`Remote interface description:`
	`Sequencing: receive disabled, send disabled`
	`Control Word: Off`
	`VCCV BFD protection active <<-- Check if VCCV-BFD is configured and applied on this VC.`
	`BFD Template - nsn`
	`CC Type - 1`
	`CV Type - fault detection only with IP/UDP headers`
	`VC statistics: <<--- Displays stats of traffic flowing through this VC.`
	`transit packet totals: receive 40366, send 1405`
	`transit byte totals: receive 2099032, send 84300`
	`transit packet drops: receive 0, seq error 0, send 0`

Scenarios/Problems	Solution

How do I check the VC summary?	Use the **show mpls l2transport summary** command. This is a sample output:
	`PE1#show mpls l2transport summary`
	`Destination address: 11.11.11.11, total number of vc: 1`
	`0 unknown, 1 up, 0 down, 0 admin down, 0 recovering, 0 standby, 0 hotstandby`
	`1 active vc on MPLS interface GigabitEthernet1/1`

How do I verify whether the adjacency is present or not?	Use the **show mls cef mpls labels detail** command. In the command ouput check whether the EoS bit and the adjacency entry is all proper as expected. This is a sample output:
	`PE1#show mls cef mpls labels 41 detail`
	`Codes: M - mask entry, V - value entry, A - adjacency index, P - FIB Priority`
	`D - FIB Don't short-cut, m - mod-num, E - ELSP?`
	`Format: MPLS - (b | xtag vpn pi cr mcast label1 exp1 eos1 valid2 label2 exp2 eos2)`
	`V(2231 ): B | 1 0 0 0 0 16 0 1 0 0 0 0 (A:262144 ,P:0,D:0,m:0 :E:1)`
	`M(2231 ): F | 1 FFF 0 0 1 FFFFF 0 1 0 0 0 0`
	In the above output, 262144 is the adjacency. To check further details of this adjacency, use the **show mls cef adjacency entry** command. This is a sample output:
	`PE1#show mls cef adjacency entry 262144 detail`
	`Index: 262144 smac: a100.000d.0003, dmac: 0000.0000.000d`
	`mtu: 4504, vlan: 1022, dindex: 0xBF, l3rw_vld: 1 <<-- outgoing interface's vlan, l3rw_vld = EARL does rewrite on the packet.`
	`format: MPLS, flags: 0x8600 <<-- flags`
	`label0: 0, exp: 0, ovr: 0`
	`label1: 0, exp: 0, ovr: 0`
	`label2: 9, exp: 0, ovr: 0 <<--- Label imposed`
	`op: REPLACE_LABEL2 <<--- Label operation performed (REPLACE, PUSH, POP)`
	`packets: 558937, bytes: 36115682 <<--- Traffic stats of packets hitting this adjacency.`

How do I check the LFIB entries with the specified VPN routing and forwarding (VRF) instance?	Use the **show mpls forwarding-table vrf** command**.** This is a sample output:
	`PE1#show mpls forwarding-table vrf vrf401`
	`Local Outgoing Prefix Bytes Label Outgoing Next Hop`
	`Label Label or VC or Tunnel Id Switched interface`
	`36 Pop Label IPv4 VRF[V] 472 aggregate/vrf401`

How do I check the internal VLAN allocation?	Use the **show vlan internal usage** command. This example shows how to display the internal VLAN allocation for a specific VLAN:
	`Router# show vlan id 1030 internal usage`
	`VLAN Usage`
	`---- --------------------`
	`1030 GigabitEthernet1/2`

How do I display information about the VPN ID Cisco Express Forwarding table?	Use the **show mls cef vpn** command. This is a sample output:
	`PE1-sp#show mls cef vpn 256 166.1.1.0`
	`Codes: decap - Decapsulation, + - Push Label`
	`Index Prefix Adjacency`
	`3221 166.1.1.0/24 PO9/2/0 49 ,18`

Scenarios/Problems	Solution

How do I collect the adjacency-entry information for a specified index?	Use the **show mls cef adjacency entry** command. This example shows the detailed adjacency-entry information:
	`PE1-sp#show mls cef adjacency entry 98305 detail`
	`Index: 98305  smac: 0013.1abf.3300, dmac: 0000.0950.ffff`
	`mtu: 4488, vlan: 1034, dindex: 0x0, l3rw_vld: 1`
	`format: MPLS, flags: 0x208418`
	`label0: 0, exp: 0, ovr: 0`
	`label1: 49, exp: 0, ovr: 0`
	`label2: 18, exp: 0, ovr: 0`
	`op: PUSH_LABEL2_LABEL1`
	`packets: 0, bytes: 0`

Scenarios/Problems	Solution

How I do debug the control plane events?	Use the **debug mpls l2transport vc** command. This is a sample output:
	`Router# debug mpls l2transport vc event`
	`AToM vc event debugging is on`
	`Router# debug mpls l2transport vc fsm`
	`AToM vc fsm debugging is on`
	`Router# show debugging`
	`AToM:`
	`  AToM vc event debugging is on`
	`  AToM vc fsm debugging is on`
	`*Mar 24 23:17:24.371: AToM MGR [10.9.9.9, 50]: Event provision, state changed from idle to provisioned`
	`*Mar 24 23:17:24.371: AToM MGR [10.9.9.9, 50]: Provision vc`
	`*Mar 24 23:17:24.371: AToM SMGR [10.9.9.9, 50]: Requesting VC create, vc_handle 61A09930`
	`*Mar 24 23:17:24.371: AToM MGR [10.9.9.9, 50]: Event local up, state changed from provisioned to local standby`
	`*Mar 24 23:17:24.371: AToM MGR [10.9.9.9, 50]: Update local vc label binding`
	`*Mar 24 23:17:24.371: AToM SMGR [10.9.9.9, 50]: sucessfully processed create request`
	`*Mar 24 23:17:24.875: %SYS-5-CONFIG_I: Configured from console by console`
	`*Mar 24 23:17:25.131: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up`
	`*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Event ldp up, state changed from local standby to local ready`
	`*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Advertise local vc label binding`
	`*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Event remote up, state changed from local ready to establishing`
	`*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Remote end up`
	`*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Event remote validated, state changed from establishing to established`
	`*Mar 24 23:17:28.567: AToM MGR [10.9.9.9, 50]: Validate vc, activating data plane`
	`*Mar 24 23:17:28.567: AToM SMGR [10.9.9.9, 50]: Processing imposition update, vc_handle 61A09930, update_action 3, remote_vc_label 21`
	`*Mar 24 23:17:28.567: AToM SMGR [10.9.9.9, 50]: Imposition Programmed, Output Interface: PO5/0`
	`*Mar 24 23:17:28.567: AToM SMGR [10.9.9.9, 50]: Processing disposition update, vc_handle 61A09930, update_action 3, local_vc_label 22`
	`*Mar 24 23:17:28.571: AToM SMGR: Processing TFIB event for 10.9.9.9`
	`*Mar 24 23:17:28.571: AToM SMGR [10.9.9.9, 50]: Imposition Programmed, Output Interface: PO5/0`

Scenarios/Problems	Solution

How do I debug xconnect segments?	Use the **debug ssm cm** command. This example shows the events that occur on the CM and SM when an AToM VC is provisioned and then unprovisioned:
	
	`Router# debug ssm cm events`
	`SSM Connection Manager events debugging is on`
	`Router# debug ssm sm events`
	`SSM Segment Manager events debugging is on`
	`Router# configure terminal`
	`Router(config)# interface ethernet1/0`
	`Router(config-if)# xconnect 10.55.55.2 101 pw-class mpls`
	`16:57:34: SSM CM: provision switch event, switch id 86040`
	`16:57:34: SSM CM: [Ethernet] provision first segment, id 12313`
	`16:57:34: SSM CM: CM FSM: state Idle - event Provision segment`
	`16:57:34: SSM CM: [SSS:Ethernet:12313] provision segment 1`
	`16:57:34: SSM SM: [SSS:Ethernet:12313] event Provison segment`
	`16:57:34: SSM CM: [SSS:Ethernet] shQ request send ready event`
	`16:57:34: SSM CM: SM msg event send ready event`
	`16:57:34: SSM SM: [SSS:Ethernet:12313] segment ready`
	`16:57:34: SSM SM: [SSS:Ethernet:12313] event Found segment data`
	`16:57:34: SSM CM: Query AToM to Ethernet switching, enabled`
	`16:57:34: SSM CM: [AToM] provision second segment, id 16410`
	`16:57:34: SSM CM: CM FSM: state Down - event Provision segment`
	`16:57:34: SSM CM: [SSS:AToM:16410] provision segment 2`
	`16:57:34: SSM SM: [SSS:AToM:16410] event Provison segment`
	`16:57:34: SSM CM: [AToM] send client event 6, id 16410`
	`16:57:34: label_oce_get_label_bundle: flags 14 label 19`
	`16:57:34: SSM CM: [SSS:AToM] shQ request send ready event`
	`16:57:34: SSM CM: SM msg event send ready event`
	`16:57:34: SSM SM: [SSS:AToM:16410] segment ready`
	`16:57:34: SSM SM: [SSS:AToM:16410] event Found segment data`
	`16:57:34: SSM SM: [SSS:AToM:16410] event Bind segment`
	`16:57:34: SSM SM: [SSS:Ethernet:12313] event Bind segment`
	`16:57:34: SSM CM: [AToM] send client event 3, id 16410`

Scenarios/Problems	Solution
	Router# **configure terminal** Router(config)# **interface e1/0** Router(config-if)# **no xconnect** 16:57:26: SSM CM: [Ethernet] unprovision segment, id 16387 16:57:26: SSM CM: CM FSM: state Open - event Free segment 16:57:26: SSM CM: [SSS:Ethernet:16387] unprovision segment 1 16:57:26: SSM SM: [SSS:Ethernet:16387] event Unprovison segment 16:57:26: SSM CM: [SSS:Ethernet] shQ request send unprovision complete event 16:57:26: SSM CM: [SSS:AToM:86036] unbind segment 2 16:57:26: SSM SM: [SSS:AToM:86036] event Unbind segment 16:57:26: SSM CM: SM msg event send unprovision complete event 16:57:26: SSM SM: [SSS:Ethernet:16387] free segment class 16:57:26: SSM SM: [SSS:Ethernet:16387] free segment 16:57:26: SSM SM: [SSS:Ethernet:16387] event Free segment 16:57:26: SSM SM: last segment class freed 16:57:26: SSM CM: unprovision switch event, switch id 12290 16:57:26: SSM CM: [SSS:AToM] shQ request send unready event 16:57:26: SSM CM: SM msg event send unready event 16:57:26: SSM SM: [SSS:AToM:86036] event Unbind segment 16:57:26: SSM CM: [AToM] unprovision segment, id 86036 16:57:26: SSM CM: CM FSM: state Down - event Free segment 16:57:26: SSM CM: [SSS:AToM:86036] unprovision segment 2 16:57:26: SSM SM: [SSS:AToM:86036] event Unprovison segment 16:57:26: SSM CM: [SSS:AToM] shQ request send unprovision complete event 16:57:26: SSM CM: SM msg event send unprovision complete event 16:57:26: SSM SM: [SSS:AToM:86036] free segment class 16:57:26: SSM SM: [SSS:AToM:86036] free segment 16:57:26: SSM SM: [SSS:AToM:86036] event Free segment 16:57:26: SSM SM: last segment class freed

CHAPTER 25

Configuring IPv4 Multicast VPN Support

This chapter describes how to configure IPv4 Multicast Virtual Private Network (MVPN) support on Cisco 7600 series routers. MVPN is supported when a PFC3B, PFC3BXL, PFC3C, or PFC3CXL is installed in the router.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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 Cisco 7600 series routers uses the existing PFC 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.

2. If a particular multicast stream exceeds the defined threshold, the sending PE router dynamically creates a data MDT for that particular multicast traffic.
3. The sending PE router then transmits a DATA-MDT JOIN request (which is a User Datagram Protocol (UDP) message to port 3232) to the other PE routers, informing them of the new data MDT.
4. Receiving PE routers examine their VRF routing tables to determine if they have any customers interested in receiving this data stream. If so, they use the PIM protocol to transmit a PIM JOIN message for this particular data MDT group (in the global table PIM instance) to accept the stream. Routers that do not currently have any customers for this stream still cache the information, in case any customers request it later on.
5. Three seconds after sending the DATA-MDT JOIN message, the sending PE router removes the high-bandwidth multicast stream from the default MDT and begins transmitting it over the new data MDT.
6. The sending PE router continues to send a DATA-MDT JOIN message every 60 seconds, as long as the multicast stream continues to exceed the defined threshold. If the stream falls below the threshold for more than 60 seconds, the sending PE router stops sending the DATA-MDT JOIN messages, and moves the stream back to the default MDT.
7. Receiving routers age out the cache information for the default MDT when they do not receive a DATA-MDT JOIN message for more than three minutes.

Data MDTs allow for high-bandwidth sources inside the VPN while still ensuring optimal traffic forwarding in the MPLS VPN core.

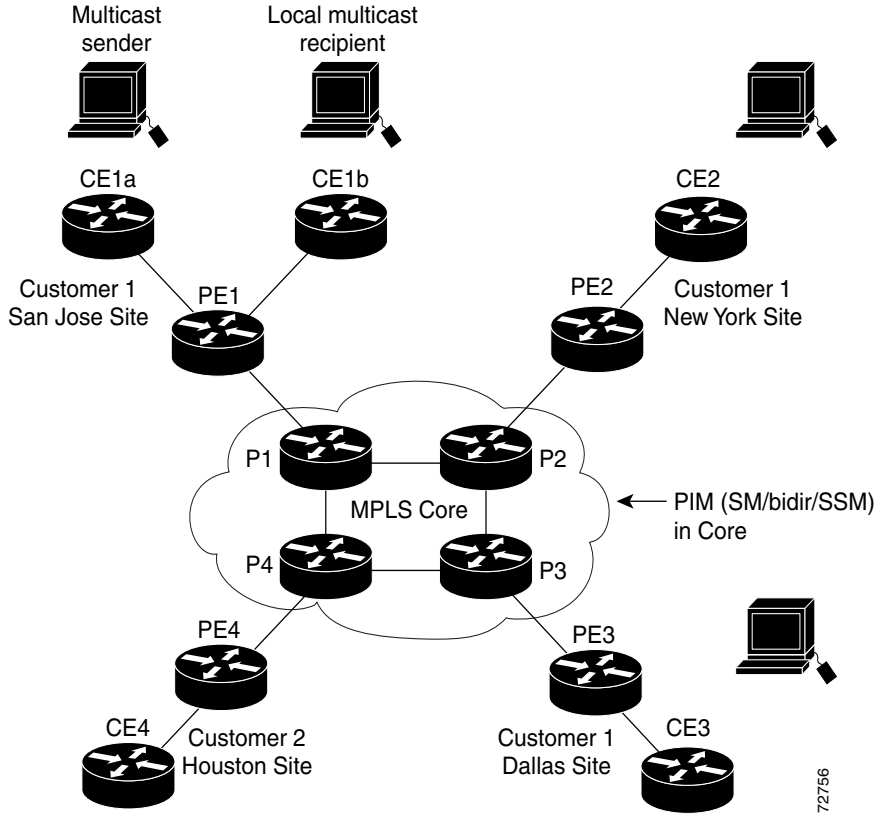
Note For technical information about the DATA-MDT JOIN message and other aspects of the data MDT creation and usage, see the Internet-Draft, *Multicast in MPLS/BGP IP VPNs*, by Eric C. Rosen et al.

In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. The San Jose site is transmitting a one-way multicast presentation. The service provider network supports all three sites associated with this customer, in addition to the Houston site of a different enterprise customer.

The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. Although PE4 is interconnected to these other routers in the MPLS core, PE4 is associated with a different customer and is therefore not part of the default MDT.

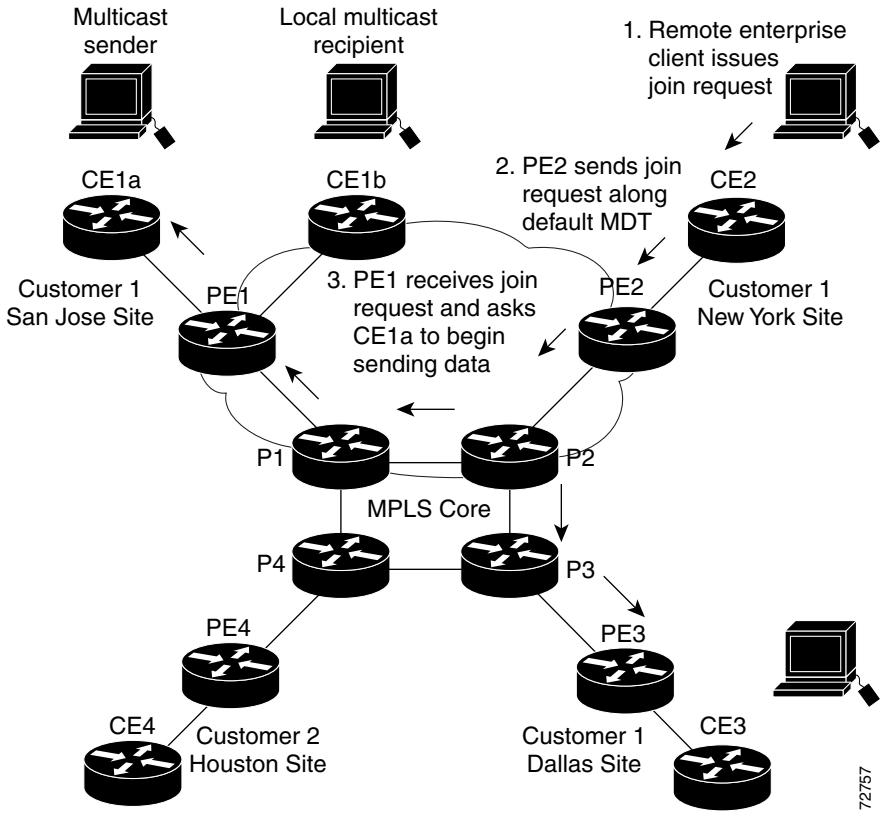
Figure 25-1 shows the situation in this network when no one outside of San Jose has joined the multicast broadcast, which means that no data is flowing along the default MDT. Each PE router maintains a PIM relationship with the other PE routers over the default MDT, as well as a PIM relationship with its directly attached PE routers.

Figure 25-1 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 25-2 shows how the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 25-2 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

The PFC supports hardware acceleration for IPv4 multicast over VPN traffic, which forwards multicast traffic to the appropriate VPNs at wire speed without increased MSFC3 CPU utilization.

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

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

MVPN Configuration Guidelines and Restrictions

When configuring MVPN, follow these guidelines and restrictions:

- The Cisco 7600 series router must have a PFC3B, PFC3BXL, PFC3C, or PFC3CXL intalled to run MVPN.
- All PE routers in the multicast domain need to be running a Cisco IOS software image that supports the MVPN feature. There is no requirement for MVPN support on the P and CE routers.
- Support for IPv4 multicast traffic must also be enabled on all backbone routers.
- The Border Gateway Protocol (BGP) routing protocol must be configured and operational on all routers supporting multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.
- Only ingress replication is supported when MVPN is configured. If the router is currently configured for egress replication, it is forced into ingress replication when the first MVRF is configured.
- When the router 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 and RPR+ redundancy modes. MPLS can coexist with NSF with SSO redundancy mode, but there is no support for stateful MPLS switchover.
- If your MPLS VPN network already contains a network of VRFs, you do not need to delete them or recreate them to be able to support MVRF traffic. Instead, configure the **mdt default** and **mdt data** commands, as listed in the following procedure, to enable multicast traffic over the VRF.
- BGP should be already configured and operational on all routers that are sending or receiving multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.
- The same MVRF must be configured on each PE router that is to support a particular VPN connection.
- Each PE router that supports a particular MVRF must be configured with the same **mdt default** command.
- The router supports only ingress replication when MVPN is configured. If a router is currently configured for egress replication, it is forced into ingress replication when the first MVRF is configured. If a router 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 25-8
- Configuring a Multicast VPN Routing and Forwarding Instance, page 25-10
- Configuring Multicast VRF Routing, page 25-15
- Configuring Interfaces for Multicast Routing to Support MVPN, page 25-20

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 router 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 router into software switching until the table entries can be rebuilt.

To avoid disrupting customer traffic, we recommend verifying that the router is already in ingress multicast replication mode before configuring any MVRFs.

This example shows how to verify the multicast replication mode:

```
Router# show platform software multicast ip capability
Current System HW Replication Mode : Egress
Auto-detection of Replication Mode : ON

Slot Replication-Capability Replication-Mode
   2 Egress                 Egress
   3 Egress                 Egress
   4 Egress                 Egress
   6 Egress                 Egress
Router#
```

Note Starting 12.2 (33) SRE release, the **show mls ip multicast capability** command is changed to **show platform software multicast ip capability.**

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)# ip multicast hardware-switching replication-mode {egress | ingress}`	Configures ingress multicast replication mode and disables automatic detection of the replication mode (enabled by default).
	`Router(config)# no ip multicast hardware-switching replication-mode ingress`	Enables automatic detection of the replication mode.
Step 3	`Router(config)# show platform software multicast ip capability`	Verifies the configuration.

This example shows how to configure ingress multicast replication mode and verify the configuration:

```
Router(config)# ip multicast hardware-switching replication-mode ingress
Router(config)# show platform software multicast ip capability
Current System HW Replication Mode : Egress
Auto-detection of Replication Mode : ON

Slot Replication-Capability Replication-Mode
   2 Egress                 Egress
   3 Egress                 Egress
   4 Egress                 Egress
   6 Egress                 Egress
Router#
```

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 or Action	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 or Action	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 or Action	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 or Action	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:

- 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 or Action	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 bitmask 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 or Action	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 router, 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 display entries for a specific VRF, use the **show platform software multicast ip vrf** command. The following example shows typical output for this command:

```
Router# show platform software multicast ip vrf

  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 or Action	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 or Action	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 25-10.
- **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 or Action	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 or Action	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 or Action	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 or Action	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 or Action	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 or Action	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

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 or Action	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface type {slot/port | number}`	Enters interface configuration mode for the specified interface.
Step 3	`Router(config-if)# ip pim {dense-mode | sparse-mode | sparse-dense-mode}`	Enables PIM on the interface.
	`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 or Action	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

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
ip multicast hardware-switching 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
ip multicast hardware-switcing 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
...
```

Troubleshooting

This section describes how to troubleshoot common Multicast VPN issues.

Scenarios/Problems	Solution

How do I display the PIM RP information?	Use the **show ip pim rp** command. This example shows a sample output from the command:
	`PE1#show ip pim rp map`
	`PIM Group-to-RP Mappings`
	`Group(s) 224.0.0.0/4`
	`RP 1.1.1.1 (?), v2v1 <------------------Rp is P1`
	`Info source: 1.1.1.1 (?), elected via Auto-RP`
	`Uptime: 5d20h, expires: 00:02:22`

How do I display the detailed PIM information for each VRF instance?	Use the **show ip pim vrf** command**.** This example shows a sample output from the command:
	`PE1#show ip pim vrf red neighbor`
	`PIM Neighbor Table`
	`Neighbor Interface Uptime/Expires Ver DR Address Prio/Mode`
	`10.10.4.2 FastEthernet3/2 3d20h/00:01:41 v2 N /`
	`200.200.200.200 Tunnel0 00:08:40/00:01:29 v2 1 / DR B S <-------Pim neighbor on the MTI Interface`

How do I check the multicast routing table for a specific VRF instance?	Use the **show ip mroute vrf** command. This example shows a sample output from the command:
	`PE1#show ip mroute vrf red`
	`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`
	`(*, 239.10.10.15), 00:11:01/00:03:16, RP 10.10.4.2, flags: S`
	`Incoming interface: FastEthernet3/2, RPF nbr 10.10.4.2, RPF-MFD`
	`Outgoing interface list:`
	`Tunnel0, Forward/Sparse-Dense, 00:11:01/00:03:16, H`
	`(10.10.2.1, 239.10.10.15), 00:10:32/00:03:27, flags: T*y <-------y flag indicates it is switched to Data MDT*`
	`Incoming interface: FastEthernet3/2, RPF nbr 10.10.4.2, RPF-MFD`
	`Outgoing interface list:`
	`Tunnel0, Forward/Sparse-Dense, 00:11:01/00:03:16, H`
	`(*, 239.10.10.16), 00:11:02/00:03:20, RP 10.10.4.2, flags: S`
	`Incoming interface: FastEthernet3/2, RPF nbr 10.10.4.2, RPF-MFD`
	`Outgoing interface list:`
	`Tunnel0, Forward/Sparse-Dense, 00:11:02/00:03:20, H`
	`(10.10.2.1, 239.10.10.16), 00:10:32/00:03:27, flags: T*y*`
	`Incoming interface: FastEthernet3/2, RPF nbr 10.10.4.2, RPF-MFD`
	`Outgoing interface list:`
	`Tunnel0, Forward/Sparse-Dense, 00:11:02/00:03:20, H`

Scenarios/Problems	Solution

How do I check a multicast route?	Use the **show ip mroute** command. This example shows a sample output from the command:
	
	`PE1#show ip mroute`
	`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`
	`(*, 226.1.1.1), 00:04:22/stopped, RP 1.1.1.1, flags: SJCFZ`
	`Incoming interface: FastEthernet3/3, RPF nbr 10.10.6.1, RPF-MFD`
	`Outgoing interface list:`
	`MVRF red, Forward/Sparse-Dense, 00:03:47/00:02:12, H`
	`(*, 226.1.1.128), 00:04:26/stopped, RP 1.1.1.1, flags: SJPFZ`
	`Incoming interface: FastEthernet3/3, RPF nbr 10.10.6.1, Partial-SC`
	`Outgoing interface list: Null`
	`(100.100.100.100, 226.1.1.128), 00:03:52/00:03:28, flags: FTZ <--Data MDT`
	`Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD`
	`Outgoing interface list:`
	`FastEthernet3/3, Forward/Sparse-Dense, 00:03:53/00:02:34, H`
	`(*, 226.1.1.129), 00:04:27/stopped, RP 1.1.1.1, flags: SJPFZ`
	`Incoming interface: FastEthernet3/3, RPF nbr 10.10.6.1, Partial-SC`
	`Outgoing interface list: Null`
	`(100.100.100.100, 226.1.1.129), 00:03:53/00:03:27, flags: FTZ <---Data MDT`
	`Incoming interface: Loopback0, RPF nbr 0.0.0.0, RPF-MFD`
	`Outgoing interface list:`
	`FastEthernet3/3, Forward/Sparse-Dense, 00:03:53/00:02:31, H`
	
	To display the output of a specific multicast route, specify the IP multicast group address with the command. For instance, **show ip mroute 232.6.6.6**

How do I verify that the multicast packets are incrementing on the ingress interface?	Use the **show interface counters** command. This example shows a sample output from the command:
	
	`PE1#show int f3/2 counters`
	`Port InOctets InUcastPkts InMcastPkts InBcastPkts`
	`Fa3/2 426816 1 6668 0`
	`Port OutOctets OutUcastPkts OutMcastPkts OutBcastPkts`
	`Fa3/2 80 0 1 0`
	
	If the ingress is DFC, then log onto the module. If you do not have DFC module, then log onto the PFC.

Scenarios/Problems	Solution

How do I display the IP entries in the Multilayer Switching (MLS)-hardware Layer 3-switching table on the switch processor?	Use the **show mls cef ip multicast** command. This example shows a sample output from the command:
	`PE1-sp#show mls cef ip multicast vrf red group 239.10.10.15`
	`Multicast CEF Entries for VPN#1`
	`Flags: R - Control, S - Subnet, B - Bidir, C - Complete, P - Partial, E - Encapsulation, D - Decapsulation`
	`c - Central Rewrite, p - Primary Input, r - Recirculation, h - Entry sitting on Encap/Decap VRF layer`
	`Source/mask Destination/mask RPF/DF Flags #packets #bytes rwindex Output Vlans/Info`
	`+-------+-----------+-----+-----+-------+-------+-------+-------------+--------+`
	`* 239.10.10.15/32 Vl1020 Cp 93 5952 0x7FFA E-VRF"red"(1018) [1 oifs]`
	`10.10.2.1/32 239.10.10.15/32 Vl1020 Cp 1158932 74171648 0x7FFA E-VRF"red"(1018) [1 oifs]`
	`Found 2 entries. 2 are mfd entries`

How do I know the mapping information about the platform software for Virtual Private Networks (VPNs)?	Use the **show platform software vpn** command. This example shows a sample output from the command:
	`PE1-sp#show platform software vpn mapping`
	`Type | VRF Name | Table id || HW table id | App Bitmask | App Data mask | Reference counters and App data`
	`----------+---------+------||--------+--------+----------+-----------+----------------------------`
	`IOS | Default-table | 0 || 0 | 0x00000031 | 0x00000000 | R[0]:26`
	`IOS | red | 1 || 256 | 0x00000015 | 0x00000000 | R[0]:10`
	`MLS | | 0 || 257 | 0x00000010 | 0x00000010 | D[4]:0x0`
	`MLS | | 1 || 258 | 0x00000010 | 0x00000010 | D[4]:0x1`
	`MLS | | 4094 || 4094 | 0x00000020 | 0x00000000 |`
	`MLS | | 4095 || 4095 | 0x00000020 | 0x00000000 |`

How do I display information about Multicast Multilayer Switching (MMLS)?	Use the **show mmls msc** command. This example shows a sample output from the command:
	`PE1-sp#show mmls msc mdt`
	`MDT GROUP VRF Name IOSID hwid ENCVLAN DECVLAN ENCID hwid DECID hwid Ref Count`
	`+--------+------+----+---+-------+-------+-----+---+-----+-----+------+----`
	`226.1.1.1 red 1 256 1018 1019 0 257 1 258 1`
	`226.1.1.128 red 1 256 1018 1019 0 257 1 258 1`
	`226.1.1.129 red 1 256 1018 1019 0 257 1 258 1`

Scenarios/Problems	Solution

How do I display information about activity in the multicast route (mroute) table?	Use the **debug ip mrouting** command. This is a sample output of the command:
	`router# debug ip mrouting 224.2.0.1`
	`MRT: Delete (10.0.0.0/8, 224.2.0.1)`
	`MRT: Delete (10.4.0.0/16, 224.2.0.1)`
	`MRT: Delete (10.6.0.0/16, 224.2.0.1)`
	`MRT: Delete (10.9.0.0/16, 224.2.0.1)`
	`MRT: Delete (10.16.0.0/16, 224.2.0.1)`
	`MRT: Create (*, 224.2.0.1), if_input NULL`
	`MRT: Create (224.69.15.0/24, 225.2.2.4), if_input Ethernet0, RPF nbr 224.69.61.15`
	`MRT: Create (224.69.39.0/24, 225.2.2.4), if_input Ethernet1, RPF nbr 0.0.0.0`
	`MRT: Create (10.0.0.0/8, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0`
	`MRT: Create (10.4.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0`
	`MRT: Create (10.6.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0`
	`MRT: Create (10.9.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0`
	`MRT: Create (10.16.0.0/16, 224.2.0.1), if_input Ethernet1, RPF nbr 224.0.0.0`

How do I know the PIM packets received, sent, and also the PIM-related events?	Use the **debug ip pim** command. This is a sample output of the command:
	`router# debug ip pim 224.2.0.1`
	`PIM: Received Join/Prune on Ethernet1 from 172.16.37.33`
	`PIM: Received Join/Prune on Ethernet1 from 172.16.37.33`
	`PIM: Received Join/Prune on Tunnel0 from 10.3.84.1`
	`PIM: Received Join/Prune on Ethernet1 from 172.16.37.33`
	`PIM: Received Join/Prune on Ethernet1 from 172.16.37.33`
	`PIM: Received RP-Reachable on Ethernet1 from 172.16.20.31`
	`PIM: Update RP expiration timer for 224.2.0.1`
	`PIM: Forward RP-reachability packet for 224.2.0.1 on Tunnel0`
	`PIM: Received Join/Prune on Ethernet1 from 172.16.37.33`
	`PIM: Prune-list (10.221.196.51/32, 224.2.0.1)`
	`PIM: Set join delay timer to 2 seconds for (10.221.0.0/16, 224.2.0.1) on Ethernet1`
	`PIM: Received Join/Prune on Ethernet1 from 172.16.37.6`
	`PIM: Received Join/Prune on Ethernet1 from 172.16.37.33`
	`PIM: Received Join/Prune on Tunnel0 from 10.3.84.1`
	`PIM: Join-list: (*, 224.2.0.1) RP 172.16.20.31`
	`PIM: Add Tunnel0 to (*, 224.2.0.1), Forward state`
	`PIM: Join-list: (10.0.0.0/8, 224.2.0.1)`
	`PIM: Add Tunnel0 to (10.0.0.0/8, 224.2.0.1), Forward state`
	`PIM: Join-list: (10.4.0.0/16, 224.2.0.1)`
	`PIM: Prune-list (172.16.84.16/28, 224.2.0.1) RP-bit set RP 172.16.84.16`
	`PIM: Send Prune on Ethernet1 to 172.16.37.6 for (172.16.84.16/28, 224.2.0.1), RP`
	`PIM: For RP, Prune-list: 10.9.0.0/16`
	`PIM: For RP, Prune-list: 10.16.0.0/16`
	`PIM: For RP, Prune-list: 10.49.0.0/16`
	`PIM: For RP, Prune-list: 10.84.0.0/16`
	`PIM: For RP, Prune-list: 10.146.0.0/16`
	`PIM: For 10.3.84.1, Join-list: 172.16.84.16/28`
	`PIM: Send periodic Join/Prune to RP via 172.16.37.6 (Ethernet1)`

Scenarios/Problems	Solution

How do I display information about Multilayer Switching Protocol (MLSP)?	Use the **debug mls rp ip multicast** command. This example shows output from the command using the **error** keyword:
	`Router# debug mls rp ip multicast error`
	`mlsm error debugging is on`
	`chtang-7200#`
	`06:06:45: MLSMERR: scb is INACTIVE, free INSTALL_FE`
	`06:06:46: MLSM: --> mlsm_proc_sc_ins_req(10.0.0.1, 224.2.2.3, 10)`

How do I display the run-time errors and sequence of events for the multicast distributed switching services (MDSS)?	Use the **debug mdss** command. This example shows the output of the command with **all** keyword:
	Note This debug command is specific to 12.2(33)SRD Release. This will not work in 12.2(33)SRE Release and later.
	`Router# debug mdss all`
	`mdss all debugging is on`
	`Router# clear ip mroute *`
	`Router#`
	`01:31:03: MDSS: got MDFS_CLEARALL`
	`01:31:03: MDSS: --> mdss_flush_all_sc`
	`01:31:03: MDSS: enqueue a FE_GLOBAL_DELETE`
	`01:31:03: MDSS: got MDFS_MROUTE_ADD for (0.0.0.0, 224.0.1.40)`
	`01:31:03: MDSS: --> mdss_free_scmdb_cache`
	`01:31:03: MDSS: got MDFS_MROUTE_ADD for (0.0.0.0, 239.255.158.197)`
	`01:31:03: MDSS: got MDFS_MROUTE_ADD for (192.1.21.6, 239.255.158.197)`
	`01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197, Vlan21) +Vlan22`
	`01:31:03: MDSS: -- mdss_add_oif`
	`01:31:03: MDSS: enqueue a FE_OIF_ADD (192.1.21.6, 239.255.158.197, Vlan21) +Vlan22`
	`01:31:03: MDSS: mdb (192.1.21.6, 239.255.158.197) fast_flags | MCACHE_MTU`
	`01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197, Vlan21) +Vlan23`
	`01:31:03: MDSS: -- mdss_add_oif`
	`01:31:03: MDSS: enqueue a FE_OIF_ADD (192.1.21.6, 239.255.158.197, Vlan21) +Vlan23`
	`01:31:03: MDSS: mdb (192.1.21.6, 239.255.158.197) fast_flags | MCACHE_MTU`
	`01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197, Vlan21) +Vlan26`
	`01:31:03: MDSS: -- mdss_add_oif`
	`01:31:03: MDSS: enqueue a FE_OIF_ADD (192.1.21.6, 239.255.158.197, Vlan21) +Vlan26`
	`01:31:03: MDSS: mdb (192.1.21.6, 239.255.158.197) fast_flags | MCACHE_MTU`
	`01:31:03: MDSS: got a MDFS_MIDB_ADD for (192.1.21.6, 239.255.158.197,u Vlan21) +Vlan27`

Scenarios/Problems	Solution

How do I track the Bidir DF?	Use the **show ip pim interface df** command. This example shows the output of the command:
	`router# show ip pim interface df`
	`Interface RP DF Winner Metric Uptime`
	`Ethernet3/3 10.10.0.2 10.4.0.2 0 00:03:49`
	`10.10.0.3 10.4.0.3 0 00:01:49`
	`10.10.0.5 10.4.0.4 409600 00:01:49`
	`Ethernet3/4 10.10.0.2 10.5.0.2 0 00:03:49`
	`10.10.0.3 10.5.0.2 409600 00:02:32`
	`10.10.0.5 10.5.0.2 435200 00:02:16`
	`Loopback0 10.10.0.2 10.10.0.2 0 00:03:49`
	`10.10.0.3 10.10.0.2 409600 00:02:32`
	`10.10.0.5 10.10.0.2 435200 00:02:16`

How do I display the mappings for the PIM-Bidir group to active rendezvous points?	Use the **show mls ip multicast rp-mapping df-cache** command. This example shows how to display information that is based on the DF list in the mapping cache of the route processor:
	`Router# show mls ip multicast rp-mapping df-cache`
	`RP Address State DF State`
	`10.9.9.9 H Vl30 H`

How do I display information on the group/mask ranges in the rendezvous-point mapping cache in the hardware?	Use the **show mls ip multicast rp-mapping gm-cache** command. This example shows how to display information that is based on the mapping cache of the route processor:
	`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`
	`10.0.0.60 H 172.16.0.0 255.255.0.0 H 100/6400`

CHAPTER 26

Configuring IP Unicast Layer 3 Switching

This chapter describes how to configure IP unicast Layer 3 switching on Cisco 7600 series routers.

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

- The Cisco 7600 Series Routers Command References at this URL:
  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- The Release 12.2 publications at this URL:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/index.htm

This chapter consists of these sections:

Note

- IPX traffic is fast switched on the MSFC. For more information, refer to this URL:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fatipx_c/index.htm
- For information about IP multicast Layer 3 switching, see Chapter 28, "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 PFC and DFCs, instead of the MSFC, to forward IP unicast traffic between subnets. Hardware Layer 3 switching provides wire-speed forwarding on the PFC and DFCs, instead of in software on the MSFC. Hardware Layer 3 switching requires minimal support from the MSFC. The MSFC routes any traffic that cannot be hardware Layer 3 switched.

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

Hardware Layer 3 switching runs equally on the PF3 and DFCs to provide IP unicast Layer 3 switching locally on each module. Hardware Layer 3 switching provides the following functions:

- Hardware access control list (ACL) switching for policy-based routing (PBR)
- Hardware NetFlow switching for TCP intercept, reflexive ACL forwarding decisions
- Hardware Cisco Express Forwarding (CEF) switching for all other IP unicast traffic

Hardware Layer 3 switching on the PFC supports modules that do not have a DFC. The MSFC 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 Cisco 7600 series router.

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 47, “Configuring NetFlow and 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 Cisco 7600 series router performs a packet rewrite at the egress port based on information learned from the MSFC so that the packets appear to have been routed by the MSFC.

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

To perform Layer 3 switching, the router 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 MSFC. The Layer 3 addresses remain the same.

In IP unicast and IP multicast traffic, the router decrements the Layer 3 TTL value by 1 and recomputes the Layer 3 packet checksum. The router 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		
MSFC MAC	*Source A MAC*	*Destination B IP*	*Source A IP*	*n*	*calculation1*		

After the router rewrites an IP unicast packet, it is formatted (conceptually) as follows:

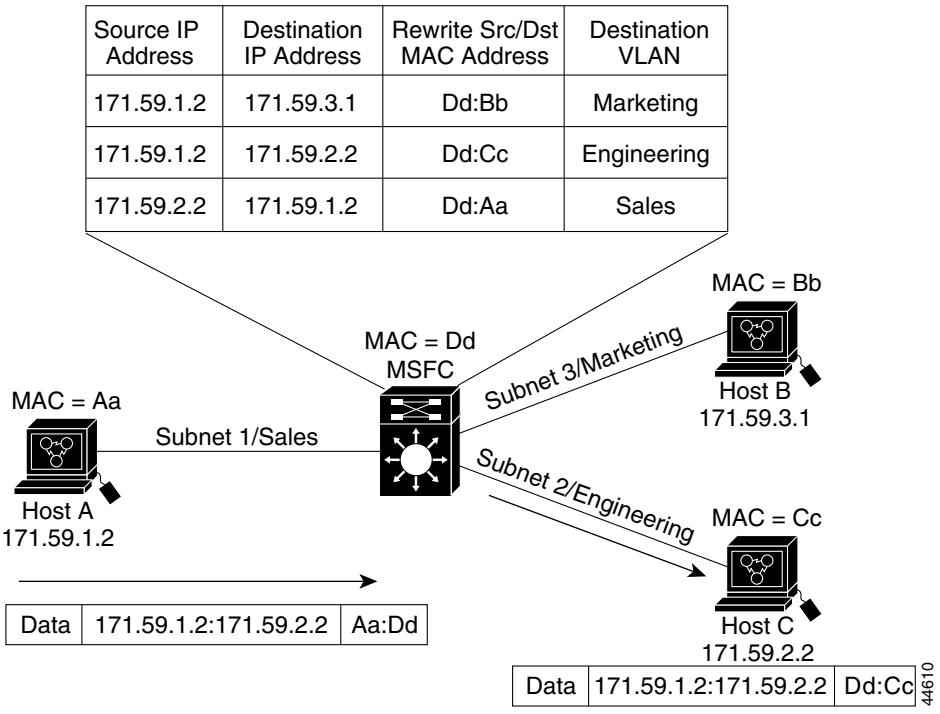
Layer 2 Frame Header		Layer 3 IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
Destination B MAC	*MSFC MAC*	*Destination B IP*	*Source A IP*	*n-1*	*calculation2*		

Hardware Layer 3 Switching Examples

Figure 26-1 on page 26-3 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 26-1 Hardware Layer 3 Switching Example Topology



Default Hardware Layer 3 Switching Configuration

Table 26-1 shows the default hardware Layer 3 switching configuration.

Table 26-1 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 MSFC	Enabled (cannot be disabled)
Cisco IOS dCEF[1] enable state on MSFC	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 MSFC, see Chapter 21, "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.

With a PFC (and DFCs, if present), hardware Layer 3 switching uses per-flow load balancing based on IP source and destination addresses. Per-flow load balancing avoids the packet reordering that can be necessary with per-packet load balancing. For any given flow, all PFC- and DFC-equipped switches make exactly the same load-balancing decision, which can result in nonrandom load balancing.

The Cisco IOS CEF **ip load-sharing per-packet**, **ip cef accounting per-prefix**, and **ip cef accounting non-recursive** commands on the MSFC apply only to traffic that is CEF-switched in software on the MSFC. The commands do not affect traffic that is hardware Layer 3 switched on the PFC or on DFC-equipped switching modules.

For information about Cisco IOS CEF and dCEF on the MSFC, refer to these publications:

- The "Cisco Express Forwarding" sections at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt1/index.htm
- The *Cisco IOS Switching Services Command Reference* publication at this URL:

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

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 27

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 Cisco 7600 series routers:

- The *Cisco IOS IPv6 Configuration Library*, "Implementing IPv6 Multicast":

  http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-multicast.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 Cisco 7600 series routers:

Features that Support IPv6 Multicast

These features support IPv6 multicast:

- RPR and RPR+ redundancy mode—See Chapter 7, "Configuring RPR and RPR+ Supervisor Engine Redundancy."
- Multicast Listener Discovery version 2 (MLDv2) snooping—See Chapter 29, "Configuring MLDv2 Snooping for IPv6 Multicast Traffic."

  Note MLDv1 snooping is not supported.

- IPv6 Multicast rate limiters—See Chapter 39, "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/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ipv6_vgf.htm

- SSM mapping for IPv6—See this publication:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ipv6_vgf.htm

IPv6 Multicast Guidelines and Restrictions

These guidelines and restrictions apply to IPv6 multicast support on Cisco 7600 series routers:

- 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 7, “Configuring RPR and RPR+ Supervisor Engine Redundancy”)
  - Ingress and egress PFC QoS (see Chapter 44, “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

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.

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:

Note The show commands in the following sections are for a router 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 multicast ipv6 capability** command:

```
Router# show platform software multicast ipv6 capability | i switching
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 : Egress
Auto-detection of Replication Mode : ON

Slot Replication-Capability Replication-Mode
   2 Egress                 Egress
   3 Egress                 Egress
   4 Egress                 Egress
   6 Egress                 Egress

PE1-7600#
```

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.
```

Displaying the Replication Mode Capabilities

This example shows how to display the replication mode capabilities of the installed modules:

```
Router# show platform software multicast ipv6 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 : Egress
Auto-detection of Replication Mode : ON

Slot Replication-Capability Replication-Mode
   2 Egress                 Egress
   3 Egress                 Egress
   4 Egress                 Egress
   6 Egress                 Egress
```

Displaying Subnet Entries

This example shows how to display subnet entries:

```
Router# show platform software multicast ipv6 connected
IPv6 Multicast Subnet entries
Flags : H - Installed in ACL-TCAM
        X - Not installed in ACL-TCAM due to
            label-full exception
```

Displaying the IPv6 Multicast Summary

This example shows how to display the IPv6 multicast summary:

```
Router# show platform software multicast ipv6 summary module 4
IPv6 Multicast Netflow SC summary on Slot[4]:
Shortcut Type               Shortcut count
----------------------------+--------------
(S, G)                      0
(*, G)                      0

IPv6 Multicast FIB SC summary on Slot[4]:
Shortcut Type               Shortcut count
----------------------------+--------------
(*, G/128)                  0
(*, G/m)                    3
```

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 multicast ipv6 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 multicast ipv6 shared-adjacencies module 4

---- SLOT [4] ----

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      0            0
StarG (spt == INF) adjacency  0xB      0            0
StarG (spt != INF) adjacency  0xC      0            0
Router
```

CHAPTER 28

Configuring IPv4 Multicast Layer 3 Switching

This chapter describes how to configure IPv4 multicast Layer 3 switching on the Cisco 7600 series routers.

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

- The Cisco 7600 Series Routers Command References at this URL:
  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- The Release 12.2 publications at this URL:
  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/index.htm

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 (PFC) provides Layer 3 switching for IP multicast flows using the hardware replication table and hardware Cisco Express Forwarding (CEF), which uses the forwarding information base (FIB) and the adjacency table on the PFC. In systems with Distributed Forwarding Cards (DFCs), IP multicast flows are Layer 3 switched locally using Multicast Distributed Hardware Switching (MDHS). MDHS uses local hardware CEF and replication tables on each DFC to perform Layer 3 switching and rate limiting of reverse path forwarding (RPF) failures locally on each DFC-equipped switching module.

The PFC and the DFCs support hardware switching of (*,G) state flows. The PFC and the DFCs support rate limiting of non-RPF traffic.

Also termed as hardware switching, 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 and mcast rate-limiters limit the traffic relayed to the route processor.

The PFC and the DFCs all use the Layer 2 multicast forwarding table to determine on which ports Layer 2 multicast traffic should be forwarded (if any). The multicast forwarding table entries are populated in conjunction with Internet Group Management Protocol (IGMP) snooping (see Chapter 30, “Configuring IGMP Snooping for IPv4 Multicast Traffic”).

Current implementation of IPV4 multicast in 7600 uses the platform specific distribution mechanism from Route Processor (RP) to Switch Processor (SP). With the introduction of MFIB, MFIB provides support for distribution of the information in a platform independent way to the Switch Processor (SP) and Line cards (LC’s). In 12.2(33)SRE, this feature is supported on SUP720, Sup32, RSP720 and compatible DFCs.

For more information on the MDSS (Multicast Distributed Switching Services) implementation used prior to MFIB implementation, see
http://www.cisco.com/en/US/docs/ios/12_2/switch/configuration/guide/xcdmdc.htm

Multicast Layer 3 Switching Cache

This section describes how the PFC and the DFCs maintain Layer 3 switching information in hardware tables.

The PFC and DFC populate the (S,G) or (*,G) flows in the hardware FIB table with the appropriate masks; for example, (S/32, G/32) and (*/0, G/32). The RPF interface and the adjacency pointer information is also stored in each entry. The adjacency table contains the rewrite information and pointers to the multicast expansion table (MET) table. 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 MSFC updates its multicast routing table and forwards the new information to the PFC whenever it receives traffic for a new flow. In addition, if an entry in the multicast routing table on the MSFC ages out, the MSFC deletes the entry and forwards the updated information to the PFC. In systems with DFCs, flows are populated symmetrically on all DFCs and on the PFC.

The Layer 3 switching cache contains flow information for all active Layer 3-switched flows. After the switching cache is populated, multicast packets identified as belonging to an existing flow can be Layer 3 switched based on the cache entry for that flow. For each cache entry, the PFC maintains a list of outgoing interfaces for the IP multicast group. From this list, the PFC determines onto which VLANs traffic from a given multicast flow should be replicated.

These commands affect the Layer 3 switching cache entries:

- When you clear the multicast routing table using the **clear ip mroute** command, all multicast Layer 3 switching cache entries are cleared.
- When you disable IP multicast routing on the MSFC using the **no ip multicast-routing** command, all multicast Layer 3 switching cache entries on the PFC are purged.

Layer 3-Switched Multicast Packet Rewrite

When a multicast packet is Layer 3 switched from a multicast source to a destination multicast group, the PFC and the DFCs perform a packet rewrite that is based on information learned from the MSFC and stored in the adjacency table.

For example, Server A sends a multicast packet addressed to IP multicast group G1. If there are members of group G1 on VLANs other than the source VLAN, the PFC must perform a packet rewrite when it replicates the traffic to the other VLANs (the router also bridges the packet in the source VLAN).

When the PFC receives the multicast packet, it is (conceptually) formatted as follows:

Layer 2 Frame Header		**Layer 3 IP Header**				**Data**	**FCS**
Destination	Source	Destination	Source	TTL	Checksum		
Group G1 MAC[1]	*Source A MAC*	*Group G1 IP*	*Source A IP*	*n*	*calculation1*		

1. In this example, Destination B is a member of Group G1.

The PFC rewrites the packet as follows:

- Changes the source MAC address in the Layer 2 frame header from the MAC address of the host to the MAC address of the MSFC (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.)
- Decrements the IP header Time to Live (TTL) by one and recalculates the IP header checksum

The result is a rewritten IP multicast packet that appears to have been routed. The PFC replicates the rewritten packet onto the appropriate destination VLANs, where it is forwarded to members of IP multicast group G1.

After the PFC performs the packet rewrite, the packet is (conceptually) formatted as follows:

Frame Header		**IP Header**				**Data**	**FCS**
Destination	Source	Destination	Source	TTL	Checksum		
Group G1 MAC	*MSFC 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 hardware switched and at least one outgoing interface is not hardware switched, that flow is considered partially switched. When a partially switched flow is created, all multicast traffic belonging to that flow still reaches the MSFC and is forwarded by software on those outgoing interfaces that are not hardware switched.

These sections describe partially and completely switched flow:

Partially Switched Flows

A flow might be partially switched instead of completely switched in these situations:

- If the router 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 router is the first-hop router to the source in PIM sparse mode (in this case, the router 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 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 PFC prevents multicast traffic bridged on the source VLAN for that flow from reaching the MSFC interface in that VLAN, freeing the MSFC of the forwarding and replication load for that flow.

One consequence of a completely switched flow is that multicast statistics on a per-packet basis for that flow cannot be recorded. Therefore, the PFC periodically sends multicast packet and byte count statistics for all completely switched flows to the MSFC. The MSFC 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

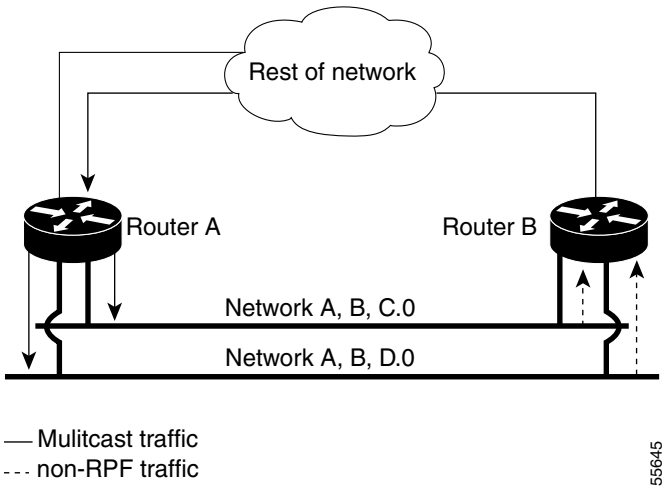
These sections describe non-RPF traffic processing:

Non-RPF Traffic Overview

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 28-1). 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 Cisco 7600 series router processes non-RPF traffic in hardware on the PFC by filtering (dropping) or rate limiting the non-RPF traffic.

Figure 28-1 Redundant Multicast Router Configuration in a Stub Network



Filtering of RPF Failures for Stub Networks

The PFC and the DFCs support ACL-based filtering of RPF failures for sparse mode stub networks. When you enable the ACL-based method of filtering RPF failures by entering the **mls ip multicast stub** command on the redundant router, the following ACLs automatically download to the PFC and are applied to the interface you specify:

```
access-list 100 permit ip A.B.C.0 0.0.0.255 any
access-list 100 permit ip A.B.D.0 0.0.0.255 any
access-list 100 permit ip any 224.0.0.0 0.0.0.255
access-list 100 permit ip any 224.0.1.0 0.0.0.255
access-list 100 deny ip any 224.0.0.0 15.255.255.255
```

The ACLs filter RPF failures and drop them in hardware so that they are not forwarded to the router.

Use the ACL-based method of filtering RPF failures only in sparse mode stub networks where there are no downstream routers. For dense mode groups, RPF failure packets have to be seen on the router for the PIM assert mechanism to function properly. Use CEF-based or NetFlow-based rate limiting to limit the rate of RPF failures in dense mode networks and sparse mode transit networks.

For information on configuring ACL-based filtering of RPF failures, see the "Configuring ACL-Based Filtering of RPF Failures" section on page 28-15.

Rate Limiting of RPF Failure Traffic

When you enable rate limiting of packets that fail the RPF check (non-RPF packets), most non-RPF packets are dropped in hardware. According to the multicast protocol specification, the router needs to receive the non-RPF packets for the PIM assert mechanism to function properly, so all non-RPF packets cannot be dropped in hardware.

When a non-RPF packet is received, a NetFlow entry is created for each non-RPF flow.

When the first non-RPF packet arrives, the PFC bridges the packet to the MSFC 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 MSFC.

To support the PIM assert mechanism, the PFC periodically forwards a percentage of the non-RPF flow packets to the MSFC.

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 enabled by default.

Understanding How IPv4 Bidirectional PIM Works

The PFC3 supports hardware forwarding of IPv4 bidirectional PIM groups. To support IPv4 bidirectional PIM groups, the PFC3 implements a new mode called designated forwarder (DF) mode. The designated forwarder is the router elected to forward packets to and from a segment for a IPv4 bidirectional PIM group. In DF mode, the 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 28-22.

Default IPv4 Multicast Layer 3 Switching Configuration

Table 28-1 shows the default IP multicast Layer 3 switching configuration.

Table 28-1 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

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 30, "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 225.0.0.* through 239.0.0.* and 224.128.0.* through 239.128.0.*.

  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 router is the RP for the group.
- For IPv4 bidirectional PIM entries when a DF interface or RPF interface is a tunnel.
- Supervisor Engine 32 does not support egress multicast replication and cannot detect the multicast replication mode.
- In a MFIB implementation, **ip multicast rate-limit** command that limits the number of data packets in either direction is not supported in hardware configurations.
- In a MFIB implementation, **ip multicast ttl-threshold** command is not supported in hardware configurations.
- In a MFIB implementation, Network Address Translation (NAT) is not supported in hardware configurations.
- Following MDSS commands are invalid after MFIB IPv4 implementation:
  - **debug mdss [vrf <vrf-name>] [all | error | events | mdt | p2p | packet]**
  - **mls ip multicast [vrf <name>] connected {config command - global and interface-level}**
  - **mls ip multicast consistency-check {config command - global and interface-level}**
  - **show mls ip multicast consistency-check**
  - **show mls ip multicast rp-mapping**
- Following commands are deprecated post MFIB implementation:
  - **mls ip multicast non-rpf aging fast**
  - **mls ip multicast non-rpf aging global**
  - **ip multicast replication-mode egress**
  - **mls ip multicast replication-mode ingress**
  - **mls ip multicast flow-stat timer**

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/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcpt3/1cfssm.htm

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/iproute/command/reference/fiprrp_r.html
- *Cisco IOS IP and IP Routing Command Reference*, Release 12.1, at this URL:

  http://www.cisco.com/en/US/docs/ios/12_1/iproute/command/reference/ip_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 platform software multicast ip`	Displays brief information about the packet flows in the system.

This example shows how to globally enable hardware switching of multicast routes:

```
Router(config)# mls ip multicast
```

```
Router(config)# show platform software multicast ip
(40.0.0.2, 232.0.1.4) Incoming interface: Lspvif0, Packets Switched: 119954142
Hardware switched outgoing interfaces:
GigabitEthernet3/6
Total hardware switched flows: 1
```

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 28-10.

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. **Note** Starting 12.2 (33) SRE release, the **mls ip multicast** command is changed to **ip multicast hardware-switching.**
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)#
```

Configuring the Replication Mode

Note Supervisor Engine 32 supports only ingress replication mode.

The Supervisor Engine 720 supports the **egress** keyword. Support for the **egress** keyword is called "Multicast Enhancement - Replication Mode Detection" in the release notes and Feature Navigator.

By default, a Supervisor Engine 720 automatically detects the replication mode based on the module types installed in the system. If all modules are capable of egress replication, the system uses egress-replication mode. If the supervisor engine detects modules that are not capable of egress replication, the replication mode automatically changes to ingress replication. You can override this action by entering the **ip multicast hardware-switching replication-mode egress** command so that the system continues to work in egress-replication mode even if there are fabric-enabled modules installed that do not support egress replication (for example, OSMs). You can also configure the system to operate only in ingress-replication mode.

If the system is functioning in automatic detection mode, and you install a module that cannot perform egress replication, the following occurs:

- The system reverts to ingress mode
- A system log is generated
- A system reload occurs to revert to the old configuration

If the system is functioning in forced egress mode, a system log is created that will display the presence of modules that are not capable of egress replication mode.

Note If you configure forced egress mode in a system that has fabric-enabled modules that are not capable of egress replication, you must make sure that these modules are not sourcing or receiving multicast traffic.

During a change from egress- to ingress-replication mode, traffic interruptions may occur because the shortcuts will be purged and reinstalled. To avoid interruptions in traffic forwarding, enter the **ip multicast hardware-switching replication-mode ingress** command in global configuration mode. This command forces the system to operate in ingress-replication mode.

The **no** form of the **ip multicast hardware-switching replication-mode ingress** command restores the system to auomatic detection mode.

To enable IP multicast Layer 3 switching, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip multicast hardware-switching replication-mode [egress | ingress]`	Specifies the replication mode.
Step 2	`Router# show platform software multicast ip capability`	Displays the configured replication mode. **Note** Starting 12.2 (33) SRE release, the **show mls ip multicast capability** command is changed to **show platform software multicast ip capability.**
Step 3	`Router# show platform software multicast ip summary`	Displays the replication mode and if automatic detection is enabled or disabled.

This example shows how to enable the replication mode:

```
Router (config)# ip multicast hardware-switching replication-mode egress
Router# show platform software multicast ip capability
Current System HW Replication Mode : Egress
Auto-detection of Replication Mode : ON

Slot Replication-Capability Replication-Mode
   2 Egress                 Egress
```

```
   3 Egress                  Egress
   4 Egress                  Egress
   6 Egress                  Egress
Router#

Router# show platform software multicast ip summary

IPv6 Multicast Netflow SC summary on Slot[7]:
Shortcut Type               Shortcut count
--------------------------+--------------
(S, G)                      0

IPv6 Multicast FIB SC summary on Slot[7]:
Shortcut Type               Shortcut count
--------------------------+--------------
(*, G/128)                  0
(*, G/m)                    0
Router (config)#
```

Enabling Local Egress Replication

Note Supervisor Engine 32 supports only ingress replication mode.

With a Supervisor Engine 720, you can unconditionally enable local egress replication. This feature is called “Multicast enhancement - egress replication performance improvement” in the release notes and Feature Navigator.

DFC-equipped modules with dual switch-fabric connections host two packet replication engines, one per fabric connection. Each replication engine is responsible for forwarding packets to and from the interfaces associated with the switch-fabric connections. The interfaces that are associated with a switch-fabric connection are considered to be “local” from the perspective of the packet replication engine.

You can prevent redundant replication of multicast packets across the switch-fabric connection by entering a command that instructs the two replication engines on these modules to forward packets only to local interfaces which are associated with the switch-fabric connection that the replication engine supports.

When you enable this feature, the multicast expansion table (MET) for each replication engine is populated with the local Layer 3 interfaces only. This action prevents replication for interfaces that are not supported by the replication engine (nonlocal interfaces) and increases replication performance.

Local egress replication is supported with the following software configuration and hardware:

- IPv4 egress replication mode
- Dual fabric-connection DFC-equipped modules
- Layer 3-routed interfaces and subinterfaces that are not part of a port channel

The local egress replication feature is not supported for the following internal VLANs:

- Egress internal VLAN
- Partial-shortcut internal VLAN
- Internal VLAN for Multicast VPN Multicast Distribution Tree (MDT) tunnel
- Point-to-point tunnel internal VLAN
- QoS internal VLAN

Note The local egress replication feature is not supported with IPv6 multicast or in a system that has a mix of IPv4 and IPv6 multicast enabled.

To enable local egress replication, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls ip multicast egress local`	Enables local egress replication. **Note** This command requires a system reset for the configuration to take effect.
Step 2	`Router # reload`	Reloads the system.
Step 3	`Router# show platform software multicast ip capability`	Displays the configured replication mode.

This example shows how to enable local egress replication:

```
Router (config)# mls ip multicast egress local
Router (config)# exit
Router # reload
Router # show platform software multicast ip capability
Current System HW Replication Mode : Egress
Auto-detection of Replication Mode : ON

Slot Replication-Capability Replication-Mode
   2 Egress                 Egress
   3 Egress                 Egress
   4 Egress                 Egress
   6 Egress                 Egress
```

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 MSFC. 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 platform software multicast ip 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 MSFC 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 MSFC.

To specify how often flow statistics messages forwarded from the supervisor engine to the MSFC, 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 MSFC.
`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 MSFC every 10 seconds:

```
Router(config)# mls ip multicast flow-stat-timer 10
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**

Validating the Rate-Limiter Status

To validate the rater-limiter status, perform this task:

Command	Purpose
`Router# show mls rate-limit | i RPF`	Displays RPF failure rate-limiting information with the current state of the rate limiter.

This example shows how to display RPF failure rate-limiting information:

```
Router# show mls rate-limit | i RPF
IP RPF FAILURE         100      10
Router-1# sh mls rate-limit usage
                            Rate Limiter Type     Packets/s   Burst
                          ----------------------   ---------   -----
Layer3 Rate Limiters:
             RL# 0: Free                          -           -       -
             RL# 1: Free                          -           -       -
             RL# 2: Free                          -           -       -
             RL# 3: Free                          -           -       -
             RL# 4: Free                          -           -       -
             RL# 5: Used
                                  MCAST DFLT ADJ          10      10
             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-1# sh 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                10      10  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               -       -     -
      DHCP Snooping IN   Off               -       -     -
     DHCP Snooping OUT   Off               -       -     -
        ARP Inspection   Off               -       -     -
       LAYER_2 PORTSEC   Off               -       -     -
     LAYER_2 MiniProto   Off               -       -     -
             IP ERRORS   On                100      10  Group:0 S
           CAPTURE PKT   Off               -       -     -
            MCAST IGMP   Off               -       -     -
 MCAST IPv6 DIRECT CON   Off               -       -     -
 MCAST IPv6 ROUTE CNTL   Off               -       -     -
 MCAST IPv6 *G M BRIDG   Off               -       -     -
  MCAST IPv6 SG BRIDGE   Off               -       -     -
  MCAST IPv6 DFLT DROP   Off               -       -     -
 MCAST IPv6 SECOND. DR   Off               -       -     -
  MCAST IPv6 *G BRIDGE   Off               -       -     -
        MCAST IPv6 MLD   Off               -       -     -
  IP ADMIS. ON L2 PORT   Off               -       -     -
        LAYER_2 MACSEC   Off               -       -     -
        MCAST IPv4 PIM   Off               -       -     -
Router-1#
```

Displaying IPv4 Multicast Layer 3 Hardware Switching Summary

Note The **show interface statistics** command does not display hardware-switched packets, only packets switched by software.

The **show ip pim interface count** command displays the IP multicast Layer 3 switching enable state on IP PIM interfaces and the number of packets received and sent on the interface.

To display IP multicast Layer 3 switching information for an IP PIM Layer 3 interface, perform one of these tasks:

Command	Purpose
`Router# show ip pim interface [{{vlan vlan_ID} | {type[1] slot/port} | {port-channel number}}] count`	Displays IP multicast Layer 3 switching enable state information for all MSFC 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 platform software multicast ip** 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 platform software multicast ip group** *group-id*	Displays IP multicast Layer 3 switching group information.
Router# **show platform software multicast ip interface** [**gigabitethernet** *1-6* | **port-channel** *1-256* | **tengigabitethernet** *1-6* | **vlan** *1-4094*] | **source** *A.B.C.D.*]	Displays IP multicast Layer 3 switching details for all interfaces.
Router# **show platform software multicast ip source** *source-ip*	Displays IP multicast Layer 3 switching source information.
Router# **show platform software multicast ip summary**	Displays a summary of IP multicast Layer 3 switching information.
Router# **show platform software multicast ip statistics** [**group** *group-id*]	Displays IP multicast Layer 3 switching statistics.

This example shows how to display information on a specific IP multicast Layer 3 switching entry:

```
Router# show platform software multicast ip group 232.0.1.4
Multicast hardware switched flows:

(40.0.0.2, 232.0.1.4) Incoming interface: GigabitEthernet3/2/1, Packets Switched: 8069027
Hardware switched outgoing interfaces:
     Tunnel10

Total hardware switched flows: 1

PE1-7600
```

This example shows how to display IP multicast group information:

```
Router# show platform software multicast ip source 40.0.0.2
Multicast hardware switched flows:

(40.0.0.2, 232.0.1.4) Incoming interface: GigabitEthernet3/2/1, Packets Switched: 8778143
Hardware switched outgoing interfaces:
     Tunnel10

Total hardware switched flows: 1

Router#
```

This example shows how to display IP multicast Layer 3 switching information for gigabitethernet interface 3/2/1:

```
Router# show platform software multicast ip interface gigabitethernet 3/2/1
Multicast hardware switched flows:

(40.0.0.2, 232.0.1.4) Incoming interface: GigabitEthernet3/2/1, Packets Switched: 8206582
Hardware switched outgoing interfaces:
     Tunnel10

Total hardware switched flows: 1
```

This example shows how to display the IP multicast Layer 3 switching statistics:

```
Router# show platform software multicast ip statistics group 232.0.1.4

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

To enable IPv4 bidirectional PIM, perform this task:

Command	Purpose
`Router(config)# ip pim bidir-enable`	Enables IPv4 bidirectional PIM globally on the router.
`Router(config)# no ip pim bidir-enable`	Disables IPv4 bidirectional PIM globally on the router.

This example shows how to enable IPv4 bidirectional PIM on the router:

```
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. **Note** Starting 12.2 (33) SRE release, the **mls ip multicast** command is changed to **ip multicast hardware-switching.**

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 bidir**	Displays IPv4 bidirectional PIM information.
Router# **show ip mroute**	Displays information about the mutlicast 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 platform software multicast ip group 232.0.1.4
Multicast hardware switched flows:

(40.0.0.2, 232.0.1.4) Incoming interface: Lspvif0, Packets Switched: 120181613
Hardware switched outgoing interfaces:
GigabitEthernet3/6

Total hardware switched flows: 1
```

Troubleshooting

This section describes how to troubleshoot common IP multicast Layer 3 switching issues.

Scenarios/Problems	Solution

How do I know the IP multicast Layer 3 switching events?	Use the **debug ip multicast hardware-switching** command. This example shows output from the command using the **error** keyword:
	`Router# debug ip multicast hardware-switching error 232.0.1.4`
	`PE1-7600#debug ip multicast hardware-switching error 232.0.1.4`
	`CMFIB-RP IPv4  error debugging enabled for group 232.0.1.4`
	`PE1-7600#`
	This example shows output from the command using the **event** keyword:
	`Router# debug ip multicast hardware-switching event 232.0.1.4`
	`CMFIB-RP IPv4  event debugging enabled for group 232.0.1.4`
	This example shows output from the command using the **ha-event** keyword:
	`Router# debug ip multicast hardware-switching ha-event 232.0.1.4`
	`CMFIB-RP IPv4  ha event debugging enabled for group 232.0.1.4`
	`PE1-7600#`
	This example shows output from the command using the **ha-error** keyword:
	`Router# debug ip multicast hardware-switching ha-error 232.0.1.4`
	`CMFIB-RP IPv4  ha error debugging enabled for group 232.0.1.4`

How do I check the log events, packet information, and assert events?	Use the **debug platform software multicast** command. This example shows the output from the command using the **assert** keyword:
	`PE-3-sp#debug platform software multicast assert`
	`Assertion for Layer 2 multicast debugging is on`
	`PE-3-sp#`
	`PE-3-sp#debug platform software multicast ha l2-sso all`
	`Debug for mcast SSO all debugging is on`
	`PE-3-sp#debug platform software multicast ha l2-sso err`
	`PE-3-sp#debug platform software multicast ha l2-sso error`
	`Debug for mcast SSO error debugging is on`
	`PE-3-sp#debug platform software multicast ha l2-sso eve`
	`PE-3-sp#debug platform software multicast ha l2-sso event`
	`Debug for mcast SSO events debugging is on`
	`PE-3-sp#debug platform software multicast ha l2-sso pak`
	`PE-3-sp#debug platform software multicast ha l2-sso pak`
	`Debug for mcast SSO packets debugging is on`
	`PE-3-sp#`

How do I verify the layer 2 high availability multicast shortcuts debugging errors, events, and packet information?	Use the **debug platform software multicast ha l2-sso** command.
	`PE-3-sp#debug platform software multicast ha l2-sso`
	`ha l2-sso for Layer 2 multicast debugging is on`

How do I display the layer 2 line card multicast events?	Use the **debug platform software multicast lc** command. This example shows output from the command:
	`PE-3-sp#debug platform software multicast lc`
	`Debug from mls_mcast_lc library debugging is on`

Scenarios/Problems	Solution

How do I check the CGMP debugging event and packet information?	Use the **debug platform software multicast cgmp** command. This example shows output from command using the **event** keyword:
	`PE-3-sp#debug platform software multicast cgmp event`
	`Router Discovery (CGMP Protocol) event log debugging is on`
	This example shows output from the command using the **pak** keyword:
	`PE-3-sp#debug platform software multicast cgmp pak`
	`Router Discovery (CGMP Protocol) packet log debugging is on`

How do I check the multicast hal error, event, timer, and packet information?	Use the **debug platform software multicast ip hal** command. This example shows output from the command using the **event** keyword:
	`PE-3-sp#debug platform software multicast ip  hal event`
	`Multicast HAL event log debugging is on`
	`PE-3-sp#`
	`*Oct 30 09:24:48.078 EDT: SP: hal_timer_event: NRPF-AG`
	`*Oct 30 09:24:48.790 EDT: SP: hal_timer_event: S-CHECK`
	`*Oct 30 09:24:49.754 EDT: SP: hal_timer_event: NRPF-AG`
	`*Oct 30 09:24:51.530 EDT: SP: hal_timer_event: NRPF-AG`
	`*Oct 30 09:24:53.298 EDT: SP: hal_timer_event: NRPF-AG`
	`*Oct 30 09:24:55.154 EDT: SP: hal_timer_event: NRPF-AG`

How do I display the IGMP debugging events and packet information?	Use the **debug platform software multicast igmp** command. This example shows output from the command using the **pak** keyword: `PE-3-sp#debug platform software multicast igmp pak` `IGMP snooping packet log debugging is on` `PE-3-sp#` `*Oct 30 09:26:22.143 EDT: SP: RELAYED PAK to index 0x0008440B, vlan 1035` `*Oct 30 09:26:22.143 EDT: SP: Packet dump:` `18000070:              0100 5E000016 00000E00        ..^.......` `18000080: 02000800 45000028 00000000 400254BC  ....E..(....@.T<` `18000090: 46000002 E0000016 2200CBF6 00000001  F...`...".Kv....` `180000A0: 01000001 E8000104 28000002 00010203 ....h...(.......` `180000B0: 04058C`

How do I check the events and packet information for MLD debugging?	Use the **debug platform software multicast mld** command. This example shows output from the command using the event keyword:
	`PE-3-sp#debug platform software multicast mld event`
	`multicast snooping event log debugging is on`

How do I check the multicast router events and packet information?	Use the **debug platform software multicast mrouter** command. This example shows output from the command using the **event** keyword:
	`PE-3-sp#debug platform software multicast mrouter event`
	`Router Discovery (MLD MROUTER Protocol) event log debugging is on`
	This example shows output from the command using the **pak** keyword:
	`PE-3-sp#debug platform software multicast mrouter pak`
	`Router Discovery (MLD MROUTER Protocol) packet log debugging is on`

How do I check the multicast shortcut debugging information?	Use the **debug platform software multicast msc** command. This example shows output from the command using the **error** keyword:
	`PE-3-sp#debug platform software multicast msc error`
	`Multicast Shortcuts error log debugging is on`

Scenarios/Problems	Solution

How do I check the multicast RGMP debugging information?	Use the **debug platform software multicast rgmp** command. This example shows output from the command using the **event** keyword:
	`PE-3-sp#debug platform software multicast rgmp event`
	`RGMP event log debugging is on`

How do I display the multicast bidirectional df debugging information?	Use the **debug platform software multicast rpdf** command. This example shows output from the command using the **error** keyword:
	`PE-3-sp#debug platform software multicast rpdf error`
	`Multicast Shortcuts error log debugging is on`

How do I display the multicast titan debugging information?	Use the **debug platform software multicast titan** command. This example shows output from the command using the error keyword:
	`PE-3-sp#debug platform software multicast titan error`
	`Multicast Bidir RP/DF error log debugging is on`

How do I display the MFIB IPv6 platform code debugging and multicast HAL IPv6 debug command information?	Use the **debug platform software multicast ipv6** command. This example shows output from the command using the **error, eve,** and **stats** keywords:
	`PE-3-sp#debug platform software multicast ipv6 cmfib error`
	`CMFIB-LC IPv6 error debugging enabled`
	`PE-3-sp#debug platform software multicast ipv6 cmfib eve`
	`CMFIB-LC IPv6 event debugging enabled`
	`PE-3-sp#debug platform software multicast ipv6 cmfib stats`
	`CMFIB-LC IPv6 stats debugging enabled`

How do I display the multicast ip cmfib errors, shortcut events, and export the hardware statistics information?	Use the **debug platform software multicast ip cmfib** command. This example shows output from the command using the **error** keyword:
	`PE-3-sp#debug platform software multicast ip cmfib error`
	`CMFIB-LC IPv6 error debugging enabled`

How do I display the source or group IP address and the mfib IPv4 pending entry error information?	Use the **debug platform software multicast ip cmfib error** command. This example shows output from command:
	`PE-3-sp#debug platform software multicast ip cmfib error 232.0.1.4 verbose`
	`CMFIB-LC IPv4 verbose error debugging enabled for group 232.0.1.4`

How do I display the source or group IP address, multicast forwarding information base (mfib) IPv4 control(ctrl) entries events, mfib hw-api events, mfib IPv4 table events, mfib IPv4 pending entry events, and mfib IPv4 table events?	Use the **debug platform software multicast ip cmfib event** command. This example shows output from the command using the **ctrl** keyword:
	`PE-3-sp#debug platform software multicast ip cmfib event ctrl`
	`CMFIB-LC IPv4 event control debugging enabled`

Scenarios/Problems	Solution

How do I check the HSRP IPv4 multicast address on an interface?	Use the **show ip interface** command. This example shows sample output from the command:
	`router#`**`show ip interface ethernet0/0`**
	`Ethernet0/0 is up, line protocol is up`
	`  Internet address is 10.0.0.1/8`
	`  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: 224.0.0.2` **`<<< HSRP multicast IP address`**
	`  Outgoing access list is not set`
	`  Inbound  access list is not set`
	`  Proxy ARP is enabled`
	`  Local Proxy ARP is disabled`
	`  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 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`
	`  Policy routing is disabled`
	`  Network address translation is disabled`
	`  BGP Policy Mapping is disabled`
	`  Input features: MCI Check`
	`  WCCP Redirect outbound is disabled`
	`  WCCP Redirect inbound is disabled`
	`  WCCP Redirect exclude is disabled`

Scenarios/Problems	Solution
How do I check the HSRP multicast MAC address in an interface controller?	Use the **show controllers** command. This example shows a sample output from the command:

```
Router#show controllers e0/0
Interface Ethernet0/0
Hardware is AMD Unknown
ADDR: 266A5F8, FASTSEND: 18618, MCI_INDEX: 0
DIST ROUTE ENABLED: 0
Route Cache Flag: 11
 amdp2_instance=0x266C498, registers=0x266A5A0, ib=0x2677FF8
 rx ring entries=32, tx ring entries=64
 rxring=0x2678048, rxr shadow=0x2678280, rx_head=26, rx_tail=0
 txring=0x2678338, txr shadow=0x2678770, tx_head=5, tx_tail=5, tx_count=0
 running=1, port id=0x214CD58
 Software MAC address filter(hash:length/addr/mask/hits):
  0x00:  0  ffff.ffff.ffff  0000.0000.0000         0
  0x0D:  0  0000.0c07.ac01  0000.0000.0000         0 <<< HSRP Virtual MAC
address (group 1)
  0x5C:  0  0100.5e00.0002  0000.0000.0000      2265    <<<  HSRP multicast
MAC address
  0xC0:  0  0100.0ccc.cccc  0000.0000.0000        93
  0xC0:  1  0180.c200.0002  0000.0000.0000         0
  0xC5:  0  0180.c200.0007  0000.0000.0000         0
  0xCC:  0  aabb.cc00.fa00  0000.0000.0000         0
 spurious_idon=0, filtered_pak=569, throttled=0, enabled=0, disabled=0
 rx_framing_err=0, rx_overflow_err=0, rx_buffer_err=0
 rx_bpe_err=0, rx_soft_overflow_err=0, rx_no_enp=0, rx_discard=0
 tx_one_col_err=0, tx_more_col_err=0, tx_no_enp=0, tx_deferred_err=0
 tx_underrun_err=0, tx_late_collision_err=0, tx_loss_carrier_err=0
 tx_exc_collision_err=0, tx_buff_err=0, fatal_tx_err=0
 hsrp_conf=1, need_af_check=1
 tx_limited=0(64)
```

CHAPTER 29

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 Cisco 7600 series routers. MLDv2 snooping is supported on all versions of the PFC3.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- To constrain IPv4 Multicast traffic, see Chapter 30, "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 Cisco 7600 series routers to examine MLDv2 packets and make forwarding decisions based on their content.

You can configure the router 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 router 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 29-8.

MLDv2 (on a multicast router) or the MLDv2 snooping querier (on the supervisor engine) sends out periodic general MLDv2 queries that the router forwards through all ports in the VLAN, and to which hosts respond. MLDv2 snooping monitors the Layer 3 MLDv2 traffic.

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

MLDv2 uses these 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 router 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 router 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 router forwards general queries from IPv6 multicast routers to all ports in a VLAN). The router snoops these reports.

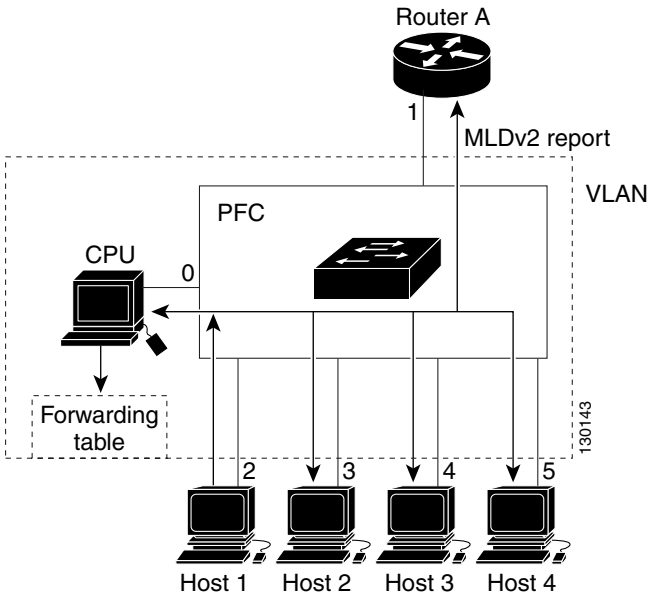
In response to a snooped MLDv2 report, the router creates an entry in its Layer 2 forwarding table for the VLAN on which the report was received. When other hosts that are interested in this multicast traffic send MLDv2 reports, the router snoops their reports and adds them to the existing Layer 2 forwarding table entry. The router 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 router forwards multicast traffic for the multicast group specified in the report to the interfaces where reports were received (see Figure 29-1).

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 29-1 Initial MLDv2 Listener Report



Multicast router A sends an MLDv2 general query to the router, 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 router snoops the MLDv2 report multicast by Host 1, the router uses the information in the MLDv2 report to create a forwarding-table entry, as shown in Table 29-1, that includes the port numbers of Host 1, the multicast router, and the router.

Table 29-1 MLDv2 Snooping Forwarding Table

Destination MAC Address	Type of Packet	Ports
0100.5exx.xxxx	MLDv2	0
0100.5e01.0203	!MLDv2	1, 2

The router hardware can distinguish MLDv2 information packets from other packets for the multicast group. The first entry in the table tells the router to send only MLDv2 packets to the CPU. This prevents the router from becoming overloaded with multicast frames. The second entry tells the router to send frames addressed to the 0x0100.5E01.0203 multicast MAC address that are not MLDv2 packets (!MLDv2) 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 29-2), the router snoops that message and adds the port number of Host 4 to the forwarding table as shown in Table 29-2. Because the forwarding table directs MLDv2 messages only to the router, the message is not flooded to other ports. Any known multicast traffic is forwarded to the group and not to the router.

Figure 29-2 Second Host Joining a Multicast Group

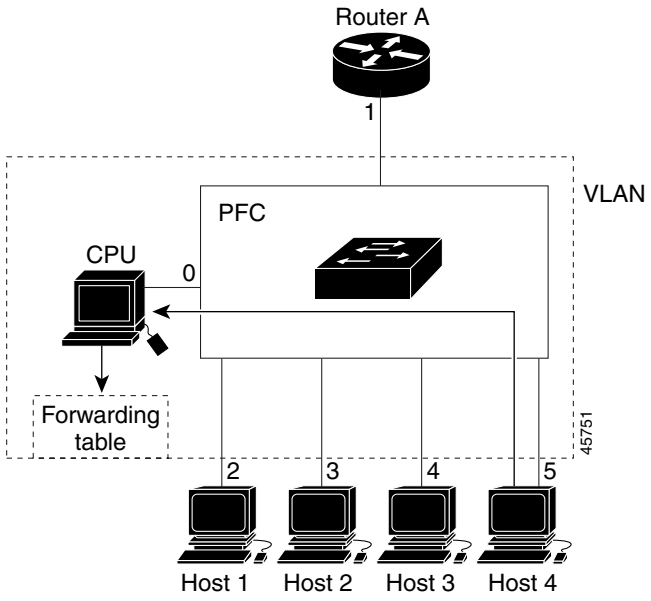


Table 29-2 Updated MLDv2 Snooping Forwarding Table

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

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 router 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 router 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 router 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 router 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 router 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 router 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 Cisco 7600 series routers 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 router as the MLDv2 snooping querier.

You can configure a router to generate MLDv2 queries on a VLAN regardless of whether or not IP multicast routing is enabled.

Default MLDv2 Snooping Configuration

Table 29-3 shows the default MLDv2 snooping configuration.

Table 29-3 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 14, "Configuring VLANs").
- Configure an IPv6 address on the VLAN interface (see Chapter 21, "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 Cisco 7600 series routers in the VLAN that support it. One router 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 29-8).

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 router 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 router, 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 router. 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`	Enables fast-leave processing in the VLAN.
	`Router(config-if)# no ipv6 mld snooping fast-leave`	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`	Enables SSM safe reporting.
	`Router(config-if)# no ipv6 mld snooping ssm-safe-reporting`	Clears the configuration.

This example shows how to SSM safe reporting:

```
Router(config)# interface vlan 10
Router(config-if)# ipv6 mld snooping ssm-safe-reporting
```

Configuring Explicit Host Tracking

Note Disabling explicit host tracking disables fast-leave processing and proxy reporting.

To enable explicit host tracking on a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ipv6 mld snooping explicit-tracking`	Enables explicit host tracking.
	`Router(config-if)# no ipv6 mld snooping explicit-tracking`	Clears the explicit host tracking configuration.
Step 3	`Router# show ipv6 mld snooping explicit-tracking vlan vlan_ID`	Displays the status of explicit host tracking.

This example shows how to enable explicit host tracking:

```
Router(config)# interface vlan 25
Router(config-if)# ipv6 mld snooping explicit-tracking
Router(config-if)# end
Router# show ipv6 mld snooping explicit-tracking vlan 25
Source/Group                    Interface    Reporter        Filter_mode
--------------------------------------------------------------------------
10.1.1.1/226.2.2.2              Vl25:1/2     16.27.2.3       INCLUDE
10.2.2.2/226.2.2.2              Vl25:1/2     16.27.2.3       INCLUDE
```

Configuring Report Suppression

To enable report suppression on a VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface vlan vlan_ID`	Selects a VLAN interface.
Step 2	`Router(config-if)# ipv6 mld snooping report-suppression`	Enables report suppression.
	`Router(config-if)# no ipv6 mld snooping report-suppression`	Clears the report suppression configuration.
Step 3	`Router# show ipv6 mld interface vlan_ID | include report-suppression`	Displays the status of report suppression.

This example shows how to enable explicit host tracking:

```
Router(config)# interface vlan 25
Router(config-if)# ipv6 mld snooping report-suppression
Router(config-if)# end
Router# Router# show ipv6 mld interface vlan 25 | include report-suppression
  MLD snooping report-suppression is enabled
```

Displaying MLDv6 Snooping Information

These sections describe displaying MLDv6 snooping information:

Displaying Multicast Router Interfaces

When you enable IGMP snooping, the router 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 30

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 Cisco 7600 series routers.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- To constrain IPv6 Multicast traffic, see Chapter 29, "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 router 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 28, "Configuring IPv4 Multicast Layer 3 Switching."

You can configure the IGMP snooping querier on the router 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 30-9.

IGMP (on a multicast router) or the IGMP snooping querier (on the supervisor engine) sends out periodic general IGMP queries that the router 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 router forwards general queries from multicast routers to all ports in a VLAN).

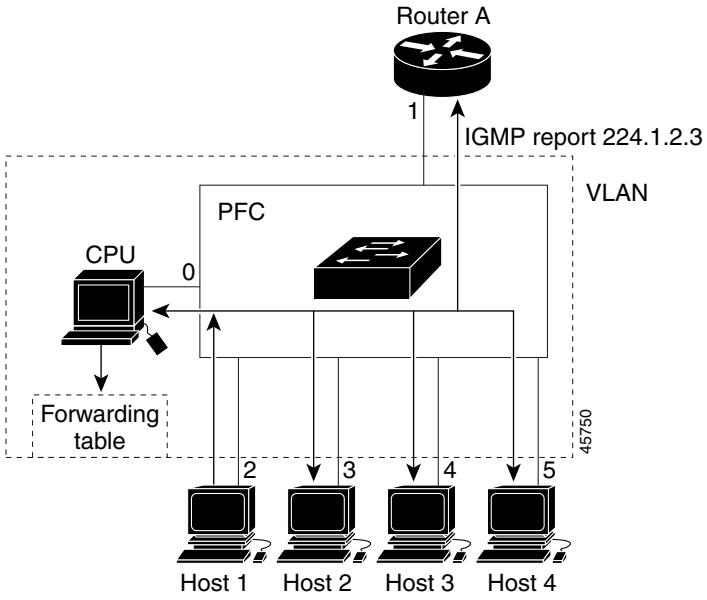
In response to an IGMP join request, the router creates an entry in its Layer 2 forwarding table for the VLAN on which the join request was received. When other hosts that are interested in this multicast traffic send IGMP join requests, the router adds them to the existing Layer 2 forwarding table entry. The router 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 router forwards multicast traffic for the multicast group specified in the join message to the interfaces where join messages were received (see Figure 30-1).

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 30-1 Initial IGMP Join Message



Multicast router A sends a general query to the router, 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 30-1, that includes the port numbers of Host 1, the multicast router, and the router internal CPU.

Table 30-1 IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1, 2

The router 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 30-2), the CPU receives that message and adds the port number of Host 4 to the forwarding table as shown in Table 30-2. 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 30-2 Second Host Joining a Multicast Group

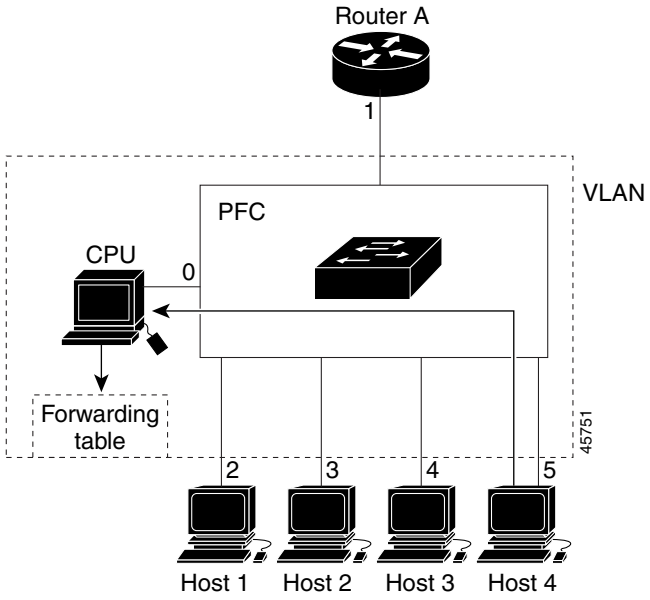


Table 30-2 Updated IGMP Snooping Forwarding Table

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 30-4
- Fast-Leave Processing, page 30-5

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 router 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 router as the IGMP querier so that it can send queries.

When enabled, the IGMP snooping querier sends out periodic IGMPv3 queries that trigger IGMP report messages from the router 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 Cisco 7600 series routers 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 router as the IGMP snooping querier.

You can configure a router to generate IGMP queries on a VLAN regardless of whether or not IP multicast routing is enabled.

Understanding IGMP Version 3 Support

These sections describe IGMP version 3 support:

IGMP Version 3 Support Overview

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 Cisco 7600 series router, 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 router 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 router 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 router does not remove the host from the LTL index until the host is no longer interested in receiving traffic from any source.

Proxy Reporting

IGMP supports proxy reporting for IGMPv1 and IGMPv2 messages to handle group-specific queries. These queries are not sent downstream, but the switch does respond to them directly. When the switch recieves a group-specific query, the switch terminates the query and sends an IGMP proxy report if there is a receiver for the group. There is no proxy reporting for IGMPv3 messages. For IGMPv3, a group-specific query or a group source-specific query is flooded to all VLAN member ports. The database for the IGMPv3 membership report is built based on the reports received.

Host reports responding to a specific query can be suppressed by the report suppression feature. Report suppression is supported for IGMPv1, IGMPv2 and IGMPv3 messages. With report suppression enabled (by default), when the switch recieves a general query, the switch starts a suppression cycle for reports from all hosts to each group or channel (S,G). Only the first report to the discovered multicast routers are forwarded; the rest of the reports are suppressed. For IGMPv1 and IGMPv2, the time of suppression is the report response time indicated in the general query message. For IGMPv3, suppression occurs for the entire general query interval.

Note

- Source-based filtering for IGMP version 3 reports is not supported in hardware. The states are maintained only in software and used for explicit host tracking and statistics collection. The source-only entries are deleted every 5 minutes and relearned to ensure that they are still valid.
- Turning off explicit host tracking disables fast-leave processing and proxy reporting.

Explicit Host Tracking

IGMPv3 supports explicit tracking of membership information on any port. The explicit-tracking database is used for fast-leave processing for IGMPv3 hosts, proxy reporting, and statistics collection. When explicit tracking is enabled on a VLAN, the IGMP snooping software processes the IGMPv3 report it receives from a host and builds an explicit-tracking database that contains the following information:

- The port connected to the host
- The channels reported by the host
- The filter mode for each group reported by the host
- The list of sources for each group reported by the hosts
- The router filter mode of each group
- For each group, the list of hosts requesting the source

Note

- Turning off explicit host tracking disables fast-leave processing and proxy reporting.
- When explicit tracking is enabled and the router 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 30-3 shows the default IGMP snooping configuration.

Table 30-3 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
IGMPv3 SSM Safe Reporting	Disabled

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 Multilayer Switch Feature Card (MSFC) as a CGMP server. Refer to the *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.2, "IP Multicast," "Configuring IP Multicast Routing," at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcpt3/1cfmulti.htm
- 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 14, "Configuring VLANs").
- Configure an IP address on the VLAN interface (see Chapter 21, "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 Cisco 7600 series routers in the VLAN. One router 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.
Step 3	`Router(config-if)# ip igmp snooping querier` `Router(config-if)# no ip igmp snooping querier`	Enables the 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 28, "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 30-9).

IGMP snooping allows Cisco 7600 series routers to examine IGMP packets and make forwarding decisions based on their content.

These sections describe how to configure IGMP snooping:

Note Except for the global enable 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 router 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 router, 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 router. 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/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_2/gtssmma.htm

Enabling SSM Safe Reporting

Note Source-specific multicast (SSM) safe reporting is presently deprecated.

When you configure SSM safe reporting, the group mode is IGMPv3 even in the presence of IGMPv1 and IGMPv2 hosts.

To make sure the router is able to support both IGMPv1, IGMPv2, and IGMPv3 hosts in the same 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 ssm-safe-reporting`	Enables support for both IGMPv2 and IGMPv3 hosts.
	`Router(config-if)# no ip igmp snooping ssm-safe-reporting`	Clears the configuration.

This example shows how to configure the router to support both IGMPv2 and IGMPv3 hosts:

```
Router(config)# interface vlan 10
Router(config-if)# ip igmp snooping ssm-safe-reporting
```

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

When you enable IGMP snooping, the router 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#
```

Troubleshooting

This section describes how to troubleshoot common IGMP issues.

Scenarios/Problems	Solution
How do I verify whether the multicast queries are sent and reports are received from the host?	Queries are generated by the IGMP PI code and forwarded through interfaces. Use the PI group specific **debug ip igmp grp**,**debug mmls igmp-even**t, and **debug mmls igmp-pak** commands.

How do I verify the IGMP membership of the node?	Use the **show ip igmp vrf 0 groups 0** command. This example shows a sample output from the command:
	`csc76d#show ip igmp vrf blue groups 226.6.6.6`
	`IGMP Connected Group Membership`
	`Group Address Interface Uptime Expires Last Reporter Group Accounted`
	`226.6.6.6 GigabitEthernet4/0/3 00:15:32 00:02:42 200.3.3.202`
	`csc76d#`

How do I know the properties of the IGMP interface?	Use the **show ip igmp vrf 0 interface** command. The command will tell you whether it is a Querier/DR and the timer values for each interface. This example shows a sample output from this command:
	`csc76d#show ip igmp vrf blue interface gi 4/0/3`
	`GigabitEthernet4/0/3 is up, line protocol is up`
	`Internet address is 200.3.3.3/24`
	`IGMP is enabled on interface`
	`Multicast Routing table blue`
	`Current IGMP host version is 2`
	`Current IGMP router version is 2`
	`IGMP query interval is 60 seconds`
	`IGMP configured query interval is 60 seconds`
	`IGMP querier timeout is 120 seconds`
	`IGMP configured 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 200.3.3.3 (this system)`
	`IGMP querying router is 200.3.3.2`
	`No multicast groups joined by this system`
	`csc76d#`

Scenarios/Problems	Solution

How do I verify that the IGMP packets are coming into the PI code?	Use the **debug ip igmp vrf 0 0** command to verify whether the IGMP packets are coming into the PI code. The following example shows a sample output of this command:
	`*Oct 6 11:31:40.263: IGMP(1): Received v2 Report on GigabitEthernet3/5 from 200.3.3.202 for 226.6.6.6`
	`*Oct 6 11:31:40.263: IGMP(1): Received Group record for group 226.6.6.6, mode 2 from 200.3.3.202 for 0 sources`
	`*Oct 6 11:31:40.263: IGMP(1): Updating EXCLUDE group timer for 226.6.6.6`
	`*Oct 6 11:31:40.263: IGMP(1): MRT Add/Update GigabitEthernet3/5 for (*,226.6.6.6) by 0`

How do I verify that the packets have reached the switch processor?	Use the **debug platform software multicast igmp event** and **debug platform software multicast igmp pak** commands to verify whether the packets have reached the switch processor. The following example shows the debug output:
	Note This command is introduced in 12.2(SRE) release. It may not work in old releases.
	`csc76b#show vlan internal usage | i 1025`
	`1025 GigabitEthernet3/5`
	`csc76b#`
	`*Oct 6 11:31:40.259: SP: RELAYED PAK to index 0x00084401, vlan 1025`
	`*Oct 6 11:31:40.259: SP: Packet dump:`
	`18000070: 0100 5E060606 00097B04 ..^.....{.Z>`
	`HdL`
	`18000080: E4700800 45C0001C 018E0000 010203B9 dp..E@.........9Z>`
	`HdL`
	`18000090: C80303CA E2060606 160001F3 E2060606 H..Jb......sb...Z>`
	`HdL`
	`180000A0: 00000000 00000000 00000000 00000000 ................Z>`
	`HdL`
	`180000B0: 000018 ... Z>`
	`HdL`

How do I check the multicast groups with receivers that are directly connected to the router and that were learned through IGMP?	Use the **show ip igmp groups** command. This is a sample output from the command with the **group-address** argument and **detail** keyword:
	`Router# show ip igmp groups 192.168.1.1 detail`
	`Interface:  Ethernet3/2`
	`Group:  192.168.1.1`
	`Uptime:  01:58:28`
	`Group mode:  INCLUDE`
	`Last reporter:  10.0.119.133`
	`CSR Grp Exp:  00:02:38`
	`Group source list: (C - Cisco Src Report, U - URD, R - Remote S-Static, M - SSM Mapping)`
	`Source Address  Uptime  v3 Exp  CSR Exp  Fwd  Flags`
	`172.16.214.1  01:58:28  stopped  00:02:31  Yes  C`

Scenarios/Problems	Solution

How do I verify whether the incoming interface and output interfaces are proper?	To verify that the incoming interface and output interfaces are proper, follow these steps:
	• Use the **show ip mroute** command to display the MRIB information. Check the incoming interface and outgoing interface list. If this information is correct then the Mroute information is correct. If it is incorrect, then enable the **debug ip mrouting** command. This output is from the **debug ip mrouting** command:
	`13.0.0.1, 228.1.1.1), 04:02:28/00:03:19, flags: FT`
	`Incoming interface: GigabitEthernet4/0/0, RPF nbr 0.0.0.0`
	`Outgoing interface list:`
	`FastEthernet1/11, Forward/Sparse, 03:33:01/00:02:59`
	`TenGigabitEthernet2/0/0, Forward/Sparse, 03:38:23/00:03:29`
	• Use the **show ip mrib route** command to display the MRIB information with MRIB flags. Look at the flags. See whether the **A** Flag is against the accept interface and the **F Flag** is against the forwarding interface. This should match with the above output of **show ip mroute** command. If it is incorrect, then enable **debug ip mrib command**. This output is from the **debug ip mrib** command:
	`(13.0.0.1,228.1.1.1) RPF nbr: 0.0.0.0 Flags: K DDE`
	`GigabitEthernet4/0/0 Flags: A`
	`FastEthernet1/11 Flags: F NS`
	`TenGigabitEthernet2/0/0 Flags: F NS`
	• Check the output of the **show ip mfib** command. Check if the information is correct by looking at the **A Flag** against the accept interface and the **F Flag** against the forwarding interface. This should match with the output of **show ip mrib route** command. If it is not correct then enable **debug ip mfib** command. This is a sample output:

Scenarios/Problems	Solution
	(13.0.0.1,228.1.1.1) Flags: K HW DDE Platform Flags: HW Slot 5: HW Forwarding: 0/0, Platform Flags: HF Slot 4: HW Forwarding: 70515/104503230, Platform Flags: HF Slot 2: HW Forwarding: 0/0, Platform Flags: HF Slot 1: HW Forwarding: 0/0, Platform Flags: HF SW Forwarding: 1/0/1482/0, Other: 84/0/84 HW Forwarding: 70515/5/1482/57, Other: 0/0/0 GigabitEthernet4/0/0 Flags: RA A Platform Flags: FastEthernet1/11 Flags: RF F NS Platform Flags: HW CEF: Adjacency with MAC: 01005E01010100152BE0AEC00800 Pkts: 0/0 TenGigabitEthernet2/0/0 Flags: RF F NS Platform Flags: HW CEF: Adjacency with MAC: 01005E01010100152BE0AEC00800 Pkts:0/0 • Check the output of the **show ip rpf** command. Compare the output with the source address of the stream and ensure that it is same as the one pointed by the incoming interface in MROUTE, MRIB and MFIB outputs above. 7606-3#sh ip rpf 13.0.0.1 RPF information for ? (13.0.0.1) RPF interface: GigabitEthernet4/0/0 <<<<<<<<<<<<<<<<<<<< RPF neighbor: ? (13.0.0.1) - directly connected RPF route/mask: 13.0.0.0/8 RPF type: multicast (connected) Doing distance-preferred lookups across tables RPF topology: ipv4 multicast base 7606-3#
How do I display information about PIM neighbors discovered by PIMv1 router query messages or PIMv2 hello messages?	Use the **show ip pim neighbor** command in user EXEC or privileged EXEC mode. This example shows output from the command: Router# **show ip pim neighbor** PIM Neighbor Table Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority, S - State Refresh Capable Neighbor Interface Uptime/Expires Ver DR Address Prio/Mode 10.0.0.1 GigabitEthernet10/2 00:01:29/00:01:15 v2 1 / S 10.0.0.3 GigabitEthernet10

Scenarios/Problems	Solution

How do I know the active rendezvous points (RPs) that are cached with associated multicast routing entries?	Use the **show ip pim rp** command in user EXEC or privileged EXEC mode. This is a sample output from the command:
	`Router# show ip pim rp`
	`Group:227.7.7.7, RP:10.10.0.2, v2, v1, next RP-reachable in 00:00:48`
	
	This is a sample output from the **show ip pim rp command** when the **mapping** keyword is specified:
	`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) 227.0.0.0/8`
	`  RP 10.10.0.2 (?), v2v1, bidir`
	`    Info source:10.10.0.2 (?), via Auto-RP`
	`         Uptime:00:01:42, expires:00:00:32`
	`Group(s) 228.0.0.0/8`
	`  RP 10.10.0.3 (?), v2v1, bidir`
	`    Info source:10.10.0.3 (?), via Auto-RP`
	`         Uptime:00:01:26, expires:00:00:34`
	`Group(s) 229.0.0.0/8`
	`  RP 10.10.0.5 (mcast1.cisco.com), v2v1, bidir`
	`    Info source:10.10.0.5 (mcast1.cisco.com), via Auto-RP`
	`         Uptime:00:00:52, expires:00:00:37`
	`Group(s) (-)230.0.0.0/8`
	`  RP 10.10.0.5 (mcast1.cisco.com), v2v1, bidir`
	`    Info source:10.10.0.5 (mcast1.cisco.com), via Auto-RP`
	`         Uptime:00:00:52, expires:00:00:37`
	
	This is a sample output from the **show ip pim rp command** when the **metric** keyword is specified:
	`Router# show ip pim rp metric`
	`RP Address  Metric Pref  Metric  Flags  RPF Type Interface`
	`10.10.0.2  0  0  L  unicast Loopback0`
	`10.10.0.3  90  409600  L  unicast Ethernet3/3`
	`10.10.0.5  90  435200  L  unicast Ethernet3/3`

Scenarios/Problems	Solution

How do I know information about interfaces configured for PIM?	Use the **show ip pim interface** command in user EXEC or privileged EXEC mode. This is a sample output from the command:
	
	`Router# show ip pim interface`
	`Address Interface Ver/ Nbr Query DR`
	`DR`
	`Mode Count Intvl`
	`Prior`
	`10.1.0.1 GigabitEthernet0/0 v2/SD 0 30 1`
	`10.1.0.1`
	`10.6.0.1 GigabitEthernet0/1 v2/SD 1 30 1`
	`10.6.0.2`
	`10.2.0.1 ATM1/0.1 v2/SD 1 30 1`
	`0.0.0.0`
	
	This is a sample output from the **show ip pim interface** command when an interface is specified:
	
	`Router# show ip pim interface Ethernet1/0`
	`Address Interface Ver/ Nbr Query DR`
	`DR`
	`Mode Count Intvl`
	`Prior`
	`172.16.1.4 Ethernet1/0 v2/S 1 100 ms 1`
	`172.16.1.4`
	
	This is a sample output from the **show ip pim interface** command when the **count** keyword is specified:
	
	`Router# show ip pim interface count`
	`Address Interface FS Mpackets In/Out`
	`172.16.121.35 Ethernet0 * 548305239/13744856`
	`172.16.121.35 Serial0.33 * 8256/67052912`
	`192.168.12.73 Serial0.1719 * 219444/862191`
	
	This is a sample output from the **show ip pim interface** command when the **count** keyword is specified and IP MMLS is enabled. The example lists the PIM interfaces that are fast switched and process switched, and the packet counts for these interfaces. The H flag is added to interfaces where IP MMLS is enabled.

Scenarios/Problems	Solution

```
Router# show ip pim interface count
States: FS - Fast Switched, H - Hardware Switched
Address          Interface          FS  Mpackets In/Out
192.168.10.2       Vlan10             * H 40886/0
192.168.11.2       Vlan11             * H 0/40554
192.168.12.2       Vlan12             * H 0/40554
192.168.23.2       Vlan23             *   0/0
192.168.24.2       Vlan24             *   0/0
```

These are two sample outputs from the **show ip pim interface** command when the **df** keyword is specified:

```
Router# show ip pim interface df
Interface          RP               DF Winner        Metric
Uptime
Ethernet3/3        10.10.0.2        10.4.0.2         0
00:03:49
                   10.10.0.3        10.4.0.3         0
00:01:49
                   10.10.0.5        10.4.0.4         409600
00:01:49
Ethernet3/4        10.10.0.2        10.5.0.2         0
00:03:49
                   10.10.0.3        10.5.0.2         409600
00:02:32
                   10.10.0.5        10.5.0.2         435200
00:02:16
Loopback0          10.10.0.2        10.10.0.2        0
00:03:49
                   10.10.0.3        10.10.0.2        409600
00:02:32
                   10.10.0.5        10.10.0.2        435200
00:02:16
Router# show ip pim interface Ethernet3/3 df 10.10.0.3
Designated Forwarder election for Ethernet3/3, 10.4.0.2, RP
10.10.0.3
  State                           Non-DF
  Offer count is                  0
  Current DF ip address           10.4.0.3
  DF winner up time               00:02:33
  Last winner metric preference   0
  Last winner metric
```

Scenarios/Problems	Solution

How do I know the replication mode for the system?	Use the **show platform software multicast ip capability** command. The command displays the replication mode. There are two replication mode: Ingress and Egress.
	
	In Ingress mode, the ingress DFC line card or the SP (in case the packets arrive on a non DFC line card) replicates for each of the outgoing interfaces. The ingress EARL does a lookup and sends the result to the replication engine to perform the replication of the packets.
	
	In Egress mode, the ingress line card just transmits one copy of the packet to each of the egress line card which is a DFC. The DFC looks for the packet and replicates the interfaces local to its line card. This mode of replication is better from the fabric backplane utilization perspective. The special unique EGRESS VLAN is used for sending the packet to the egress line card. This is the output of s**how vlan internal usage | i Egress multicast**.
	
	`7606-3#`
	`7606-3(config)#`**`ip multicast hardware-switching`** `replication-mode egress`
	`Warning: This command will change the replication mode for all address families.`
	`Warning: Egress replication-mode forced by CLI in presence of an egress-incapable card`
	`7606-3(config)#`
	`02:09:02: %CONST_MFIB_RP-6-REPLICATION_MODE_CHANGE: Replication Mode Change Detected. Current system replication mode is Egress`
	`7606-3(config)#end`
	`7606-3#`
	`7606-3`**`#show platform software multicast ip capability`**
	
	`Current System HW Replication Mode : Egress`
	`Auto-detection of Replication Mode : OFF`
	
	`Slot Replication-Capability Replication-Mode`
	`1 Ingress Egress`
	`2 Egress Egress`
	`4 Egress Egress`
	`5 Egress Egress`
	`7606-3#`

Scenarios/Problems	Solution
	7606-3#**show vlan internal usage | i Egress multicast** 1015 IPv4 VPN 0 Egress multicast 7606-3# 7606-3#**conf t** Enter configuration commands, one per line. End with CNTL/Z. 7606-3(config)#**ip multicast hardware-switching** replication-mode ingress Warning: This command will change the replication mode for all address families. 7606-3(config)#end 7606-3# 02:11:54: %CONST_MFIB_RP-6-REPLICATION_MODE_CHANGE: Replication Mode Change Detected. Current system replication mode is Ingress 7606-3# 7606-3#**show platform software multicast ip capability** Current System HW Replication Mode : Ingress Auto-detection of Replication Mode : OFF Slot Replication-Capability Replication-Mode 1 Ingress Ingress 2 Egress Ingress 4 Egress Ingress 5 Egress Ingress 7606-3#
How do I display information about the internal VLAN allocation?	Use the **show vlan internal usage** command in privileged EXEC mode. These are internal vlans which are used by the 7600 platform. The scope of these vlans are limited to the box and has no meaning outside the scope of the box. Each vlan corresponds to one interface on the Cisco 7600 router. This sample output is of the IIF and OIF being represented as a VLAN. 7606-3#s**how vlan internal usage | i 1028** 1028 FastEthernet1/13 **<<<<< Internal vlan 1028 is** mapped to interface Fa1/13 7606-3#

Scenarios/Problems	Solution

How do I identify the features enabled on the VLAN?	Use the **show mls vlan-ram** command. The command displays the features enabled on VLAN. It also helps you to validate the VLAN to MLS VPN mapping. MLS VPN is different from IOS VPN, and used for HW switching. This is a sample output from the command:
	`sp#show mls vlan-ram 1029 1031`
	`vlan eom nf-vpn mpls mc-base siteid stats rpf vpn-num bgp-grp l2-metro rpf-pbr-ovr`
	`----+---+------+----+-------+------+-----+---+-------+-------+---- ----+-----------`
	`1029 - - * 0 0 - - 0 0 - *`
	`1030 - - * 0 0 - - 260 0 - *`
	`1031 - - * 0 0 - - 261 0 - *`
	
	In the above output, the VLAN 1030 has MPLS VPN number 260 assigned to it. This may be different from IOS VPN number.

How do I know the MPLS label and COS bits used for the VPN number?	Use the **show mls vpn-cam** command. The command displays MPLS label and COS bits used for the VPN number. This is a sample output from the command:
	`ESM-20G-2#show mls vpn-cam start 0 end 0 all`
	`TYCHO Sindex VPN RAM: Dumping entries 0 -> 0`
	`Key: * => Set, - => Clear`
	
	`Index MPLS Label VPN COS`
	`=====+============+======+=====`
	`0 0 0 0`
	`ESM-20G-2#`

CHAPTER 31

Configuring PIM Snooping

This chapter describes how to configure protocol independent multicast (PIM) snooping on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding How PIM Snooping Works

In networks where a Layer 2 router interconnects several routers, such as an Internet exchange point (IXP), the router floods IP multicast packets on all multicast router ports by default, even if there are no multicast receivers downstream. With PIM snooping enabled, the router 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 router 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 Cisco 7600 series router. 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 31-1 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 31-1 PIM Join Message Flow without PIM Snooping

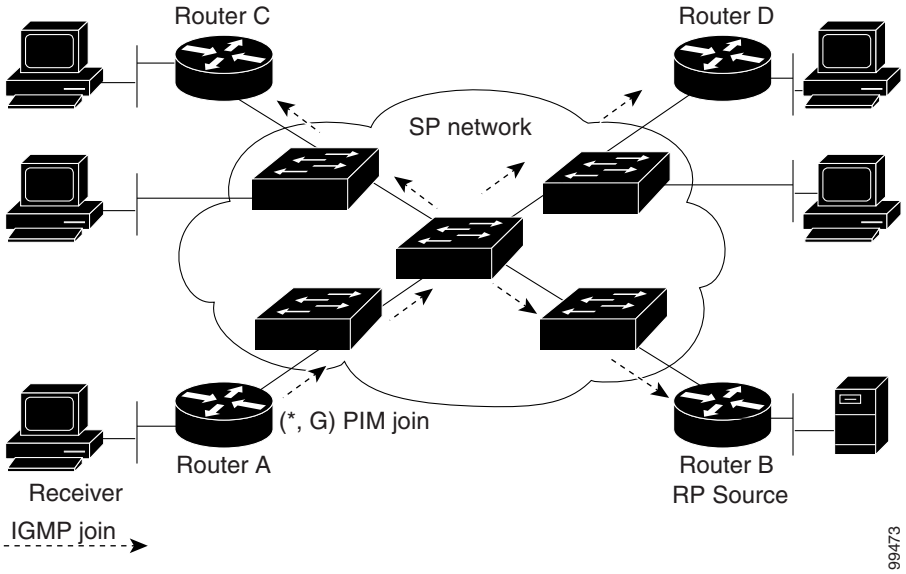


Figure 31-2 shows the flow of a PIM join message with PIM snooping enabled. In the figure, the switches restrict the PIM join message and forward it only to the router that needs to receive it (Router B).

Figure 31-2 PIM Join Message Flow with PIM Snooping

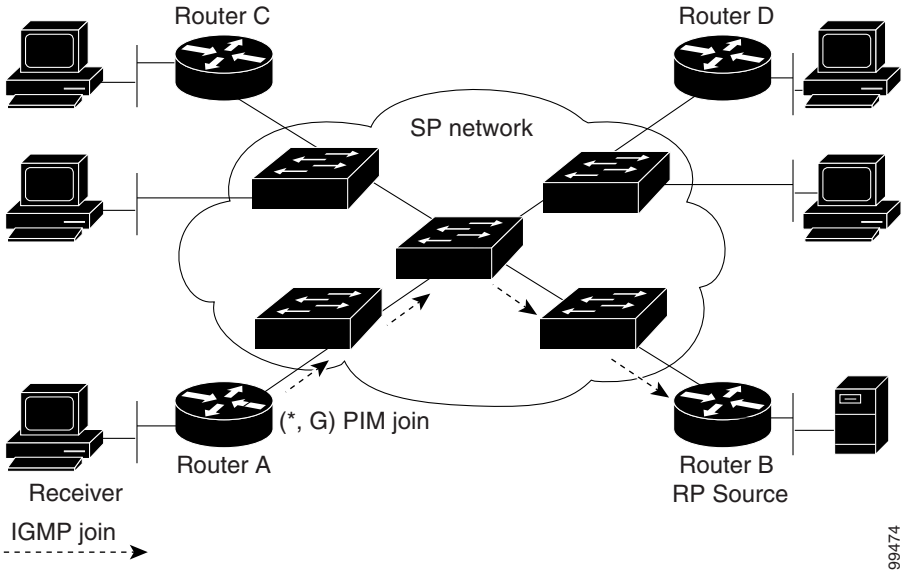


Figure 31-3 shows the flow of data traffic without PIM snooping enabled. In the figure, the switches flood the data traffic intended for Router A to all connected routers.

Figure 31-3 Data Traffic Flow without PIM Snooping

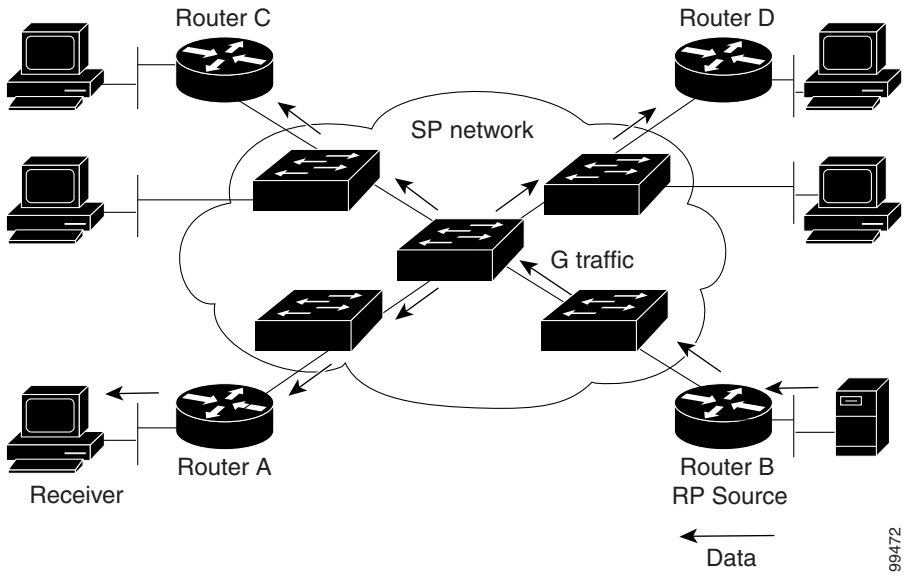
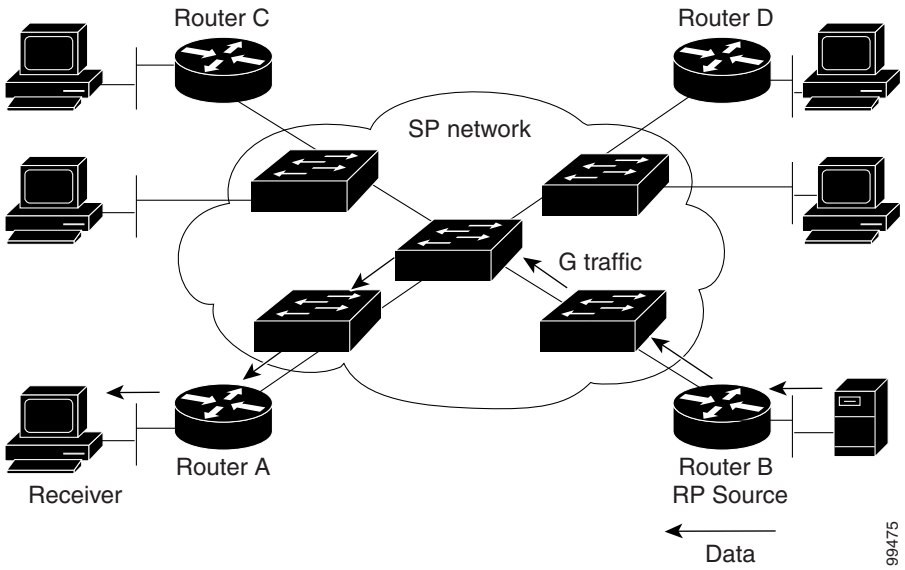


Figure 31-4 shows the flow of data traffic with PIM snooping enabled. In the figure, the switches forward the data traffic only to the router that needs to receive it (Router A).

Figure 31-4 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 router 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 31-5
- Enabling PIM Snooping in a VLAN, page 31-5
- Disabling PIM Snooping Designated-Router Flooding, page 31-6

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

- The PIM snooping DR flooding enhancement is supported with the Supervisor Engine 720
- Do not disable designated-router flooding on routers in a Layer 2 broadcast domain that supports multicast sources.

By default, routers 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 32

Configuring Network Security

This chapter contains network security information unique to the Cisco 7600 series routers, 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
- The Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

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

- The *Cisco 7600 Series Router Cisco IOS Command Reference* at this URL:

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/prod_command_reference_list.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 PFC3 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 PFC3 provides hardware support for RPF check of traffic from multiple interfaces.

With strict-method Unicast RPF check, the PFC3 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 PFC3 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 PFC3 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 PFC determines whether or not traffic matches the ACL. The PFC sends the traffic denied by the RPF ACL to the MSFC 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 MSFC for the Unicast RPF check, they can overload the MSFC.
- The PFC provides hardware support for traffic that does not match the Unicast RPF check ACL, but that does match an input security ACL.
- The PFC does not provide hardware support 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 32-3
- Configuring the Multiple-Path Unicast RPF Check Mode on a PFC3, page 32-5
- Enabling Self-Pinging, page 32-6

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]` `Router(config-if)# no ip verify unicast`	Configures the Unicast RPF check mode. 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 router.

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 PFC3

To configure the multiple-path Unicast RPF check mode on a PFC3, 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 PFC3.
	`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 multiple path RPF check, note the following information:

- **punt** (default)—The PFC3 performs the Unicast RPF check in hardware for up to two interfaces per prefix. Packets arriving on any additional interfaces are redirected (punted) to the MSFC3 for Unicast RPF check in software.
- **pass**—The PFC3 performs the Unicast RPF check in hardware for single-path and two-path prefixes. Unicast RPF check is disabled for packets coming from multipath prefixes with three or more reverse-path interfaces (these packets always pass the Unicast RPF check).
- **interface-group**—The PFC3 performs the Unicast RPF check in hardware for single-path and two-path prefixes. The PFC3 also performs the Unicast RPF check for up to four additional interfaces per prefix through user-configured multipath Unicast RPF check interface groups. Unicast RPF check is disabled for packets coming from other multipath prefixes that have three or more reverse-path interfaces (these packets always pass the Unicast RPF check).

This example shows how to configure multiple path RPF check:

```
Router(config)# mls ip cef rpf mpath punt
```

Configuring Multiple-Path Interface Groups on a PFC3

To configure multiple-path Unicast RPF interface groups on a PFC3, 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 PFC3.
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 router 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 router 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 33

Understanding Cisco IOS ACL Support

This chapter describes Cisco IOS ACL support on the Cisco 7600 series routers:

For complete information about configuring Cisco IOS ACLs, refer to the *Cisco IOS Security Configuration Guide*, Release 12.2 at this URL:

http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfacls.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 34, "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 PFC does not provide hardware support for Cisco IOS IPX ACLs. Cisco IOS IPX ACLs are supported in software on the MSFC.
- By default, the MSFC 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 MSFC to be dropped (10 packets per second, maximum), which generates ICMP-unreachable messages.

  To eliminate the load imposed on the MSFC 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.

Hardware and Software ACL Support

Access control lists (ACLs) can be processed in hardware by the Policy Feature Card (PFC), a Distributed Forwarding Card (DFC), or in software by the Multilayer Switch Feature Card (MSFC). The following behavior describes software and hardware handling of ACLs:

- The PFC 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. Cisco 7600 series routers do not support the **access-enable host timeout** command.

- IP accounting for an ACL access violation on a given port is supported by forwarding all denied packets for that port to the MSFC for software processing without impacting other flows.
- The PFC does not provide hardware support for Cisco IOS IPX ACLs. Cisco IOS IPX ACLs are supported in software on the MSFC.
- 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 PFC3" section on page 33-3).
- 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.

Optimized ACL Logging with a PFC3

These sections describe OAL:

Understanding OAL

Optimized ACL Logging (OAL) provides hardware support for ACL logging. Unless you configure OAL, packets that require logging are processed completely in software on the MSFC. OAL permits or drops packets in hardware on the PFC3 and uses an optimized routine to send information to the MSFC3 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 router. With OAL configured, use SPAN to capture traffic.
- OAL is supported only on the PFC3.
- OAL supports only IPv4 unicast packets.
- OAL supports VACL logging of permitted ingress traffic
- OAL does not 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 MSFC3 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.

Configuring OAL

These sections describe how to configure OAL:

Note

- For complete syntax and usage information for the commands used in this section, refer to the *Cisco 7600 Series Router Cisco IOS Command Reference*.
- 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: 8000.
- **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 33-5
- Determining Logical Operation Unit Usage, page 33-6

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 33-6 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 34

Configuring VLAN ACLs

This chapter describes how to configure VLAN ACLs (VACLs) on Cisco 7600 series routers.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- OAL and VACL capture are incompatible. Do not configure both features on the router. With OAL configured (see the "Optimized ACL Logging with a PFC3" section on page 33-3), 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 34-1 shows a VACL applied on bridged packets.

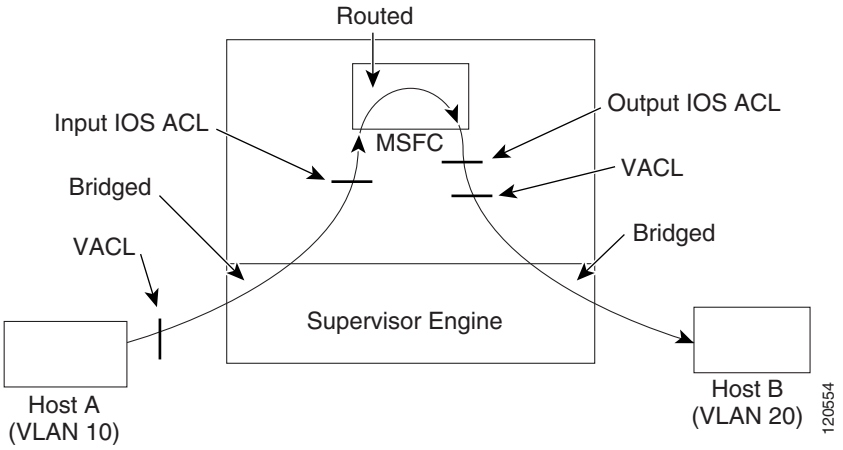
Figure 34-1 Applying VACLs on Bridged Packets

Routed Packets

Figure 34-2 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 34-2 Applying VACLs on Routed Packets

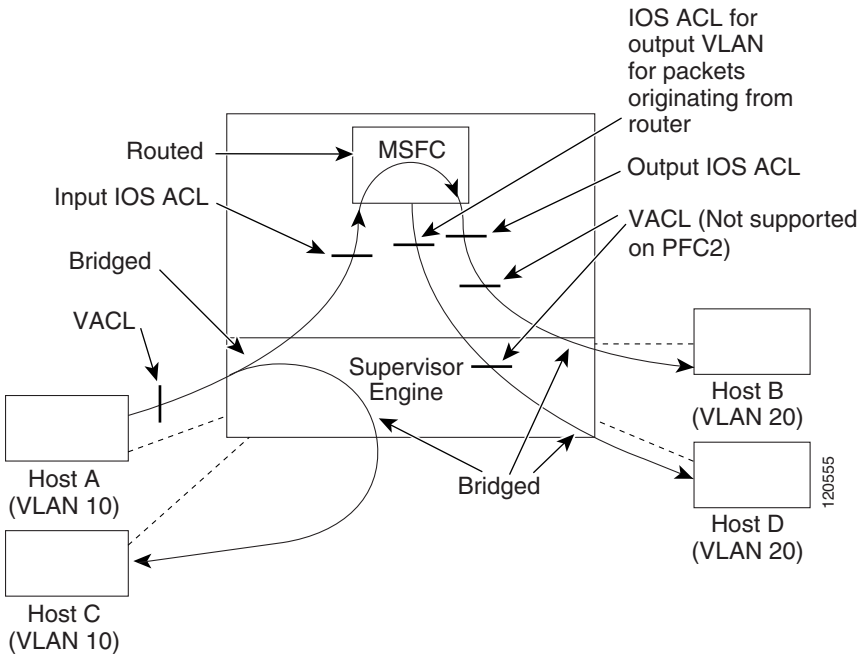


Multicast Packets

Figure 34-3 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 34-3 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 44-55) 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.

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 34-8.

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 44-55.
- 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 34-8.

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 34-9.
- 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.
- With a PFC3, 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 34-8.

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.
`Router(config)# no vlan filter map_name [vlan-list vlan_list | interface type[1] number[2]]`	Removes the VLAN access map from 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 34-8.

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 10-8 and the "Configuring the Layer 2 Trunk Not to Use DTP" section on page 10-9).

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 10-7) 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 10-14) before you enter the **switchport capture** command.
- The capture port supports only egress traffic. No traffic can enter the router 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 34-4 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 35

Private Hosts (Using PACLs)

This chapter describes the Private Hosts feature, which is being introduced for the Cisco 7600 series router in Cisco IOS Release 12.2SRB. This chapter contains the following sections:

Overview

The Private Hosts feature provides Layer 2 (L2) isolation between the hosts in a VLAN. You can use Private Hosts as an alternative to the Private VLAN isolated-trunks feature, which is currently not available on the Cisco 7600 router.

Service Providers (SPs) worldwide face increasing demand to provide their customers with triple-play services (voice, video, and data) over a single physical interface (copper or fiber). Typically, triple-play services are delivered over three different VLANs for each user, even though the VLAN for video traffic is often shared by multiple end users.

The key benefits of the Private Hosts feature are the ability to:

- Isolate traffic among hosts (subscribers) that share the same VLAN ID
- Reuse VLAN IDs across different subscribers, which improves VLAN scalability by making better use of the 4096 VLANs allowed
- Prevent MAC spoofing to prevent denial of service (DOS) attacks

The Private Hosts feature uses port-based access control lists (PACLs) to provide Layer 2 isolation between hosts on trusted ports within a purely Layer 2 domain. The PACLs isolate the hosts by imposing Layer 2 forwarding constraints on the router ports.

Note In Release 12.2SRB, PACLs are supported only as part of Private Hosts; you cannot configure your own PACLs. Instead, the router creates and applies PACLs based on your Private Hosts configuration.

The sections that follow provide more detail about the following Private Hosts concepts:

- Port ACLs, page 35-5

Isolating Hosts in a VLAN

Typically, triple-play services (voice, video, and data) are delivered over three different VLANs for each user, even though the VLANs for the same set of services could be shared among multiple end users. For example, if 10 end users all receive the same set of services, Private Hosts can be used to deliver the services to all of 10 end users over a single set of VLANs. However, to allow VLAN sharing, the service provider must be able to isolate traffic between the users (hosts) at Layer 2.

The Private Hosts feature provides Layer 2 isolation among hosts (end users) in a VLAN. By isolating the hosts, a service provider can use a single set of VLANs to deliver the same set of broadband or metro Ethernet services to multiple end users while ensuring that none of the hosts in the VLAN can communicate directly with each other. For example, VLAN 10 can be used for voice traffic, VLAN 20 for video traffic, and VLAN 30 for data traffic.

When the Cisco 7600 router is used as a DSLAM Gigabit Ethernet (GE) aggregator, the DSLAM is connected to the router through a trunk port that can carry data for multiple VLANs. The service provider uses a single physical port and a single set of VLANs to deliver the same set of services to different end users (isolated hosts). A separate VLAN is used for each service (voice, video, and data).

Figure 35-1 shows an example of triple-play services being delivered from the Cisco 7600 router to multiple end users attached to a DSLAM. In the figure note that:

- A single trunk link (between the router and the DSLAM) carries traffic for all three VLANs.
- Virtual circuits deliver the VLAN traffic from the DSLAM to individual end users.

Figure 35-1 VC to VLAN Mapping

A	Trunk link carries: • One voice VLAN • One video VLAN • One data VLAN	**B**	DSLAM maps voice, video, and data traffic between VLANs and VCs.
		C	Individual VCs carry voice, video, and data traffic between DSLAM and each host.

Restricting Traffic Flow (Using Private Hosts Port Mode and PACLs)

The Private Hosts feature uses PACLs to restrict the type of traffic that is allowed to flow through each of the ports configured for Private Hosts. A port's mode (specified when you enable Private Hosts on the port) determines what type of PACL is applied to the port. Each type of PACL restricts the traffic flow for a different type of traffic (for example, from content servers to isolated hosts, from isolated hosts to servers, and traffic between isolated hosts).

The following list describes the port modes used by the Private Hosts feature (see Figure 35-2):

- Isolated—Ports connected to the DSLAMs that the end users (isolated hosts) are connected to. The hosts on the VLANs on these ports need to be isolated from each other. Hosts connected through these ports are allowed to pass unicast traffic to upstream devices only.
- Promiscuous—Ports that face the core network or the Broadband Remote Access Server (BRAS) devices and multicast servers that are providing the broadband services.
- Mixed—Ports that interconnect Cisco 7600 routers. These ports can act as either an isolated port or a promiscuous port, depending on Spanning Tree Protocol (STP) topology changes. These ports allow unicast traffic to upstream devices (such as BRAS and multicast servers) only.

The following list summarizes how the Private Hosts feature restricts traffic flow:

- Broadcast traffic at the ingress of the service provider network is redirected to BRAS and multicast servers (such as video servers).
- All unicast traffic between access routers (Cisco 7600 routers connected to each other) is blocked except for traffic directed toward BRAS and multicast servers.
- The unknown unicast flood blocking (UUFB) feature is used to block unknown unicast traffic on DSLAM-facing ports.

Figure 35-2 shows the different types of port modes (isolated, promiscuous, and mixed) used in a Private Hosts configuration.

Figure 35-2 Private Hosts Port Types (Modes)

A	Promiscuous ports	Permit all traffic from BRAS to hosts.
B	Mixed-mode ports	Permit broadcast traffic from BRAS. Redirect broadcast traffic from hosts to promiscuous and mixed-mode ports. Permit traffic from BRAS to hosts and from hosts to BRAS. Deny all other host to host traffic.
C	Isolated ports	Permit unicast traffic from host to BRAS only; block unicast traffic between ports. Redirect all broadcast traffic from host to BRAS. Deny traffic from BRAS (to prevent spoofing). Permit multicast traffic (IPv4 and IPv6).

Note The term BRAS represents upstream devices such as BRAS, multicast servers (such as video servers), or core network devices that provide access to these devices.

Port ACLs

The Private Hosts software creates several types of port ACLs (PACLs) to impose Layer 2 forwarding constraints on router ports. Each type of PACL restricts traffic flow for a particular type of traffic (for example, from content servers to isolated hosts, from isolated hosts to servers, and traffic between isolated hosts).

The software creates PACLs for the different types of Private Hosts ports based on the MAC addresses of the content servers providing broadband services and the VLAN IDs of the isolated hosts to deliver those services to. You specify the mode in which each Private Hosts port is to operate and the software applies the appropriate PACL to the port based on the port's mode (isolated, promiscuous, or mixed).

Following are examples of the different types of PACLs that are used by the Private Hosts feature.

Isolated Hosts PACL

Following is an example of a PACL for isolated ports:

```
deny host BRAS_MAC any
permit any host BRAS_MAC
redirect any host FFFF.FFFF.FFFF to LTLIndex 6
permit any 0100.5E00.0000/0000.007F.FFFF
permit any 3333.0000.0000/000.FFFF.FFFF
deny any any
```

Promiscuous Port PACL

Following is an example of a PACL for promiscuous ports:

```
permit host BRAS_MAC any
deny any any
```

Mixed-Mode Port PACL

Following is an example of a PACL for mixed-mode ports:

```
permit host BRAS_MAC ffff.ffff.ffff
redirect any host FFFF.FFFF.FFFF to LTLIndex 6
permit host BRAS_MAC any
permit any host BRAS_MAC
deny any any
```

Configuration Guidelines and Limitations

Observe the following guidelines and limitations as you configure the Private Hosts feature on Cisco 7600 routers:

- Software and hardware requirements:
  - Cisco IOS Release 12.2SRB or later
  - RSP720 (with PFC3C or PFC3CXL), Sup720 (with PFC3B or PFC3BXL), or Sup32
  - Supported on line cards with Fast Ethernet or Gigabit Ethernet (GE) interfaces that can be configured as switch ports (for example, SIP-600, ESM-20, and 67xx LAN cards). (Note that the SIP-400 and Enhanced FlexWAN do not support Private Hosts.)
- Private Hosts and Private VLANs cannot both be configured on the same port (interface). Both features can co-exist on the router, but each feature must be configured on different ports.
- Private Hosts is an end-to-end feature. You must enable the feature on all of the routers between the DSLAMs and upstream devices like BRAS and multicast servers.

- Currently, only trusted ports can be configured as isolated ports.
- Supported on Layer 2 interfaces that are configured as switchports (802.1q or ISL trunk ports).
- Supported on port-channel interfaces (Etherchannel, FastEtherchannel, and GigabitEtherchannel). You must enable Private Hosts on the port-channel interface; you cannot enable the feature on member ports.
- DAI and DHCP snooping cannot be enabled on a Private Hosts port unless all of the VLANs on the port are configured for snooping.

The following protocol-independent MAC ACL restrictions also apply:

- You can configure the following interface types for Protocol-independent MAC ACL filtering:
  - VLAN interfaces with no IP address
  - Physical LAN ports that support EoMPLS
  - Logical LAN subinterfaces that support EoMPLS
- Protocol-independent MAC ACL filtering applies MAC ACLs to all ingress traffic types (for example, IPv4 traffic, IPv6 traffic, and MPLS traffic, in addition to MAC-layer traffic).
- Ingress traffic that is permitted or denied by a protocol-independent MAC ACL is processed by egress interfaces as MAC-layer traffic. You cannot apply egress IP ACLs to traffic permitted or denied by a MAC ACL on an interface configured for protocol-independent MAC ACL filtering.
- Do not configure protocol-independent MAC ACL filtering on VLAN interfaces where you have configured an IP address.
- Do not configure protocol-independent MAC ACL filtering with microflow policing when the permitted traffic would be bridged or Layer 3 switched in hardware by the PFC3.
- Protocol-independent MAC ACL filtering supports microflow policing when the permitted traffic is routed in software by the MSFC.

The following limitations are applicable when you apply a private host to a port with only a subset of configured VLANS:

Table 35-1 PACL Scenarios and Limitations

Scenarios	Limitations
For all the VLANs in the Private-Host enabled port associated with Private-Host VLAN-list	• Since IP and non-IP traffic are subjected to PACL, based on the configured MAC addresses in the mac-list , both IP and non-IP traffic are permitted or denied. • From 12.2(SRD4) onwards, you can associate Private Host VLAN-list with one VLAN having cross connect(xconnect). Traffic received from the isolated Port is subjected to PACL.If another VLAN in the VLAN-list is configured with cross-connect, it is rejected. Similarly if a VLAN configured with cross-connect is added to Private Host VLAN-list, it is rejected if the VLAN-list already has a VLAN with cross-connect. • In the above case traffic which are permitted by PACL will be switched over the cross-connect.
For all the VLANs in the Private-Host enabled port not associated with Private-Host VLAN-list	• All Non-IP traffic are subjected to the PACL. The traffic is permitted or denied based on mac addresses configured in the Private Host mac-list. • IP traffic on these VLANs are not subjected to PACL. • If cross connect is configured on these VLANs, the IP traffic is switched to the cross connect without being subjected to PACL. • If the VLAN is Layer-3 routed (SVI), and the IP traffic is Layer-3 routed, the default ACL on the port denies all the packet. However, the L3 packets are forwarded to the CPU (MSFC) where it is rate-limited. • If the VLAN is associated with cross-connect, non-IP traffic is switched on the cross- connect only if the PACL permits the traffic. • If the VLAN is configured with **mac packet-classify**, then both IP and non-IP traffic on the VLAN are subjected to PACL.

- The following protocol-independent MAC ACL restrictions also apply from release 12.2(33) SRD4 onwards:
  - The Private Host feature prevents any traffic on the VLAN, which has Private Host configured, from passing directly between any two subscribers that share the VLAN.
  - You can configure the system where one VLAN on the system operates with Private Host and cross-connect connectivity.

- If one VLAN is configured for cross-connect and Private Host, then the configured VLAN's cross-connect is in promiscuous mode. However, you cannot apply a ACL on the configured cross-connect when traffic is relayed from the core side.
- There is no change in scale or performance when you apply the MAC ACL restrictions.
- Private Host limits VPLS support for only one VLAN. If the Private Host VLAN-list already has a VPLS VLAN (VLAN with cross-connect), the addition of another VPLS VLAN is blocked. Similarly, if any VLAN in the VLAN-list has cross-connect configured, the configuration of cross-connect on another VLAN in the VLAN-list is blocked.

Note In Release 12.2SRB, PACLs are supported only as part of Private Hosts; you cannot configure your own PACLs. Instead, the router creates and applies PACLs based on your Private Hosts configuration.

ACL Guidelines

The following configuration guidelines and limitations apply to access control lists (ACLs):

- 12.2 (33) SRD4 release of the Private Hosts feature uses Protocol Independent MAC ACLs. Do not apply IP-based ACLs to any port configured for Private Hosts or you will break the Private Hosts feature (because the router will not be able to apply a Private Hosts MAC ACL to the port).
- VLAN ACLs (VACLs) and port ACLs cannot both be applied to the same interface.
- Routing ACLs (RACLs) and PACLs cannot both be applied to the same interface. However, you can apply ACLs to separate interfaces.

VLANs on the Trunk Port

The following guidelines and limitations apply to VLANs:

- You can enable IGMP snooping on VLANs that use trunk ports configured for Private Hosts.
- You cannot enable IP multicast on a VLAN that uses a trunk port that is configured for Private Hosts.
- Because PACLs operate in override mode on trunk ports, you cannot apply VLAN-based features to switchports.
- The Multicast VLAN Registration (MVR) feature can co-exist with Private Hosts as long as the multicast source exists on a promiscuous port.

Interaction with Other Features

The following list describes how the Private Hosts feature interacts with other features that are configured on the router:

- Private Hosts feature does not affect Layer 2 based services such as MAC limiting, unicast flood protection (UFP), or unknown unicast flood blocking (UUFB).
- Private Hosts feature does not affect IGMP snooping. However, if IGMP snooping is globally disabled, IGMP control packets will be subject to ACL checks. To permit IGMP control packets, the Private Hosts software adds a multicast permit statement to the PACLs for isolated hosts. Note that this behavior occurs automatically and no user intervention is required.
- Port security can be enabled on isolated ports to provide added security to those ports.
- When enabled on promiscuous or mixed-mode ports, the port security feature may restrict a change in source port for upstream devices (such as BRAS or multicast servers).
- When enabled on an access port, 802.1*x* is not affected by the Private Hosts.

Spoofing Protection

The Private Hosts feature prevents MAC address spoofing but does not validate the customer MAC or IP address. To prevent MAC address spoofing the Private Hosts feature:

- Uses a static MAC address for the BRAS and multicast servers.
- Disables learning in the Layer 2 (L2) forwarding table.
- Alerts the router software when a BRAS or multicast server moves from one source port to another. The software then validates the move, and updates the L2 forwarding table.

Multicast Operation

Multicast traffic that originates from upstream devices (such as BRAS or multicast servers) is always permitted. In addition, the Private Hosts PACLs are not applied to multicast control packets (such as IGMP query and join requests). This behavior allows isolated hosts to participate in multicast groups, respond to IGMP queries, and receive traffic from any groups of interest.

Multicast traffic that originates from a host is dropped by the Private Hosts PACLs. However, if other hosts need to receive multicast traffic originating from a host, Private Hosts does the following adds a *multicast permit* entry to the PACLs.

Configuring Private Hosts

The following sections provide information about configuring the Private Hosts feature on a Cisco 7600 series router and instructions for configuring the feature:

Configuration Summary

This section provides a summary of the steps to perform to configure the Private Hosts feature on Cisco 7600 routers. Detailed configuration instructions follow in the next section.

1. Determine which router ports (interfaces) to use for the Private Hosts feature. You can configure the feature on switchports (802.1q or ISL trunk ports) or port-channel interfaces (Etherchannel, FastEtherchannel, and GigabitEtherchannel). Note that Private Hosts must be enabled on the port-channel interface; you cannot enable the feature on member ports.
2. Configure each port (interface) for normal, non-Private Hosts service. Note that you can configure the VLANs at this point or later.
3. Determine which VLAN or set of VLANs will be used to deliver broadband services to end-users. The Private Hosts feature will provide Layer 2 isolation among the hosts in these VLANs.
4. Identify the MAC addresses of all Broadband Remote Access Servers (BRAS) and multicast servers that are being used to provide broadband services to end-users (isolated hosts).

Note If a server is not connected directly to the router, determine the MAC address of the core network device that provides access to the server.

5. (Optional) If you plan to offer different types of broadband services to different sets of isolated hosts, create multiple MAC and VLAN lists.
   - Each MAC address list identifies a server or set of servers providing a particular type of service.
   - Each VLAN list identifies the isolated hosts where that service to be delivered.
6. Configure promiscuous ports and specify a MAC and VLAN list to identify the server and receiving hosts for a particular type of service.

Note You can specify multiple MAC and VLAN combinations to allow for different types of services to be delivered to different sets of hosts. For example, the BRAS at xxxx.xxxx.xxxx could be used to deliver a basic set of services over VLANs 20, 25, and 30, and the BRAS at yyyy.yyyy.yyyy could be used to deliver a premium set of services over VLANs 5, 10, and 15.

7. Globally enable Private Hosts.
8. Enable Private Hosts on individual ports (interfaces) and specify the mode in which the port is to operate. To determine port mode, you need to know whether the port faces upstream (toward content servers or core network), faces downstream (toward DSLAM and isolated hosts), or is connected to another Cisco 7600 router (typically, in a ring topology). See Restricting Traffic Flow (Using Private Hosts Port Mode and PACLs), page 35-3.

After you enable the feature on individual ports, the router is ready to run the Private Hosts feature. The Private Hosts software uses the MAC and VLAN lists you defined to create the isolated, promiscuous, and mixed-mode PACLs for your configuration. The software then applies the appropriate PACL to each Private Hosts port based on the port's mode.

Detailed Configuration Steps

Perform the following steps to configure the Private Hosts feature. Note that these steps assume that you have already configured the Layer 2 interfaces that you plan for Private Hosts. See the "Command Reference" section on page 35-13 for detailed descriptions of the commands listed in the following table.

Note You can configure Private Hosts only on switchports (802.1q or ISL trunk ports) or Etherchannel ports. In addition, you must enable Private Hosts on all of the routers between the DSLAMs and upstream devices.

	Command or Action	Purpose
Step 1	`Router(config)# configure terminal`	Enters global configuration mode.

Step 2	`Router(config)# private-hosts mac-list mac-list-name mac-address [remark device-name | comment]`
	
	Example:
	`Router(config)# private-hosts mac-list BRAS_list 0000.1111.1111 remark BRAS_SanJose`
		Creates a list of MAC addresses that identify the BRAS and multicast servers providing broadband services, where:
		• *mac-list-name* specifies a name to assign to this list of content servers.
		• *mac-address* identifies the BRAS or multicast server (or set of servers) providing a particular broadband service or set of services.
		• **remark** allows you to specify an optional device name or comment to assign to this MAC list.
		Specify the MAC address of every content server being used to deliver services. If you plan to offer different types of services to different sets of hosts, create a separate MAC list for each server or set of servers providing a particular service.
		Note If a server is not directly connected to the router, specify the MAC address of the core network device that provides access to the server.

Step 3	`Router(config)# private-hosts vlan-list vlan-ids`
	
	Example:
	`Router(config)# private-hosts vlan-list 10,12,15,200-300`
		Creates a list of the VLANs (*vlan-ids*) whose hosts need to be isolated so that the hosts can receive broadband services.
		Create separate VLAN lists if you plan to offer particular services to different sets of hosts. Otherwise, all of the broadband services will be delivered to all isolated hosts.

Step 4	`Router(config)# private-hosts promiscuous mac-list-name [vlan-list vlan-ids]`
	
	Example:
	`Router(config)# private-hosts promiscuous BRAS_list vlan-list 1,2,3`
		Identifies the content servers for broadband services and the end users (isolated hosts) to deliver the services to, where:
		• *mac-list-name* specifies the name of the MAC address lists that identifies the BRAS or multicast server (or set of servers) providing a particular type of broadband service or set of services.
		• *vlan-ids* identifies the VLAN or set of VLANs whose hosts are to receive services from the above servers. If no VLAN list is specified, the software uses the global VLAN list (configured in Step 3).
		Note You can issue this command multiple times to configure multiple MAC and VLAN combinations, each defining the server and receiving hosts for a particular type of service.

Step 5	`Router(config)# private-hosts`	Globally enables Private Hosts on the router.
Step 6	`Router(config)# interface interface`	Selects the switchport (802.1Q or ISL trunk port) or Etherchannel port to enable for Private Hosts.

	Command or Action	Purpose

Step 7	`Router(config-if)# private-hosts mode {promiscuous | isolated | mixed}`
	
	Example:
	`Router(config-if)# private-hosts mode isolated`
		Enables Private Hosts on the port. Use one of the following keywords to define the mode that the port is to operate in:
		• **promiscuous**—upstream-facing ports that connect to broadband servers (BRAS, multicast, or video) or to core network devices providing access to the servers.
		• **isolated**—ports that connect to DSLAMs.
		• **mixed**—ports that connect to other Cisco 7600 routers, typically in a ring topology.
		Note You must perform this step on each port being used for Private Hosts.

Step 8	`Router(config-if)# end`	Exits interface and global configuration modes and returns to privileged EXEC mode. Private Hosts configuration is complete.

Configuration Examples

The following example shows the interface configuration of a Private Hosts isolated port:

```
Router# show run int gi 5/2
Building configuration...

Current configuration : 200 bytes
!
interface GigabitEthernet5/2
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 private-hosts mode isolated
end
```

The following example shows the interface configuration of a Private Hosts promiscuous port:

```
Router# show run int gi 4/2
Building configuration...

Current configuration : 189 bytes
!
interface GigabitEthernet4/2
 switchport
 switchport access vlan 200
 switchport mode access
 private-hosts mode promiscuous
end

private-hosts
private-hosts vlan-list 200
private-hosts promiscuous bras-list
private-hosts mac-list bras-list 0000.1111.1111 remark BRAS-SERVER
```

Command Reference

This section documents the commands related to the Private Hosts feature introduced in release 12.2(33) SRD4:

- **private-hosts**
- **private-hosts mac-list**
- **private-hosts mode**
- **private-hosts promiscuous**
- **private-hosts vlan-list**
- **show fm private-hosts**
- **show private-hosts access-lists**
- **show private-hosts configuration**
- **show private-hosts interface configuration**
- **show private-hosts mac-list**
- **debug fm private-hosts**
- **debug private-hosts**

private-hosts

To globally enable the Private Hosts feature, use the **private-hosts** command in global configuration mode. Use the **no** form of the command to disable the feature.

private-hosts

no private-hosts

Syntax Description

This command has no arguments or keywords.

Defaults

This command is disabled by default.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Usage Guidelines

Use this command to enable Private Hosts on the router. Then, use the **private-hosts mode** command to enable Private Hosts on individual interfaces (ports).

Examples

The following command example globally enables the Private Hosts feature on the router:

```
Router(config)# private-hosts
```

Related Commands

Command	Description
private-hosts mac list	Creates a MAC address list that identifies the content servers that are being used to provide broadband services to isolated hosts.
private-hosts mode	Specifies the operating mode for a Private Hosts port.
private-hosts promiscuous	Identifies the content servers and receiving hosts for broadband services.
private-hosts vlan-list	Identifies the VLANs whose hosts need to be isolated.
show private-hosts configuration	Displays Private Hosts configuration information for the router.
show private-hosts interface configuration	Displays Private Hosts configuration information for individual interfaces.

private-hosts mac-list

Identifies the content servers that provide broadband services to isolated hosts and create a MAC address list, use the **private-hosts mac-list** command in global configuration mode. To delete an address from the MAC address list and remove that device from the list of content servers providing services for the Private Hosts feature, use the **no** form of the command.

private-hosts mac-list *mac-list-name mac-address* [**remark** *device-name* | *comment*]

no private-hosts mac-list *mac-list-name mac-address*

Syntax Description

mac-list-name	A name assigned to the address list (up to 80 characters).
mac-address	The MAC address of a Broadband Remote Access Server (BRAS), multicast server, or video server that provides broadband services for the Private Hosts feature. **Note** If the server is not directly connected to the router, specify the MAC address of the core network device that provides access to the server.
remark *device-name* | *comment*	(Optional) Specifies an optional device name or comment to assign to this MAC address list.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Usage Guidelines

This command creates a list of MAC addresses that identify the content servers being used to provide broadband services to isolated hosts in the Private Hosts configuration.

Use this command to specify the MAC address of every content server that provides broadband services for the Private Hosts feature. A *content server* is any Broadband Remote Access Server (BRAS), multicast server, or video server that provides services to the isolated hosts in your network.

You can assign all the content servers to a single MAC address list, or you can create multiple MAC address lists, each identifying the content server providing a particular type of broadband service or set of services. When you configure the promiscuous ports for Private Hosts, you specify a MAC address list and VLAN list to identify the server and receiving hosts for broadband services.

Note The MAC address list is automatically deleted when the last address in the list is deleted.

Examples

This example creates a MAC address list named BRAS_list that identifies the MAC address of the upstream BRAS. The optional remark indicates that the BRAS is in San Jose.

```
Router(config)# private-hosts mac-list BRAS_list 0000.1111.1111 remark BRAS_San-Jose
```

Related Commands

Command	Description
show private-hosts mac-list	Displays a list of the MAC addresses that identify the content servers that are providing broadband defined for Private Hosts.

private-hosts mode

To enable Private Hosts on an interface (port) and specify the mode in which the port is to operate, use the **private-hosts mode** command in interface configuration mode. Use the **no** form of the command to disable Private Hosts on the port.

private-hosts mode {**promiscuous** | **isolated** | **mixed**}

no private-hosts

Syntax Description

promiscuous	Configures the port for promiscuous mode. Use this mode for ports that face upstream traffic. These are the ports that connect the router to servers providing broadband services (BRAS, multicast, or video), or to core network devices providing access to the servers.
isolated	Configures the port for isolated mode. Use this mode for ports that face the DSLAM to which isolated hosts are connected.
mixed	Configures the port for mixed mode. Use this mode for ports that connect to other Cisco 7600 routers, typically in a ring topology. The behavior of this port can change depending on the Spanning Tree Protocol (STP) topology.

Defaults

This command is disabled by default.
The default for **mode** is promiscuous.

Command Modes

Interface configuration (switchport or port-channel)

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Usage Guidelines

Before you use this command, you must globally enable the Private Hosts feature on the router by issuing the **private-hosts** command.

Use this command to enable the Private Hosts feature on individual ports and to define the mode of operation for the port. A port's mode determines which type of PACL is assigned to the port in order to restrict the type of traffic that is allowed to pass through the port. Each type of PACL restricts the traffic flow for a different type of traffic (for example, from content servers to isolated hosts, from isolated hosts to servers, and traffic between isolated hosts). Use the **show private-hosts interface configuration** command to display the mode assigned to Private Hosts ports.

Examples

The following command example enables Private Hosts on an interface (port) and configures the port for isolated mode:

```
Router(config-if)# private-hosts mode isolated
```

Related Commands

Command	Description
show private-hosts interface configuration	Displays Private Hosts configuration information for individual interfaces.

private-hosts promiscuous

To identify the content servers and receiving hosts that provide broadband services, use the **private-hosts promiscuous** command in global configuration mode. Use the **no** form of the command to remove a promiscuous ports setting.

private-hosts promiscuous *mac-list-name* [**vlan** *vlan-ids*]

no private-hosts promiscuous *mac-list-name*

Syntax Description

mac-list-name	The name of MAC address list that identifies the content servers (BRAS, multicast, or video) providing broadband services for Private Hosts.
vlan *vlan-ids*	(Optional) The VLAN or set of VLANs whose hosts will be allowed to receive services from the content servers identified by the MAC address list. Use commas to separate individual VLANs or specify a range of VLANs (for example, 1,3,5,20-25). **Note** If no VLAN list is specified, the global VLAN list is used.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Usage Guidelines

The MAC address list and the VLAN list define the content servers and receiving hosts that provide broadband services. If no VLAN list is specified, the system uses the global VLAN list created with the **private-hosts vlan-list** command.

You can issue this command multiple times to specify multiple combinations of MAC and VLAN lists, each defining the server and receiving hosts for a particular type of service. For example, the BRAS at xxxx.xxxx.xxxx could be used to deliver a basic set of services over VLANs 20, 25, and 30, and the BRAS at yyyy.yyyy.yyyy could be used to deliver a premium set of services over VLANs 5, 10, and 15.

Examples

The following command example configures the broadband services provided by content servers defined in the BRAS_list address list, to be delivered to isolated hosts in VLANs 10, 12, 15, and 200 through 300:

```
Router(config)# private-hosts promiscuous BRAS_list vlan 10,12,15,200-300
```

Related Commands

Command	Description
show private-hosts configuration	Displays Private Hosts configuration information for the router.
show private-hosts interface configuration	Displays Private Hosts configuration information for individual interfaces.

private-hosts vlan-list

Identifies the VLANs whose hosts need to be isolated from each other (so that the VLANs can be used to deliver broadband services), use the **private-hosts vlan-list** command in global configuration mode. Use the **no** form of the command to remove a VLAN from the list of VLANs requiring host isolation.

private-hosts vlan-list *vlan-ids*

no private-hosts vlan-list *vlan-ids*

Syntax Description

vlan-ids	A list of the VLANs whose hosts need to be isolated from each other. Use commas to separate individual VLANs or specify a range of VLANs (for example, 1,3,5,20-25).

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Usage Guidelines

This command creates a list of VLANs whose hosts need to be isolated through the Private Hosts feature. The VLAN list should include all of the VLANs that are being used to deliver broadband services to multiple end-users (isolated hosts).

If you plan to deliver different types of broadband services to different sets of hosts, you can create multiple VLAN lists and multiple MAC address lists. When you configure promiscuous ports, you can specify different combinations of MAC and VLAN lists to identify content servers and receiving hosts for each type of service.

If you do not specify a VLAN list when you configure promiscuous ports, the system uses the global VLAN list created by this command.

Note The Private Hosts feature isolates the hosts in all VLANs included in the VLAN lists; therefore, VLAN lists should include only those VLANs that are being used to deliver broadband services.

Examples

This command configures the Private Hosts feature to isolate the hosts in VLANs 10, 12, 15, and 200 through 300:

```
Router(config)# private-hosts vlan-list 10,12,15,200-300
```

Related Commands

Command	Description
show private-hosts configuration	Displays Private Hosts configuration information for the router.

show fm private-hosts

To display information about the Private Hosts feature manager, use the **show fm private-hosts** command in privileged EXEC mode.

show fm private-hosts {**all** | **interface** *intf*}

Syntax Description

all	Displays the feature manager information for all of the interfaces that are configured for Private Hosts.
interface *intf*	Specifies the interface where the feature manager information is displayed.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Examples

The following example shows sample command output:

```
Router# show fm private-hosts interface GigabitEthernet 1/2
------------------------------------------------------------------------------
FM_FEATURE_PVT_HOST_INGRESS      i/f: Gi1/2      map name:
PVT_HOST_ISOLATED
==============================================================================

------------------------------------------------------------
MAC Seq. No: 10          Seq. Result : PVT_HOSTS_ACTION_DENY
------------------------------------------------------------
Indx - VMR index      T     - V(Value)M(Mask)R(Result)
EtTy - Ethernet Type  EtCo  - Ethernet Code
+----+-+--------------+--------------+----+----+
|Indx|T|   Dest Node  |  Source Node |EtTy|EtCo|
+----+-+--------------+--------------+----+----+

 1    V 0000.0000.0000 0000.1111.4001    0 0
      M 0000.0000.0000 ffff.ffff.ffff    0 0
      TM_PERMIT_RESULT

 2    V 0000.0000.0000 0000.0000.0000    0 0
      M 0000.0000.0000 0000.0000.0000    0 0
      TM_L3_DENY_RESULT

------------------------------------------------------------
MAC Seq. No: 20          Seq. Result : PVT_HOSTS_ACTION_PERMIT
------------------------------------------------------------
+----+-+--------------+--------------+----+----+
|Indx|T|   Dest Node  |  Source Node |EtTy|EtCo|
```

```
+----+-+--------------+--------------+----+----+

 1    V 0000.1111.4001 0000.0000.0000    0 0
      M ffff.ffff.ffff 0000.0000.0000    0 0
      TM_PERMIT_RESULT

 2    V 0000.0000.0000 0000.0000.0000    0 0
      M 0000.0000.0000 0000.0000.0000    0 0
      TM_L3_DENY_RESULT

------------------------------------------------------------
MAC Seq. No: 30          Seq. Result : PVT_HOSTS_ACTION_REDIRECT
------------------------------------------------------------
+----+-+--------------+--------------+----+----+
|Indx|T|   Dest Node  |  Source Node |EtTy|EtCo|
+----+-+--------------+--------------+----+----+

 1    V ffff.ffff.ffff 0000.0000.0000    0 0
      M ffff.ffff.ffff 0000.0000.0000    0 0
      TM_PERMIT_RESULT

 2    V 0000.0000.0000 0000.0000.0000    0 0
      M 0000.0000.0000 0000.0000.0000    0 0
      TM_L3_DENY_RESULT

------------------------------------------------------------
MAC Seq. No: 40          Seq. Result : PVT_HOSTS_ACTION_PERMIT
------------------------------------------------------------
+----+-+--------------+--------------+----+----+
|Indx|T|   Dest Node  |  Source Node |EtTy|EtCo|
+----+-+--------------+--------------+----+----+

 1    V 0100.5e00.0000 0000.0000.0000    0 0
      M ffff.ff80.0000 0000.0000.0000    0 0
      TM_PERMIT_RESULT

 2    V 3333.0000.0000 0000.0000.0000    0 0
      M ffff.0000.0000 0000.0000.0000    0 0
      TM_PERMIT_RESULT

 3    V 0000.0000.0000 0000.0000.0000    0 0
      M 0000.0000.0000 0000.0000.0000    0 0
      TM_L3_DENY_RESULT

------------------------------------------------------------
MAC Seq. No: 50          Seq. Result : PVT_HOSTS_ACTION_DENY
------------------------------------------------------------
+----+-+--------------+--------------+----+----+
|Indx|T|   Dest Node  |  Source Node |EtTy|EtCo|
+----+-+--------------+--------------+----+----+

 1    V 0000.0000.0000 0000.0000.0000    0 0
      M 0000.0000.0000 0000.0000.0000    0 0
      TM_PERMIT_RESULT

 2    V 0000.0000.0000 0000.0000.0000    0 0
      M 0000.0000.0000 0000.0000.0000    0 0
      TM_L3_DENY_RESULT


Interfaces using this pvt host feature in ingress dir.:
-----------------------------------------------
  Interfaces (I/E = Ingress/Egress)
```

```
Router#
```

Related Commands

Command	Description
show private-hosts configuration	Displays Private Hosts configuration information for the router.
show private-hosts interface configuration	Displays Private Hosts configuration information for individual interfaces.

show private-hosts access-lists

To display the access lists for a Private Hosts configuration, use the **show private-hosts access-lists** command in privileged EXEC mode.

show private-hosts access-lists

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Examples

The following example shows how to display the Private Hosts access lists for the customized configuration:

```
Router# show private-hosts access-lists

Promiscuous ACLs
Action Permit   Sequence # 010
    Source:0000.1111.4001 0000.0000.0000 Destination:0000.0000.0000 ffff.ffff.ffff
Action Deny     Sequence # 020
    Source:0000.0000.0000 ffff.ffff.ffff Destination:0000.0000.0000 ffff.ffff.ffff

Isolated ACLs
Action Deny     Sequence # 010
    Source:0000.1111.4001 0000.0000.0000 Destination:0000.0000.0000 ffff.ffff.ffff
Action Permit   Sequence # 020
    Source:0000.0000.0000 ffff.ffff.ffff Destination:0000.1111.4001 0000.0000.0000 Action
Redirect Sequence # 030 Redirect index 6
    Source:0000.0000.0000 ffff.ffff.ffff Destination:ffff.ffff.ffff 0000.0000.0000
Action Permit   Sequence # 040
    Source:0000.0000.0000 ffff.ffff.ffff Destination:0100.5e00.0000 0000.007f.ffff
    Source:0000.0000.0000 ffff.ffff.ffff Destination:3333.0000.0000 0000.ffff.ffff
Action Deny     Sequence # 050
    Source:0000.0000.0000 ffff.ffff.ffff Destination:0000.0000.0000 ffff.ffff.ffff

Mixed ACLs
Action Permit   Sequence # 010
    Source:0000.1111.4001 0000.0000.0000 Destination:ffff.ffff.ffff 0000.0000.0000 Action
Redirect Sequence # 020 Redirect index 6
    Source:0000.0000.0000 ffff.ffff.ffff Destination:ffff.ffff.ffff 0000.0000.0000
Action Permit   Sequence # 030
    Source:0000.1111.4001 0000.0000.0000 Destination:0000.0000.0000 ffff.ffff.ffff
Action Permit   Sequence # 040
    Source:0000.0000.0000 ffff.ffff.ffff Destination:0000.1111.4001 0000.0000.0000
Action Deny     Sequence # 050
```

```
    Source:0000.0000.0000 ffff.ffff.ffff Destination:0000.0000.0000 ffff.ffff.ffff

Router#
```

Related Commands

Command	Description
show fm private-hosts	Displays information about the Private Hosts feature manager.
show private-hosts configuration	Displays Private Hosts configuration information for the router.
show private-hosts interface configuration	Displays Private Hosts configuration information for individual interfaces.

show private-hosts configuration

To display information about the Private Hosts configuration on the router, use the **show private-hosts configuration** command in privileged EXEC mode.

show private-hosts configuration

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Examples

The following example shows sample command output:

```
Router# show private-hosts configuration

    Private hosts enabled. BR INDEX 6 State 0000000F
    Privated hosts vlans lists:
    200
    Privated promiscuous MAC configuration:
    A '*' mark behind the mac list indicates non-existant mac-list
    -------------------------------------------------------------------------------
    MAC-list                      VLAN list
    -------------------------------------------------------------------------------
    bras-list                    *** Uses the isolated vlans (if any) ***
```

Related Commands

Command	Description
show private-hosts interface configuration	Displays Private Hosts configuration information for individual interfaces.

show private-hosts interface configuration

To display information about the Private Hosts configuration on individual interfaces (ports), use the **show private-hosts interface configuration** command in privileged EXEC mode.

show private-hosts interface configuration

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Examples

The following example shows sample command output:

```
Router# show private-hosts interface configuration

    Private hosts enabled
    Debug Events: 0 Acl: 0 API: 0
    Promiscuous interface list
    --------------------------
    GigabitEthernet4/2
    Isolated interface list
    -----------------------
    GigabitEthernet5/2
    Mixed mode interface list
    -------------------------
```

Related Commands

Command	Description
show private-hosts configuration	Displays Private Hosts configuration information for the router.

show private-hosts mac-list

To display the contents of the MAC address lists defined for Private Hosts, use the **show private-hosts mac-list** command in privileged EXEC mode.

show private-hosts mac-list [*list-name*]

Syntax Description

list-name	(Optional) The name of the MAC address list whose contents you want to display.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Examples

The following example shows sample command output:

```
Router# show private-hosts mac-list

    MAC-List: bras-list
    ----------------------------------------------------------------
    MAC address    Description
    ----------------------------------------------------------------
    0000.1111.1111 BRAS-SERVER
```

Related Commands

Command	Description
private-hosts mac-list	Creates a MAC address list that identifies a content server that is being used to provide broadband services to isolated hosts.

debug fm private-hosts

To enable debug messages for the Private Hosts feature manager, use the **debug fm private-hosts** command in privileged EXEC mode.

debug fm private-hosts {**all** | **vmr** | **unusual** | **events**}

Syntax Description

all	Enables debug messages for all Private Hosts errors and events.
vmr	Enables debug messages for the Multicast VLAN Registration (MVR) feature.
unusual	Enables debug messages for unexpected Private Hosts behavior.
events	Enables debug messages for Private Hosts events.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Examples

The following example shows sample command output:

```
Router# debug fm private-hosts all
fm private-hosts vmr debugging is on
fm private-hosts unusual debugging is on
fm private-hosts events debugging is on
Router#
```

Related Commands

Command	Description
debug private-hosts	Enables debug messages for Private Hosts.

debug private-hosts

To enable debug messages for the Private Hosts feature, use the **debug private-hosts** command in privileged EXEC mode.

debug private-hosts {**all** | **events** | **acl** | **api**}

Syntax Description

all	Enables debug messages for all Private Hosts errors and events.
events	Enables debug messages for issues related to Private Hosts events.
acl	Enables debug messages for issues and events related to ACLs.
api	Enables debug messages for issues related to the application programming interface.

Defaults

This command has no default settings.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(33)SRB	This command was introduced.

Examples

The following example shows sample command output:

```
Router# debug private-hosts all
private-hosts events debugging is on
private-hosts api debugging is on
private-hosts acl debugging is on
Router#
```

Related Commands

Command	Description
debug fm private-hosts	Enables debug messages for the Private Hosts feature manager.

CHAPTER 39

Configuring Denial of Service Protection

This chapter contains information on how to protect your Cisco 7600 series router against Denial of Service (DoS) attacks. The information covered in this chapter is unique to the Cisco 7600 series routers, 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
- The Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding How DoS Protection Works

The following sections contain an overview of the DoS protection on the Cisco 7600 series router and describe some types of DoS attack scenarios:

DoS Protection with a PFC3

This section contains information about the available methods to counteract DoS attacks with a PFC3 and includes configuration examples. The PFC3 provides a layered defense against DoS attacks using the following methods:

- CPU rate limiters—Controls traffic types.
- Control plane policing (CoPP)—Filters and rate limits control plane traffic. For information about CoPP, see the "Understanding How Control Plane Policing Works" section on page 39-19.

These sections describe DoS protection with a PFC3:

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 router interface that is pointing to the Internet. You can apply an inbound ACL on the router 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 router interface, it matches on that ACL and drops the packet before it causes damage.

When the Cisco 7600 series router 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 Cisco 7600 series routers.

QoS Rate Limiting

QoS ACLs limit the amount of a particular type of traffic that is processed by the MSFC3. If a DoS attack is initiated against the MSFC, QoS ACLs can prevent the DoS traffic from reaching the MSFC data path and congesting it. The PFC3 performs QoS in hardware, which offers an efficient means of limiting DoS traffic (once that traffic has been identified) to protect the router from impacting the MSFC.

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 MSFC 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 PFC3 in hardware. Packets that are denied in the uRPF ACL are forwarded in hardware. Packets that are permitted are sent to the CPU.

The uRPF check with a PFC3 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 MSFC 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 Cisco 7600 series router 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 MSFC 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 router 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 all PFC3 types.

ARP Policing

During an attack, malicious users may try to overwhelm the MSFC 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 configured 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 MSFC CPU against attacks such as line-rate ARP attacks, it does not only police routing protocols and ARP packets to the router 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)
```

Hardware-Based Rate Limiters on the PFC3

The PFC3 supports additional hardware-based rate limiters. The PFC3 provides eight rate-limiter registers for the new rate limiters, which are configured globally on the router. These rate-limiter registers are present in the Layer 3 forwarding engine (PFC) and are responsible for containing rate-limiting information for result packets that match the various available configured rate limiters.

Because eight rate-limiter registers are present on the PFC3, these registers can force different rate-limiting scenarios to share the same register. The registers are assigned on a first-come, first-serve basis. If all registers are being utilized, the only way to configure another rate limiter is to free one register.

The hardware-based rate limiters available on the PFC3 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

Shared Rate-Limiters

These shared rate-limiters can be configured on the router:

- IP RPF failure
- ICMP unreachable no-route
- ICMP unreachable acl-drop
- IP errors

If you enable/disbale one of the share limiter, all the other shared limiters are enable/disabled.

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 Cisco 7600 series router, reduces the need for redirects.

- TTL-Failure and CEF-Glean rate-limiters are not enabled by default. It is recommended to enable these rate-limiters.
- 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.

Ingress-Egress ACL Bridged Packets (Unicast Only)

This rate limiter rate limits packets sent to the MSFC because of an ingress/egress ACL bridge result. The router accomplishes this by altering existing and new ACL TCAM entries with a TCAM bridge result to a Layer 3 redirect result pointing to the MSFC. 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 router 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 MSFC 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 MSFC. The uRPF check rate limiters allow you to rate limit the packets per second that are bridged to the MSFC CPU when a uRPF check failure occurs.

This example shows how to rate limit the uRPF check failure packets sent to the MSFC 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 MSFC 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 MSFC). The ICMP unreachable rate limiter allows you to rate limit the packets that are sent to the MSFC containing unreachable addresses.

This example shows how to rate limit the packets that are sent to the MSFC 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 MSFC 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 MSFC. This situation occurs when traffic enters a port and contains the destination of a host on a subnet that is locally connected to the MSFC, 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 MSFC 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 MSFC 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 MSFC. For these security features, you need to rate limit the number of these packets being sent to the MSFC 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 router 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 MSFC may be overwhelmed. Rate limiting would be advantageous in this situation.

IPSec and inspection are also done by the MSFC 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 MSFC 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 router, the MSFC sends ICMP-redirect messages to the host to correct its sending path. If this traffic occurs continuously, and is not rate limited, the MSFC 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 MSFC 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 MSFC does the logging. When VACL logging is configured on the router, 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 MSFC CPU. This might cause the MSFC to be overwhelmed.

This example shows how to rate limit packets failing the MTU failures from being sent to the MSFC 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 Cisco 7600 series router is operating in truncated mode. The router uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and non fabric-enabled modules installed. In this mode, the router 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 MSFC CPU. You cannot enable the Layer 2 PDU rate limiter if the Cisco 7600 series router is operating in truncated mode. The router uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and non fabric-enabled modules installed. In this mode, the router 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 Cisco 7600 series router is operating in truncated mode. The router uses truncated mode for traffic between fabric-enabled

modules when there are both fabric-enabled and non fabric-enabled modules installed. In this mode, the router sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.

This example shows how to rate limit Layer 2 protocol tunneling packets to 10000 pps with a burst of 10 packets:

```
Router(config)# mls rate-limit layer2 l2pt 10000 10
```

IP Errors

This rate limiter limits the packets with IP checksum and length errors. When a packet reaches the PFC3 with an IP checksum error or a length inconsistency error, it must be sent to the MSFC 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 MSFC 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 MSFC3 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 MSFC3. For this reason, it may be desirable to rate limit the flow destined to the MSFC3 for forwarding and replication, which might otherwise increase CPU utilization.

The multicast directly connected rate limiter limits the multicast packets from directly connected sources.

This example shows how to rate limit the multicast packets to 30000 pps with a burst of 30:

```
Router(config)# mls rate-limit multicast ipv4 connected 30000 30
```

The **ip-option** keyword and the ip-option rate limiter are supported with a PFC3B, PFC3BXL, PFC3C, or PFC3CXL only.

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 39-1 lists the IPv6 rate limiters and the class of traffic that each rate limiter serves.

Table 39-1 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
```

MLS Rate-limiter Default Configuration

Table 39-2 shows the DoS protection default configuration for the PFC3 hardware-based rate limiters.

Table 39-2 PFC3 Hardware-based Rate Limiter Default Setting

Rate Limiter	Default Status (ON/OFF)	Default Value
Ingress/Egress ACL Bridged Packets	OFF	
RPF Failures	ON	100 pps, burst of 10 packets
FIB Receive cases	OFF	
FIB Glean Cases	OFF	
Layer 3 Security features	OFF	
ICMP Redirect	OFF	
ICMP Unreachable	ON	100 pps, burst of 10 packets
VACL Log	ON	2000 pps, burst of 10 packets
TTL Failure	OFF	
MTU Failure	OFF	
Layer 2 PDU	OFF	
Layer 2 Protocol Tunneling	OFF	
IP Errors	ON	100 pps, burst of 10 packets
Multicast IGMP	OFF	
Multicast FIB-Miss	ON	100000 pps, burst of 100 packets
Multicast Partial-SC	ON	100000 pps, burst of 100 packets
Multicast Directly Connected	OFF	
Multicast Non-RPF	OFF	
Multicast IPv6	ON	If the *packets-in-burst* is not set, a default of **100** is programmed for multicast cases.

DoS Protection Configuration Guidelines and Restrictions

The section contains these configuration guidelines and restrictions:

PFC3

When configuring DoS protection on systems configured with a PFC3, 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 39-19.

- Do not use these rate limiters if multicast is enabled in systems configured with a PFC3A:
  - TTL failure
  - MTU failure
- These rate limiters are supported only on a PFC3B, PFC3BXL, PFC3C, or PFC3CXL:
  - 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.
- In mls qos policer, if confirm action and exceed action are the same (transmit), packets are not dropped. Hence, when you execute the **show policy-map interface** command, you cannot view the packet count in the dropped counters. If exceed action is modified to police, DSCP etc (other than transmit), then the exceed action occurs on the packets.
- Mls rate-limiter statistics are not available as no hardware resource is present in PFC/DFC.

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, PFC3BXL, PFC3C, and PFC3CXL support ACL hit counters in hardware. You can use the **show tcam interface** command to display each entry in the ACL TCAM.

This example shows how to use the **show tcam interface** command to display the number of times the entry was hit:

```
Router# show tcam interface fa5/2 acl in ip detail

---------------------------------------------------------------------------------------------------------------
DPort - Destination Port   SPort - Source Port        TCP-F - U -URG Pro   - Protocol
I     - Inverted LOU       TOS   - TOS Value                - A -ACK rtr   - Router
```

```
MRFM  - M -MPLS Packet     TN    - T -Tcp Control           - P -PSH COD   - C -Bank Care Flag
      - R -Recirc. Flag          - N -Non-cachable          - R -RST       - I -OrdIndep. Flag
      - F -Fragment Flag   CAP   - Capture Flag             - S -SYN       - D -Dynamic Flag
      - M -More Fragments  F-P   - FlowMask-Prior.          - F -FIN T     - V(Value)/M(Mask)/R(Result)
X     - XTAG               (*)   - Bank Priority
-------------------------------------------------------------------------------------------------------


Interface: 1018   label: 1   lookup_type: 0
protocol: IP   packet-type: 0

+-+-----+---------------+---------------+---------------+---------------+-------+---+----+-+---+--+---+---+
|T|Index|  Dest Ip Addr | Source Ip Addr|     DPort     |     SPort     | TCP-F|Pro|MRFM|X|TOS|TN|COD|F-P|
+-+-----+---------------+---------------+---------------+---------------+------+---+----+-+---+--+---+---+
 V 18396         0.0.0.0         0.0.0.0       P=0             P=0       ------   0 ---- 0   0 -- --- 0-0
 M 18404         0.0.0.0         0.0.0.0         0               0               0 ---- 0   0
 R rslt: L3_DENY_RESULT                rtr_rslt: L3_DENY_RESULT


 V 36828         0.0.0.0         0.0.0.0       P=0             P=0        ------  0 ---- 0   0 -- --- 0-0
 M 36836         0.0.0.0         0.0.0.0         0               0               0 ---- 0   0
 R rslt: L3_DENY_RESULT (*)            rtr_rslt: L3_DENY_RESULT (*)
Router#
```

You can also use the TTL and IP options counters to monitor the performance of the Layer 3 forwarding engine.

This example shows how to use the **show mls statistics** command to display packet statistics and errors associated with the Layer 3 forwarding engine:

```
Router# show mls statistics

Statistics for Earl in Module 6

L2 Forwarding Engine
  Total packets Switched               : 25583421

L3 Forwarding Engine
  Total packets L3 Switched            : 25433414 @ 24 pps

  Total Packets Bridged                : 937860
  Total Packets FIB Switched           : 23287640
  Total Packets ACL Routed             : 0
  Total Packets Netflow Switched       : 0
  Total Mcast Packets Switched/Routed  : 96727
  Total ip packets with TOS changed    : 2
  Total ip packets with COS changed    : 2
  Total non ip packets COS changed     : 0
  Total packets dropped by ACL         : 33
  Total packets dropped by Policing    : 0

Errors
  MAC/IP length inconsistencies        : 0
  Short IP packets received            : 0
  IP header checksum errors            : 0
  TTL failures                         : 0
<----------------- TTL counters
  MTU failures                         : 0
<------------------MTU failure counters

Total packets L3 Switched by all Modules: 25433414 @ 24 pps
```

Monitoring Dropped Packets Using VACL Capture

The VACL capture feature allows you to direct traffic to ports configured to forward captured traffic. The capture action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets. Only forwarded packets can be captured.

You can use VACL capture to assign traffic from each VLAN to a different interface.

VACL capture does not allow you to send one type of traffic, such as HTTP, to one interface and another type of traffic, such as DNS, to another interface. Also, VACL capture granularity is only applicable to traffic switched locally; you cannot preserve the granularity if you direct traffic to a remote router.

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.
- High CPU utilization occurs when:
  - CoPP rate limits and drops exceeding traffic
  - **mls qos protocol** *protocol* pass-through is configured

  To avoid this, rely on the CoPP to drop excessive traffic and not **mls qos protocol** to relay traffic directly to route processor.
- 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 Cisco 7600 series router by protecting the MSFC from unnecessary or DoS traffic and giving priority to important control plane and management traffic. The PFC3 and DFC3 provide hardware support for CoPP. CoPP works with the PFC3 rate limiters.

The PFC3 supports the built-in "special case" rate limiters that can be used when an ACL cannot classify particular scenarios, such as IP options cases, TTL and MTU failure cases, packets with errors, and multicast packets. When enabling the special-case rate limiters, the special-case rate limiters override the CoPP policy for packets matching the rate-limiter criteria.

The traffic managed by the MSFC is divided into three functional components or *planes*:

- Data plane
- Management plane
- Control plane

The majority of traffic managed by the MSFC 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 and and it is recommended that you enable CoPP. For information on CoPP, see *Control Plane Policing Implementation Best Practices* at http://www.cisco.com/web/about/security/intelligence/coppwp_gs.html#9.

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.
- With PFC3A, egress QoS and CoPP cannot be configured at the same time. In this situation, CoPP is performed in the software. A warning message is displayed to inform you that egress QoS and CoPP cannot be configured at the same time.
- 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 routers. Filtering this traffic could prevent remote access to the router, requiring a console connection.
- PFC3 supports built-in special-case rate limiters, which are useful for situations where an ACL cannot be used (for example, TTL, MTU, and IP options). When you enable the special-case rate limiters, you should be aware that the special-case rate limiters will override the CoPP policy for packets matching the rate-limiter criteria.
- CoPP is not enabled in hardware unless MLS 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.
- When CoPP is configured and a unicast traffic hits the CoPP classification, regardless of trust configured on the input port, packets punted to RP are treated with trust DSCP action. If CoPP is configured, and you want the punted packets marked or trusted based on input port, then execute the platform ip features sequential command on the input port. Since multicast and broadcast traffic do not hit the hardware Copp classifcation, this behaviour is not applicable to multicast and broadcast traffic.
- When you set the policer value, note that the mls qos protocol is supported and impacts the traffic switch in the router.
- The incoming control packets needs to be trusted for them to be prioritized in control-plane SPD, else they end up competing with other data packets being punted to RP and this increases their probability of getting dropped.
- For packets ingressing on LAN interfaces like 6xxx linecards:
  - If a CoPP is not applied on the router, its preferable that either the ingress traffic DSCP/Precedence is trusted using "mls qos trust" or remarking of the incoming control protocol packets to precedence values lower than precedence-6 is avoided. The control protocol packets could be classified based on their precedence or DSCP value.
  - If a CoPP is applied and a unicast raffic hits the CoPP classification, then the CoPP implicitly overrides incoming trust state with trust-dscp and preserves DSCP/Precedence on the packets being punted to control-plane.Multicast and broadcast traffic does not hit the hardware CoPP classification and, hence this is not applicable to multicast or broadcast traffic.
  - For packets ingressing on WAN interfaces like Sip400/ES+ linecards

- Avoid remarking the incoming control protocol packets to precedence values lower than precedence-6 or 7. The control protocol packets can be identified based on their precedence or DSCP value.

For information on classifying CoPP traffic, see Traffic Classification Overview, page 39-23.

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 "Personalizing CoPP" section on page 39-23.

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 (IPv6 ACLs are not supported in hardware).
- 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
```

Personalizing CoPP

CoPP policy applied on a router should be personalized to best fit the router traffic profile getting punted to RP. Personalizing ensures that:

- The right kind of traffic is prioritized over other less priority non priority traffic
- Stabilizes the network
- Achieves control plane protection
- Understand if any trafficis missed in the classification

To personalize the policy, use the Mini Protocol Analyzer tool to analyze the traffic punted to RP. This tool helps you capture traffic being punted to RP and check what is the rate at wich it is being punted.The data obtained can be used to identify the classes and police rates required to set up CoPP. For more information on the Mini Protocol Analyzer tool, see Chapter 56, “Using the Mini Protocol Analyzer”.

Developing a CoPP Policy

Prior to developing a CoPP policy,a required volume of traffic must be identified and separated into different classes. Stratifying traffic into distinct groups based on relative importance is the recommended method:

Here are the sample classification critera used when developing CoPP policer:

- Do not use any policer in class-default .All the potential traffic should be classified in a specific class rather than in class-default.
- For catch-all traffic, use the **match ipv4/ipv6 any class** command. Though class-default serves the same purpose, we recommend you to minimize the traffic with class-default action as shown in this example:

```
Policy-map CoPP
```

```
   Class CLASS1
      Police <>
   Class CLASS2
     Police <>
Class MATCH-IPv4-ANY Match all IPv4 traffic which doesn't fall in any of the above
mentioned classes
     Police <>
   Class MATCH-IPV6-ANY Match all IPV6 traffic which doesn't fall in any of the above
mentioned classes
     Police <>
```

In the section Exampleof a CoPP Policy, page 39-25, traffic is grouped into five different classes. The actual number of classes differs and should be selected based on local requirements and security policies. These traffic classes are defined with regard to the CPU or control plane.

The five different classes are:

- Critical
  - Traffic that is crucial to the operation of the router and the network
  - Examples: routing protocols like Border Gateway Protocol (BGP)
  - Some sites might choose to classify traffic other than the ones crucial to the operation as critical when appropriate
- Important
  - Frequently used traffic that is necessary for day-to-day operations
  - Examples: traffic used for remote network access and management (telnet, Secure Shell (SSH), Network Time Protocol (NTP) and Simple Network Management Protocol (SNMP)
- Normal
  - Traffic that is functional but not essential to network operation
  - Normal traffic used to be particularly hard to address when designing control-plane protection schemes, as it should be permitted but should never pose a risk to the router. With CoPP, this traffic is permitted, but limited to a low rate.
  - Examples: ICMP echo request
- Undesirable
  - Explicitly identifies “bad” or malicious traffic that should be dropped and denied access to the Route Processor
  - Particularly useful when known traffic destined to the router should always be denied and not placed into a default category. Explicitly denying traffic allows the end-user to collect rough statistics on this traffic using the **show** commands and therefore offers some insight into the rate of denied traffic.
- Layer 2 class
  - Traffic used for address resolution protocol (ARP). Excessive ARP packets can potentially monopolize MSFC resources, depriving other important processes of resources; CoPP can be used to rate limit ARP packets to prevent this situation. Currently, ARP is the only Layer 2 protocol specifically classified using the match protocol classification criteria.

- Match-Any class
  - Matches all the other IPv4/IPv6 traffic (which doesn't fall into any of the above class), and police them as appropriate. It is primarily designed so that the class-default doesn't need a policer.
- Default
  - The remaining traffic destined to the Route Processor and has not been identified
  - A default classification helps monitoring of statistics to determine the rate of unidentified traffic destined to the control-plane. The identified traffic can be further analyzed to classifyand if needed updated with the other CoPP policy entries
  - The Sup720 in Release 12.2(18)SXD1 does not support the MQC "class-default" in hardware. The support has been added effective Release 12.2(18)SXE1 software release. Anyway, this is not a big limitation as shown in the example below, where the "class-default" is replaced by a normal class-map.
  - Certain traffic types, namely Layer 2 keepalives, CLNS, and other non-IP packets will be seen by a CoPP (only in class-default). These traffic types cannot be classified by MQC for CoPP and hence, will always fall into the aCoPP class-default class. If aCoPP is configured, it is best practice to never rate limiting class-default so thatLayer 2 keepalives and other essential control plane traffic are not dropped. This is the primary reason for always configuring a "catch all" IP class in the CoPP policy-map just prior to class-default.

Using the classification scheme defined above, commonly used traffic is identified with a series of ACLs:

- Class CoPP-CRITICAL : ACL 120: critical traffic
- Class CoPP-IMPORTANT: ACL 121: important traffic
- Class CoPP-NORMAL : ACL 122: normal traffic
- Class CoPP-UNWANTED ACL 123: explicitly denies unwanted traffic (For example, slammer worm traffic)
- Class CoPP-ARP : Match the ARP protocol
- Class CoPP-Match-all ACL 124: the rest of the traffic

The ACLs build classes of traffic that are used to define the policies.

Sample CoPP Policy

This is an example of a CoPP policy developed using the ACLs above:

```
access-list 120 remark *** ACL for CoPP-Critical ***
access-list 120 remark *** LDP ***
access-list 120 permit udp 172.0.0.0 0.0.255.255 eq 646 any
access-list 120 permit udp 172.0.0.0 0.0.255.255 any eq 646
access-list 120 permit tcp 172.0.0.0 0.0.255.255 eq 646 any
access-list 120 permit tcp 172.0.0.0 0.0.255.255 any eq 646
access-list 120 remark *** BGP ***
access-list 120 permit tcp 172.0.0.0 0.0.255.255 eq bgp any
access-list 120 permit tcp 172.0.0.0 0.0.255.255 any eq bgp
access-list 120 remark *** PIM ***
access-list 120 permit pim 172.0.0.0 0.0.255.255 any
access-list 120 permit pim any 172.0.0.0 0.0.255.255
```

```
access-list 121 remark *** ACL for CoPP-IMPORTANT
access-list 121 remark *** Telnet ***
access-list 121 permit tcp 172.0.0.0 0.0.255.255 172.0.0.0 0.0.255.255 eq telnet
access-list 121 remark *** SSH ***
access-list 121 permit tcp 172.0.0.0 0.0.255.255 152.0.0.0 0.255.255.255 eq 22
access-list 121 remark *** SNMP ***
access-list 121 permit udp 172.0.0.0 0.0.255.255  152.0.0.0 0.255.255.255 eq snmp
access-list 121 remark *** NTP ***
access-list 121 permit udp 172.0.0.0 0.0.255.255  host 192.168.70.10 eq ntp
access-list 121 permit udp 172.0.0.0 0.0.255.255  host 192.168.70.30 eq ntp
access-list 121 remark *** Syslog ***
access-list 121 permit udp 172.0.0.0 0.0.255.255  152.0.0.0 0.255.255.255 eq syslog
access-list 121 remark *** TACAS+ ***
access-list 121 permit tcp 172.0.0.0 0.0.255.255  152.0.0.0 0.255.255.255 eq tacacs
access-list 122 remark CoPP normal traffic
access-list 122 permit icmp any any ttl-exceeded
access-list 122 permit icmp any any port-unreachable

access-list 122 permit icmp any any echo-reply
access-list 122 permit icmp any any echo

access-list 123 remark *** ACL for CoPP-UNDESIRABLE
access-list 123 permit icmp any any fragments
access-list 123 permit udp any any fragments
access-list 123 permit tcp any any fragments
access-list 123 permit ip any any fragments
access-list 124 remark *** ACL for CoPP-Match-all
access-list 124 permit ip any any
access-list 124 permit ipv6 any any

class-map match-all CoPP-CRITICAL
match access-group 120
class-map match-all CoPP-IMPORTANT
match access-group 121
class-map match-all CoPP-NORMAL
match access-group 122
class-map match-all CoPP-ARP
match protocol ARP
class-map match-all CoPP-UNWANTED
match access-group 123
class-map match-all CoPP-Match-all
match access-group 124
```

Although rate limiting punted traffic is recommended, ensure that the required rates of traffic are well understood, particularly for critical traffic. A very low rate might discard or drop necessary traffic, whereas a very high rate might inundate the Route Processor with non-critical packets to process. These rates are site-specific and vary depending on the local topology and routing table size.

The policed rate depends on both determined criticality and site-specific rate values. For instance, the "normal" SNMP rates differ based on environment. Using the classification scheme mentioned above, critical traffic is permitted without limitation, while important, normal, and default traffic are permitted with appropriate rate limiting. However, this deployment causes the network to drop undesirable traffic immediately.

Table 39-3 extends this example and summarizes a sample policy. Note that the rates defined in the table are used for illustrative purposes; every environment contains different baselines. For example, a large Service Provider topology would require a higher rate of critical traffic (due to large BGP routing tables) than would a typical enterprise network.

The purpose of defining the critical traffic class is not limit rates, but tag this traffic as critical and provide it with unconditional access to the Route Processor. As the policy becomes increasingly refined, a more representative rate should be used for critical traffic and **show** commands can detect abnormal increases in traffic rates.

Table 39-3 Sample CoPP Policy

Traffic class	Rate (bps)	Conform action	Exceed action
Critical	N/A	Transmit	Transmit
Important	125,000	Transmit	Drop
Normal	64,000	Transmit	Drop
Undesirable	32,000	Drop	Drop
ARP	64,000	Transmit	Drop
MATCH-ALL	96000	Transmit	Drop
Class-default	-	-	-

Example Of a CoPP Policy

```
policy-map CoPP
class CoPP-CRITICAL
police 1000000 31250 31250 conform-action transmit exceed-action transmit violate-action
transmit
class CoPP-IMPORTANT
police 128000 4000 4000 conform-action transmit exceed-action drop violate-action drop
class CoPP-NORMAL
police 64000 2000 2000 conform-action transmit exceed-action drop violate-action drop
class CoPP-UNDESIRABLE
police 32000 1500 1500 conform-action transmit exceed-action drop violate-action drop
class CoPP-ARP
police 64000 1500 1500 conform-action transmit exceed-action drop violate-action drop
class CoPP-Match-all
police 96000 3000 3000 conform-action transmit exceed-action drop violate-action drop
class class-default
```

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 router maintains ARP entries in order to forward traffic to end devices or other routers. 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 router learns the legitimate IP address with the spoofed MAC address and begins to forward traffic to that MAC address. With sticky ARP enabled, the router 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 *Cisco 7600 Series Router Cisco IOS System Message Guide* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sr/msggd/index.htm

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 37

Configuring DHCP Snooping

This chapter describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping on Cisco 7600 series routers.

Note The DHCP snooping feature requires PFC3 and Release 12.2(18)SXE and later releases. The PFC2 does not support DHCP snooping.

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 Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

Understanding DHCP Snooping

These sections describe the DHCP snooping feature:

Overview of DHCP Snooping

DHCP snooping is a security feature that acts like a firewall between untrusted hosts and trusted DHCP servers. The DHCP snooping feature performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Rate-limits DHCP traffic from trusted and untrusted sources.
- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Utilizes the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

Other security features, such as dynamic ARP inspection (DAI), also use information stored in the DHCP snooping binding database.

DHCP snooping is enabled on a per-VLAN basis. By default, the feature is inactive on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

The DHCP snooping feature is implemented in software on the MSFC. Therefore, all DHCP messages for enabled VLANs are intercepted in the PFC and directed to the MSFC for processing.

Trusted and Untrusted Sources

The DHCP snooping feature determines whether traffic sources are trusted or untrusted. An untrusted source may initiate traffic attacks or other hostile actions. To prevent such attacks, the DHCP snooping feature filters messages and rate-limits traffic from untrusted sources.

In an enterprise network, devices under your administrative control are trusted sources. These devices include the switches, routers and servers in your network. Any device beyond the firewall or outside your network is an untrusted source. Host ports are generally treated as untrusted sources.

In a service provider environment, any device that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In the Catalyst 6500 series switch, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to devices (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.

Note For DHCP snooping to function properly, all DHCP servers must be connected to the router through trusted interfaces.

DHCP Snooping Binding Database

The DHCP snooping binding database is also referred to as the DHCP snooping binding table.

The DHCP snooping feature dynamically builds and maintains the database using information extracted from intercepted DHCP messages. The database contains an entry for each untrusted host with a leased IP address if the host is associated with a VLAN that has DHCP snooping enabled. The database does not contain entries for hosts connected through trusted interfaces.

The DHCP snooping feature updates the database when the switch receives specific DHCP messages. For example, the feature adds an entry to the database when the switch receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the switch receives a DHCPRELEASE message from the host.

Each entry in the DHCP snooping binding database includes the MAC address of the host, the leased IP address, the lease time, the binding type, and the VLAN number and interface information associated with the host.

Packet Validation

The router validates DHCP packets received on the untrusted interfaces of VLANs with DHCP snooping enabled. The switch forwards the DHCP packet unless any of the following conditions occur (in which case the packet is dropped):

- The router receives a packet (such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet) from a DHCP server outside the network or firewall.
- The router receives a packet on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match. This check is performed only if the DHCP snooping MAC address verification option is turned on.
- The router receives a DHCPRELEASE or DHCPDECLINE message from an untrusted host with an entry in the DHCP snooping binding table, and the interface information in the binding table does not match the interface on which the message was received.
- The router receives a DHCP packet that includes a relay agent IP address that is not 0.0.0.0.

In releases earlier than Release 12.2(18)SXF1, the router drops DHCP packets that include option-82 information that are received on untrusted ports. With Release 12.2(18)SXF1 and later releases, to support trusted edge routers that are connected to untrusted aggregation-router ports, you can enable the DHCP option-82 on untrusted port feature, which enables untrusted aggregation-router ports to accept DHCP packets that include option-82 information. Configure the port on the edge router that connects to the aggregation switch as a trusted port.

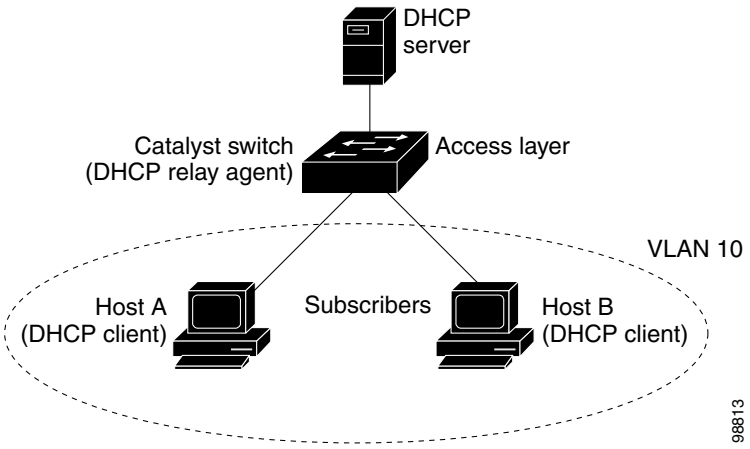
Note With the DHCP option-82 on untrusted port feature enabled, use dynamic ARP inspection on the aggregation router to protect untrusted input interfaces.

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 router, a subscriber device is identified by the router 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 router and are uniquely identified.

Figure 37-1 is an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the router 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 37-1 DHCP Relay Agent in a Metropolitan Ethernet Network



When you enable the DHCP snooping information option-82 on the router, this sequence of events occurs:

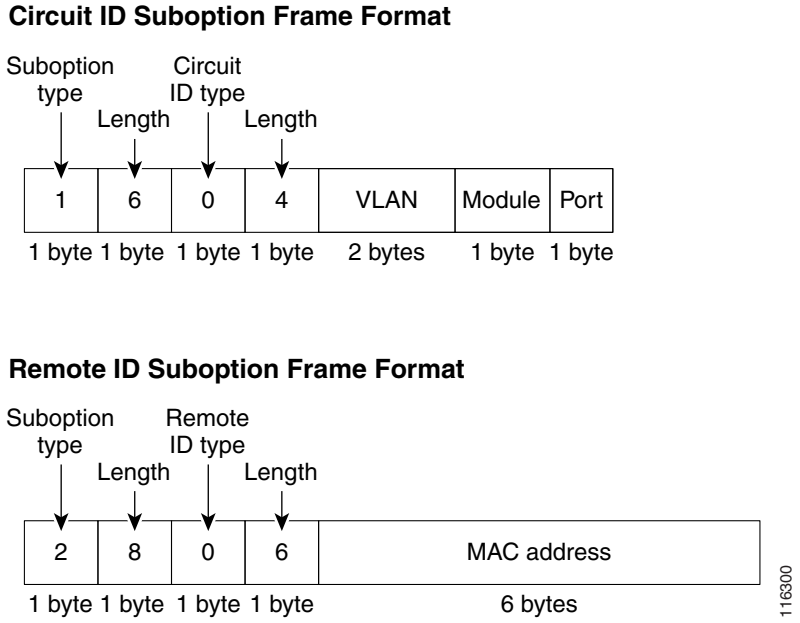
- The host (DHCP client) generates a DHCP request and broadcasts it on the network.
- When the router receives the DHCP request, it adds the option-82 information in the packet. The option-82 information contains the router MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption).
- If the IP address of the relay agent is configured, the router adds the IP address in the DHCP packet.
- The router 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 router if the request was relayed to the server by the router. When the client and server are on the same subnet, the server broadcasts the reply. The router verifies that it originally inserted the option-82 data by inspecting the remote ID and possibly the circuit ID fields. The router removes the option-82 field and forwards the packet to the router 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 37-2 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 37-2 shows the packet formats for the remote ID suboption and the circuit ID suboption. The router 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 37-2 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 router reads the file to build the database for the bindings. The router 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 router 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 router 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 router learns of new bindings or when it loses some bindings, the router 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 37-1 shows all the default configuration values for each DHCP snooping option.

Table 37-1 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 Restrictions and Guidelines

These sections provide DHCP snooping configuration restrictions and guidelines:

DHCP Snooping Configuration Restrictions

When configuring DHCP snooping, note these restrictions:

- The PFC2 does not support DHCP snooping.
- With releases earlier than Release 12.2(18)SXF5, the DHCP snooping database stores a maximum of 512 bindings. If the database attempts to add more than 512 DHCP bindings, all bindings are removed from the database.
- With Release 12.2(18)SXF5 and later releases, the DHCP snooping database stores at least 8,000 bindings.
- With Release 12.2(18)SRA and later releases, the DHCP snooping database stores at least 64,000 bindings.

DHCP Snooping Configuration Guidelines

When configuring DHCP snooping, follow these guidelines:

- DHCP snooping is not active until you enable the feature on at least one VLAN as well as globally on the router. Ensure that service DHCP is enabled (service DHCP is enabled by default).
- Before globally enabling DHCP snooping on the router, 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/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcprt1/1cfdhcp.htm
- 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.

Minimum DHCP Snooping Configuration

The minimum configuration steps for the DHCP snooping feature are as follows:

1. Define and configure the DHCP server.

   For DHCP server configuration information, refer to "Configuring DHCP" in the *Cisco IOS IP and IP Routing Configuration Guide* at:

   http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcprt1/1cfdhcp.htm

2. Enable DHCP snooping on at least one VLAN.

   By default, DHCP snooping is inactive on all VLANs. Refer to the "Enabling DHCP Snooping on VLANs" section on page 37-11

3. Ensure that DHCP server is connected through a trusted interface.

   By default, the trust state of all interfaces is untrusted. Refer to the "Configuring the DHCP Trust State on Layer 2 LAN Interfaces" section on page 37-13

4. Configure the DHCP snooping database agent.

   This step ensures that database entries are restored after a restart or switchover. Refer to the "Configuring the DHCP Snooping Database Agent" section on page 37-14

5. Enable DHCP snooping globally.

   The feature is not active until you complete this step. Refer to the "Enabling DHCP Snooping Globally" section on page 37-9

If you are configuring the switch for DHCP relay, the following additional steps are required:

1. Define and configure the DHCP relay agent IP address.

   If the DHCP server is in a different subnet from the DHCP clients, configure the server IP address in the helper address field of the client side VLAN.

2. Configure DHCP option-82 on untrusted port.

   Refer to the "Enabling the DHCP Option-82 on Untrusted Port Feature" section on page 37-10

Configuring DHCP Snooping

These sections describe how to configure DHCP snooping:

Enabling DHCP Snooping Globally

Note Configure this command as the last configuration step (or enable the DHCP feature during a scheduled maintenance period) because after you enable DHCP snooping globally, the router 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)#
```

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 router 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 router to which any untrusted devices are connected.

With Release 12.2(18)SXF1 and later releases, 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 and the client hardware address match in DHCP packets that are received on untrusted ports. The source MAC address is a Layer 2 field associated with the packet, and the client hardware address is a Layer 3 field in the DHCP packet.

To enable DHCP snooping MAC address verification, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip dhcp snooping verify mac-address`	Enables DHCP snooping MAC address verification.
	`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

By default, the DHCP snooping feature is inactive on all VLANs. You may enable the feature on a single VLAN or a range of VLANs.

When enabled on a VLAN, the DHCP snooping feature creates four entries in the VACL table in the MFC3. These entries cause the PFC3 to intercept all DHCP messages on this VLAN and send them to the MSFC. The DHCP snooping feature is implemented in MSFC software.

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`	Configures the interface as trusted.
	`Router(config-if)# no ip dhcp snooping trust`	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 from the snooping database.

When configuring the DHCP snooping database agent, note the following information:

- With releases earlier than Release 12.2(18)SXF5, the DHCP snooping database stores a maximum of 512 bindings. If the database attempts to add more than 512 DHCP bindings, all bindings are removed from the database.
- With Release 12.2(18)SXF5 and later releases, the DHCP snooping database stores at least 8,000 bindings.
- Store the file on a TFTP server to avoid consuming storage space on the router 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 router 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 37-15
- Example 2: Reading Binding Entries from a TFTP File, page 37-17
- Example 3: Adding Information to the DHCP Snooping Database, page 37-18

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 that failed on bootup.

Note Create a temporary file on the TFTP server with the **touch** command in the TFTP server daemon directory. With some UNIX implementations, the file should have full read and write access permissions (777).

DHCP snooping bindings are keyed on the MAC address and VLAN combination. If an entry in the remote file has an entry for a given MAC address and VLAN set for which the router 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 router is unable to interpret the meaning of the entries from the file.

The router 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 router bootup. These two sets of counters are cleared by the **clear** command. The total counter set may indicate the number of bindings that have been ignored since the last clear.

Example 2: Reading Binding Entries from a TFTP File

To manually read the entries from a TFTP file, perform this task:

	Command	Purpose
Step 1	`Router#` **`show ip dhcp snooping database`**	Displays the DHCP snooping database agent statistics.
Step 2	`Router#` **`renew ip dhcp snoop data`** *`url`*	Directs the router 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 router contains binding entries that correspond to untrusted ports. The table does not contain information about hosts interconnected with a trusted port because each interconnected router will have its own DHCP snooping binding table.

This example shows how to display the DHCP snooping binding information for a router:

```
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 37-2 describes the fields in the **show ip dhcp snooping binding** command output.

Table 37-2 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 38

Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) on the Cisco 7600 series router.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding DAI

These sections describe how DAI helps prevent ARP spoofing attacks:

Understanding ARP

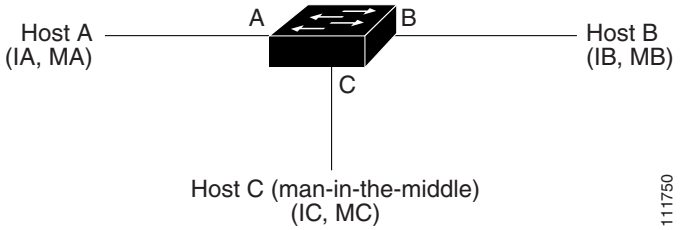
ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, Host B wants to send information to Host A but does not have the MAC address of Host A in its ARP cache. Host B generates a broadcast message for all hosts within the broadcast domain 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 38-1 shows an example of ARP cache poisoning.

Figure 38-1 ARP Cache Poisoning



Hosts A, B, and C are connected to the router 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 router 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 router 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 router, 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 router 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 router. If the ARP packet is received on a trusted interface, the router forwards the packet without any checks. On untrusted interfaces, the router 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 38-8). The router logs dropped packets (see the "Logging of Dropped Packets" section on page 38-4).

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 38-11).

Interface Trust States and Network Security

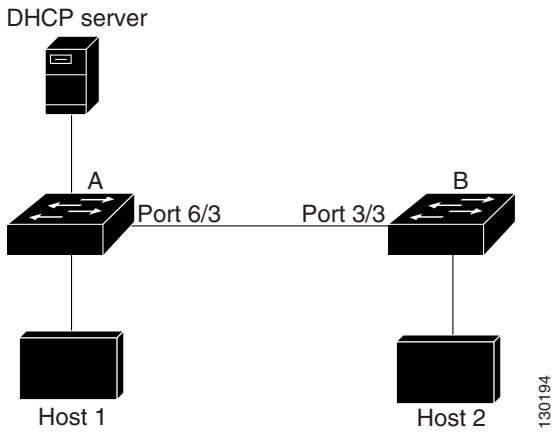
DAI associates a trust state with each interface on the router. Packets arriving on trusted interfaces bypass all DAI validation checks, and those arriving on untrusted interfaces undergo the DAI validation process.

In a typical network configuration, you configure all router ports connected to host ports as untrusted and configure all router ports connected to routers as trusted. With this configuration, all ARP packets entering the network from a given router 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 38-2, assume that both Router A and Router 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 Router A, only Router A binds the IP-to-MAC address of Host 1. Therefore, if the interface between Router A and Router B is untrusted, the ARP packets from Host 1 are dropped by Router B. Connectivity between Host 1 and Host 2 is lost.

Figure 38-2 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 Router A is not running DAI, Host 1 can easily poison the ARP cache of Router B (and Host 2, if the link between the routers is configured as trusted). This condition can occur even though Router B is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a router 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 router running DAI.

In cases in which some routers in a VLAN run DAI and other routers do not, configure the interfaces connecting such routers as untrusted. However, to validate the bindings of packets from routers where DAI is not configured, configure ARP ACLs on the router running DAI. When you cannot determine such bindings, isolate routers running DAI at Layer 3 from routers not running DAI. For configuration information, see the "Sample Two: One Switch Supports DAI" section on page 38-21.

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 routers in the VLAN.

Rate Limiting of ARP Packets

The router 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 router 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 38-9.

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 router uses ACLs only if you configure them by using the **ip arp inspection filter** global configuration command. The router first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the router also denies the packet even if a valid binding exists in the database populated by DHCP snooping.

Logging of Dropped Packets

When the router 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 router 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 38-13.

Default DAI Configuration

Table 38-1 shows the default DAI configuration.

Table 38-1 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 routers 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 37, "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 router 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.
- Cisco IOS Release 12.2(33)SRD2 provides support for ARP Scale to 512k (static or dynamic)—This feature is supported with 3CXL versions of the Cisco 7600 Series ES+ line cards with RSP720-3CXL-GE with 2G SP memory or RSP720-3CXL-10GE with 2G SP memory. Use the following guidelines:
  - If you are using an Switched Virtual Interface (SVI) as a Layer 3 interface, you need to disable MAC Learning.
  - Use the **mls cef maximum-routes** command to increase Cisco Express Forwarding (CEF) holding capacity for IPv4.

Configuring DAI

These sections describe how to configure DAI:

- Displaying DAI Information, page 38-15

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 router does not check ARP packets that it receives from the other router on the trusted interface. It simply forwards the packets.

On untrusted interfaces, the router 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 router 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 38-13.

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 router, and enter interface configuration mode.
Step 3	`Router(config-if)# ip arp inspection trust`	Configures the connection between routers as trusted (default: untrusted).
	`Router(config)# no ip arp inspection trust`	Configures the connection between routers as untrusted.
Step 4	`Router(config-if)# do show ip arp inspection interfaces`	Verifies 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 *Cisco 7600 Series Router Cisco IOS Command Reference*, for information about the **arp access-list** command.

To apply an ARP ACL, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router# ip arp inspection filter` *arp_acl_name* **vlan** {*vlan_ID* | *vlan_range*} [**static**]	Applies the ARP ACL to a VLAN.
Step 3	`Router(config)# do show ip arp inspection vlan` {*vlan_ID* | *vlan_range*}	Verifies your entries.

When applying ARP ACLs, note the following information:

- For *vlan_range*, you can specify a single VLAN or a range of VLANs:
  - To specify a single VLAN, enter a single VLAN number.
  - To specify a range of VLANs, enter a dash-separated pair of VLAN numbers.
  - You can enter a comma-separated list of VLAN numbers and dash-separated pairs of VLAN numbers.
- (Optional) Specify **static** to treat implicit denies in the ARP ACL as explicit denies and to drop packets that do not match any previous clauses in the ACL. DHCP bindings are not used.

  If you do not specify this keyword, it means that there is no explicit deny in the ACL that denies the packet, and DHCP bindings determine whether a packet is permitted or denied if the packet does not match any clauses in the ACL.
- ARP packets containing only IP-to-MAC address bindings are compared against the ACL. Packets are permitted only if the access list permits them.

This example shows how to apply an ARP ACL named example_arp_acl to VLANs 10 through 12 and VLAN 15:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip arp inspection filter example_arp_acl vlan 10-12,15
Router(config)# do show ip arp inspection vlan 10-12,15 | begin Vlan
 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
   10     Enabled          Inactive    example_arp_acl   No
   11     Enabled          Inactive    example_arp_acl   No
   12     Enabled          Inactive    example_arp_acl   No
   15     Enabled          Inactive    example_arp_acl   No

 Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
   10     Deny             Deny
   11     Deny             Deny
   12     Deny             Deny
   15     Deny             Deny
```

Configuring ARP Packet Rate Limiting

When DAI is enabled, the router performs ARP packet validation checks, which makes the router 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 router 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 38-5.

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`	Enter 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.

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** and **dst mac** validations, and a second **ip arp inspection validate** command enables IP validation only, the **src** and **dst mac** validations are disabled as a result of the second command.

- These are the additional validations:
  - **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.

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 38-2.

Table 38-2 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 38-3.

Table 38-3 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 router increments the number of forwarded packets for each ARP request and response packet on a trusted DAI port. The router 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 router increments the appropriate failure count.

To clear or display DAI logging information, use the privileged EXEC commands in Table 38-4:

Table 38-4 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 38-16
- Sample Two: One Switch Supports DAI, page 38-21

Sample One: Two Switches Support DAI

This procedure shows how to configure DAI when two routers support this feature. Host 1 is connected to Router A, and Host 2 is connected to Router B as shown in Figure 38-2 on page 38-3. Both routers are running DAI on VLAN 1 where the hosts are located. A DHCP server is connected to Router A. Both hosts acquire their IP addresses from the same DHCP server. Router A has the bindings for Host 1 and Host 2, and Router B has the binding for Host 2. Router A Fast Ethernet port 6/3 is connected to the Router 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 37, "Configuring DHCP Snooping."
- This configuration does not work if the DHCP server is moved from Router A to a different location.

- To ensure that this configuration does not compromise security, configure Fast Ethernet port 6/3 on Router A and Fast Ethernet port 3/3 on Router B as trusted.

Configuring Router A

To enable DAI and configure Fast Ethernet port 6/3 on Router A as trusted, follow these steps:

Step 1 Verify the connection between switches Router A and Router B:

```
RouterA# 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
RouterB          Fas 6/3            177         R S I     WS-C6506  Fas 3/3
RouterA#
```

Step 2 Enable DAI on VLAN 1 and verify the configuration:

```
RouterA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)# ip arp inspection vlan 1
RouterA(config)# end
RouterA# 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
RouterA#
```

Step 3 Configure Fast Ethernet port 6/3 as trusted:

```
RouterA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)# interface fastethernet 6/3
RouterA(config-if)# ip arp inspection trust
RouterA(config-if)# end
RouterA# show ip arp inspection interfaces fastethernet 6/3

Interface        Trust State     Rate (pps)
 ---------------  -----------     ----------
 Fa6/3            Trusted              None
RouterA#
```

Step 4 Verify the bindings:

```
RouterA# 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
RouterA#
```

Step 5 Check the statistics before and after DAI processes any packets:

```
RouterA# 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
RouterA#
```

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:

```
RouterA# 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
RouterA#
```

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])
RouterA# show ip arp inspection statistics vlan 1
RouterA#
```

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
RouterA#
```

Configuring Router B

To enable DAI and configure Fast Ethernet port 3/3 on Router B as trusted, follow these steps:

Step 1 Verify the connectivity:

```
RouterA# 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
RouterB          Fas 3/3            120         R S I     WS-C6506  Fas 6/3
RouterB#
```

Step 2 Enable DAI on VLAN 1, and verify the configuration:

```
RouterB# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterB(config)# ip arp inspection vlan 1
RouterB(config)# end
RouterB# 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
RouterB#
```

Step 3 Configure Fast Ethernet port 3/3 as trusted:

```
RouterB# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterB(config)# interface fastethernet 3/3
RouterB(config-if)# ip arp inspection trust
RouterB(config-if)# end
RouterB# 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>
RouterB#
```

Step 4 Verify the list of DHCP snooping bindings:

```
RouterB# 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
RouterB#
```

Step 5 Check the statistics before and after DAI processes any packets:

```
RouterB# 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
RouterB#
```

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:

```
RouterB# 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
RouterB#
```

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])
RouterB#
```

The statistics display as follows:

```
RouterB# 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
RouterB#
```

Sample Two: One Switch Supports DAI

This procedure shows how to configure DAI when Router B shown in Figure 38-2 on page 38-3 does not support DAI or DHCP snooping.

If switch Router B does not support DAI or DHCP snooping, configuring Fast Ethernet port 6/3 on Router A as trusted creates a security hole because both Router A and Host 1 could be attacked by either Router B or Host 2.

To prevent this possibility, you must configure Fast Ethernet port 6/3 on Router 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 Router A, you must separate Router A from Router B at Layer 3 and use a router to route packets between them.

To set up an ARP ACL on switch Router 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:

```
RouterA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)# arp access-list H2
RouterA(config-arp-nacl)# permit ip host 1.1.1.1 mac host 1.1.1
RouterA(config-arp-nacl)# end
RouterA# 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:

```
RouterA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)# ip arp inspection filter H2 vlan 1
RouterA(config)# end
RouterA#

RouterA# 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
RouterA#
```

Step 3 Configure Fast Ethernet port 6/3 as untrusted, and verify the configuration:

```
RouterA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RouterA(config)# interface fastethernet 6/3
RouterA(config-if)# no ip arp inspection trust
RouterA(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 Router A and a “get” is permitted by Router 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 39

Configuring Traffic Storm Control

This chapter describes how to configure the traffic storm control feature on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding Traffic Storm Control

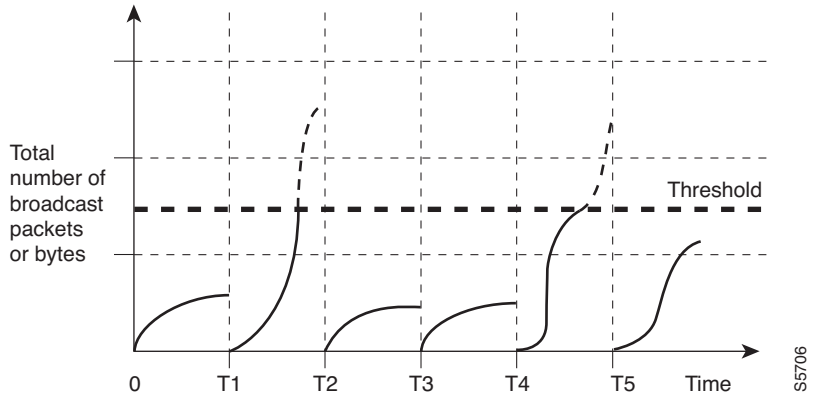
A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. The traffic storm control feature prevents LAN ports from being disrupted by a broadcast, multicast, or unicast traffic storm on physical interfaces.

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 39-1 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 39-1 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 Cisco 7600 series routers 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:

- The following switching modules do not support traffic storm control:
  - WS-X6548-GE-TX
  - WS-X6548V-GE-TX
  - WS-X6148-GE-TX
  - WS-X6148V-GE-TX
  - WS-X6248-RJ-45
  - WS-X6348-RJ-45
  - ,WS-X6148-RJ-45
  - WS-X6148-RJ-21
  - WS-X6248-RJ-21
  - WS-X6196-RJ-21
  - WS-X6148X2-RJ-45
  - WS-X6548-RJ-45
  - WS-X6548-RJ-21
- The router supports broadcast traffic storm control on all LAN ports.
- 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.
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-X6748-GE-TX

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. This configuration applies the traffic storm control level to all traffic storm control modes enabled on Gigabit Ethernet interface 3/16:

```
Router# configure terminal
Router(config)# interface gigabitethernet 3/16
Router(config-if)# storm-control multicast level 70.5
Router(config-if)# end
```

Displaying Traffic Storm Control Settings

To display traffic storm control information, use the commands described in Table 39-1.

Table 39-1 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 the **storm-control** keyword to display the discard count.

CHAPTER 40

Unknown Unicast Flood Blocking

This chapter describes how to configure the unknown unicast flood blocking (UUFB) feature on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

Understanding UUFB

Unknown unicast traffic is flooded to all Layer 2 ports in a VLAN. You can prevent this behavior by using the UUFB feature. The UUFB feature blocks unknown unicast traffic flooding and only permits egress traffic with MAC addresses that are known to exit on the port. The UUFB feature is supported on all ports that are configured with the **switchport** command, including private VLAN (PVLAN) ports.

Configuring UUFB

To configure UUFB, 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 block unicast`	Enables UUFB on the port.
Step 4	`Router(config-if)# do show interfaces [`*type*[1] *slot/port*`] switchport | include unicast`	Verifies the configuration.

1. *type* = **ethernet**, **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 block unicast
Router(config-if)# do show interface fastethernet 5/12 switchport | include unicast
Unknown unicast blocked: enabled
```

CHAPTER 44

Configuring PFC QoS

This chapter describes how to configure quality of service (QoS) as implemented on the Policy Feature Card (PFC) and Distributed Forwarding Cards (DFCs) on the Cisco 7600 series routers.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- For information about QoS and MPLS, see Chapter 43, "Configuring MPLS QoS on the PFC."
- QoS on the Cisco 7600 series routers (PFC QoS) uses some Cisco IOS modular QoS CLI (MQC). Because PFC QoS is implemented in hardware, it supports only a subset of the MQC syntax.
- The PFC does not support Network-Based Application Recognition (NBAR).

Note

Effective with Cisco IOS Software Release 15.0(1)S, a number of QoS commands are hidden in the software image. For more information on replacement MQC commands please see the *Legacy QoS Command Deprecation* feature document:
http://www.cisco.com/en/US/docs/ios/ios_xe/qos/configuration/guide/legacy_qos_cli_deprecation_xe.html
The *Legacy QoS Command Deprecation* document explains that **show queueing** has been replaced by **show policy-map interface**. However, **show policy-map interface** will not show any queueing information on LAN cards because LAN cards do not follow the MQC configuration model. The alternative command: **show mls qos queuing interface** must therefore be used to show queueing information on LAN cards on Cisco 7600 routers.

This chapter contains these sections:

Understanding How PFC QoS Works

The term "PFC QoS" refers to QoS on the Cisco 7600 series router. PFC QoS is implemented on various router components in addition to the PFC and any DFCs. These sections describe how PFC QoS works:

Port Types Supported by PFC QoS

The PFC 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

PFC QoS supports *LAN ports*. LAN ports are Ethernet ports on Ethernet switching modules such as 6xxx cards, except for the 4-port Gigabit Ethernet WAN (GBIC) modules (OSM-4GE-WAN and OSM-2+4GE-WAN+). Some OSMs have four Ethernet LAN ports in addition to WAN ports.

Note OSMs are not supported from 12.2(33) SRE onwards.

PFC QoS supports optical services module (OSM) ports. *OSM ports* are the WAN ports on OSMs.

Overview

Typically, networks operate on a *best-effort* delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

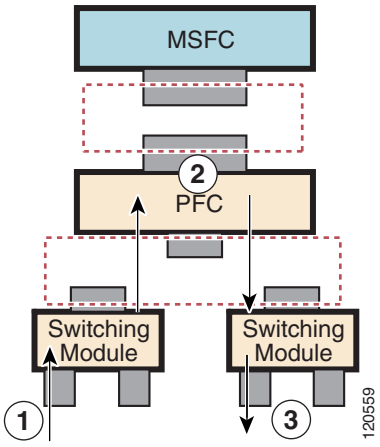
QoS makes network performance more predictable and bandwidth utilization more effective. QoS selects (classifies) network traffic, uses or assigns QoS labels to indicate priority, makes the packets comply with the configured resource usage limits (polices the traffic and marks the traffic), and provides congestion avoidance where resource contention exists.

PFC QoS classification, policing, marking, and congestion avoidance is implemented in hardware on the PFC, DFCs, and in LAN switching module port Application Specific Integrated Circuits (ASICs). ESM 20 ingress QOs happens in the DFCs.

Note Cisco 7600 series routers 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, Cisco 7600 series routers do not support MQC-configured queuing.

Figure 44-1 shows an overview of QoS processing in a Cisco 7600 series router.

Figure 44-1 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.
   - Cos, IP prec and DSCP values are derived from the internal DSCP values using the show mls Qos maps command.
   - 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 a port is configured as untrusted.

     —If a port is configured as trusted, but the traffic is not in an ISL, 802.1Q, or 802.1p frame.

     On OSM ATM and POS ports, PFC QoS always sets CoS equal to zero.
   - 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. Layer 3 DSCP-based queue mapping is available only on WS-X6708-10GE ports.
2. PFC and DFC QoS features:
   - Internal DSCP—On the PFC and DFCs, QoS associates an internal DSCP value with all traffic to classify it for processing through the system. There is an initial internal DSCP based on the traffic trust state and a final internal DSCP. The final internal DSCP can be the same as the initial value or an MQC policy map can set it to a different value.
   - MQC policy maps—MQC policy maps can do one or more of these operations:

     —Change the trust state of the traffic (bases the internal DSCP value on a different QoS label)

     —Set the initial internal DSCP value (only for traffic from untrusted ports)

     —Mark the traffic

     —Police the traffic
3. Egress Ethernet LAN port QoS features:
   - Layer 3 DSCP marking with the final internal DSCP (optionally with PFC)

  - Layer 2 CoS marking mapped from the final internal DSCP
  - Layer 2 CoS-based congestion avoidance. (Layer 3 DSCP-based queue mapping is available only on WS-X6708-10GE ports.)

These figures provide more detail about the relationship between QoS and the router components:

- Figure 44-2, Traffic Flow and PFC QoS Features with the PFC
- Figure 44-3, PFC QoS Features and Component Overview

Figure 44-2 Traffic Flow and PFC QoS Features with the PFC

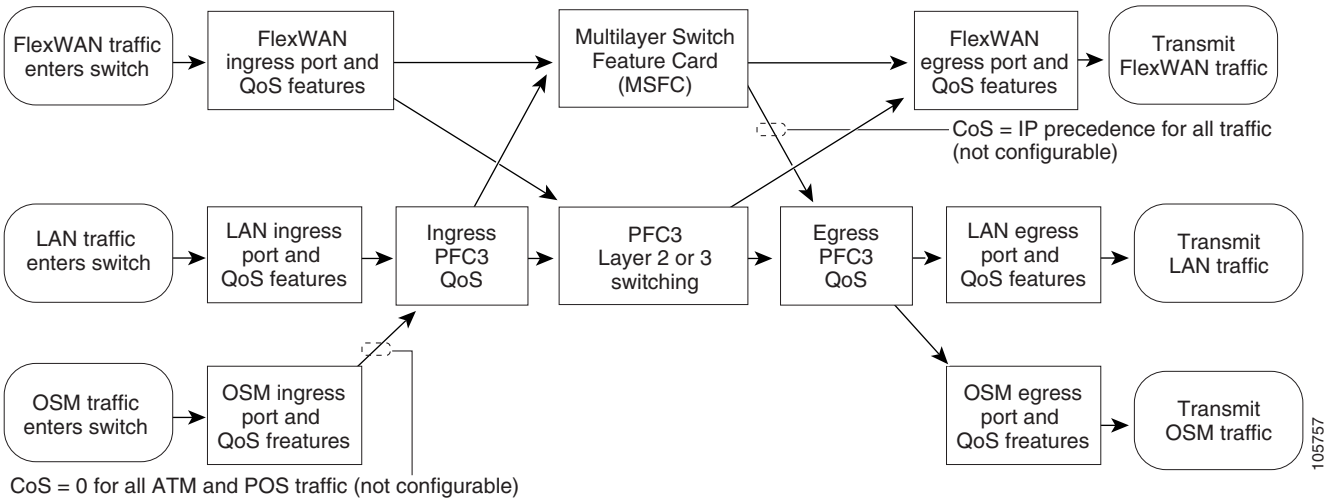
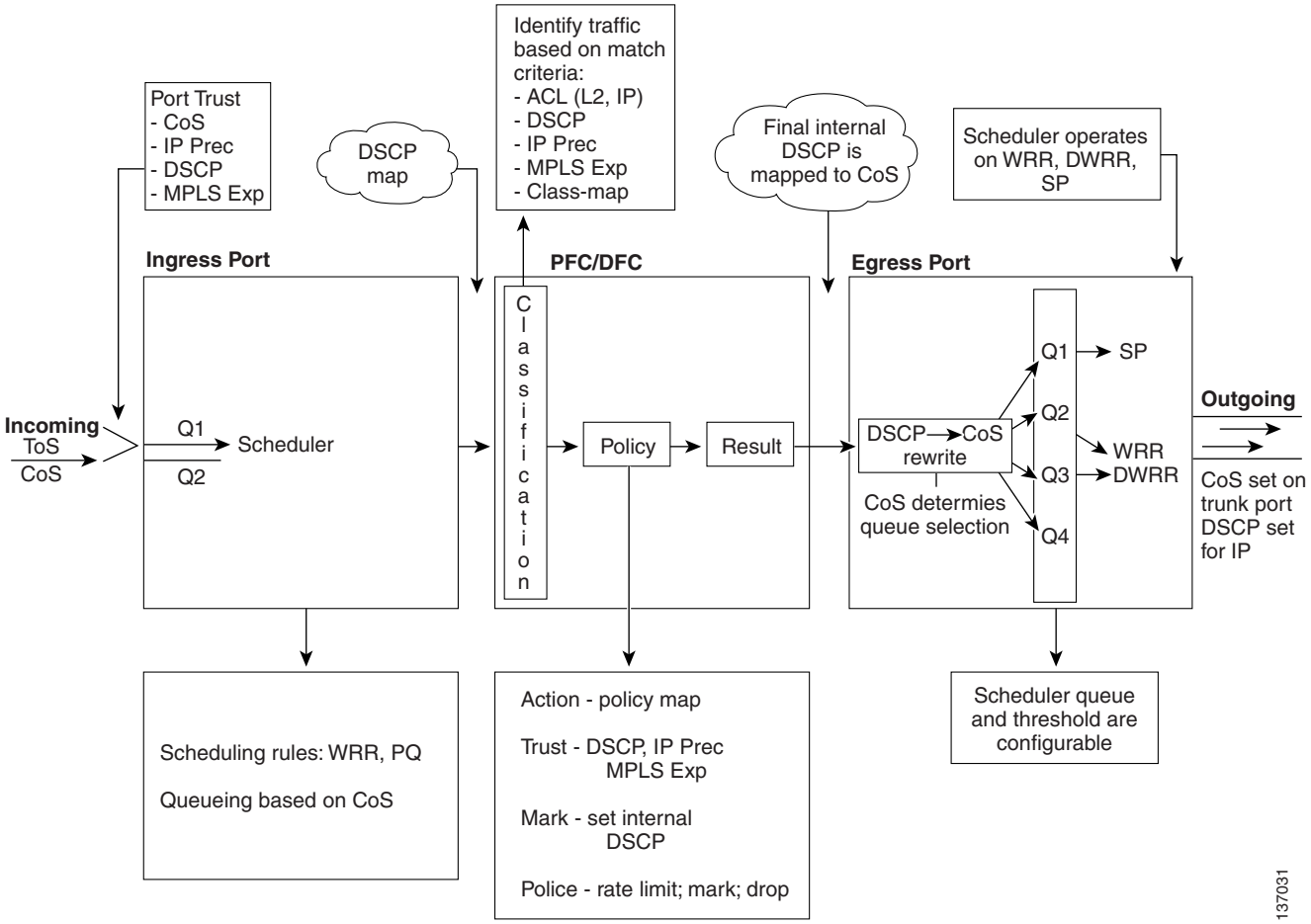


Figure 44-2 shows how traffic flows through the QoS features with a PFC:

- Traffic can enter on any type of port and exit on any type of port.
- DFCs implement PFC QoS locally on switching modules.
- For FlexWAN module traffic:
  - Ingress FlexWAN QoS features can be applied to FlexWAN ingress traffic.
  - Ingress FlexWAN traffic can be Layer 3-switched by the PFC or routed in software by the MSFC.
  - Egress PFC QoS is not applied to FlexWAN ingress traffic.
  - Egress FlexWAN QoS can be applied to FlexWAN egress traffic.
- For LAN-port traffic:
  - Ingress LAN-port QoS features can be applied to LAN-port ingress traffic.
  - Ingress PFC QoS can be applied to LAN-port ingress traffic.
  - Ingress LAN-port traffic can be Layer-2 or Layer-3 switched by the PFC or routed in software by the MSFC.
  - Egress PFC QoS and egress LAN-port QoS can be applied to LAN-port egress traffic.
- For OSM traffic:
  - Ingress OSM-port QoS features can be applied to OSM-port ingress traffic.
  - Ingress PFC QoS can be applied to OSM-port ingress traffic.

- Ingress OSM-port traffic can be Layer-3 switched by the PFC or routed in software by the MSFC.
- Egress PFC QoS and egress OSM-port QoS can be applied to OSM-port egress traffic.

Figure 44-3 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 44-6
- PFC and DFC QoS Features, page 44-7
- Egress Port QoS Features, page 44-10

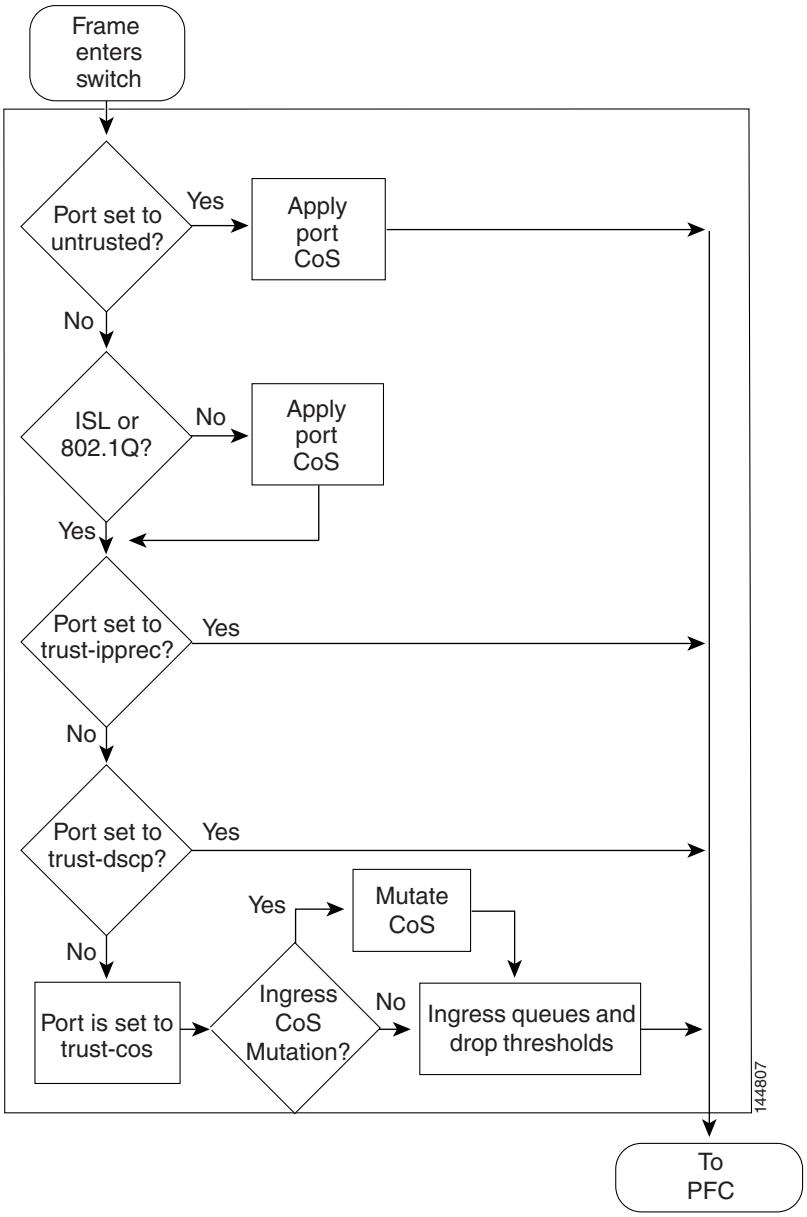
Ingress LAN Port PFC QoS Features

These sections provide an overview of the ingress port QoS features:

Flowchart of Ingress LAN Port PFC QoS Features

Figure 44-4 shows how traffic flows through the ingress LAN port PFC QoS features.

Figure 44-4 Ingress Port QoS Features



Note Ingress CoS mutation is supported only on 802.1Q tunnel ports. DSCP-based queue mapping is supported only on WS-X6708-10GE 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. Received CoS values are preserved only on ports configured to trust CoS.

- 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.

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. In Release 12.2(33)SRC and later releases, you can configure WS-X6708-10GE trust DSCP ports to use received DSCP values for congestion avoidance. See the "Classification and Marking at Trust CoS Ingress LAN Ports" section on page 44-15 for more information about ingress congestion avoidance.

PFC and DFC QoS Features

These sections describe PFCs and DFCs as they relate to QoS:

Supported Policy Feature Cards

The policy feature card (PFC) is a daughter card that resides on the supervisor engine. The PFC provides QoS in addition to other functionality. The following PFCs are supported on Cisco 7600 series routers:

- PFC3B on the Supervisor Engine 720 and Supervisor Engine 32
- PFC3BXL on the Supervisor Engine 720
- The PFC3C and PFC3CXL on the Route Switch Processor 720 (RSP720) and RSP720 10 gigabyte.

Supported Distributed Forwarding Cards

The PFC sends a copy of the QoS policies to the distributed forwarding card (DFC) to provide local support for the QoS policies, which enables the DFCs to support the same QoS features that the PFC supports.

DFCs support supports 6xxx LAN DFC cards such as 6708, 6748.

PFC and DFC QoS Feature List and Flowchart

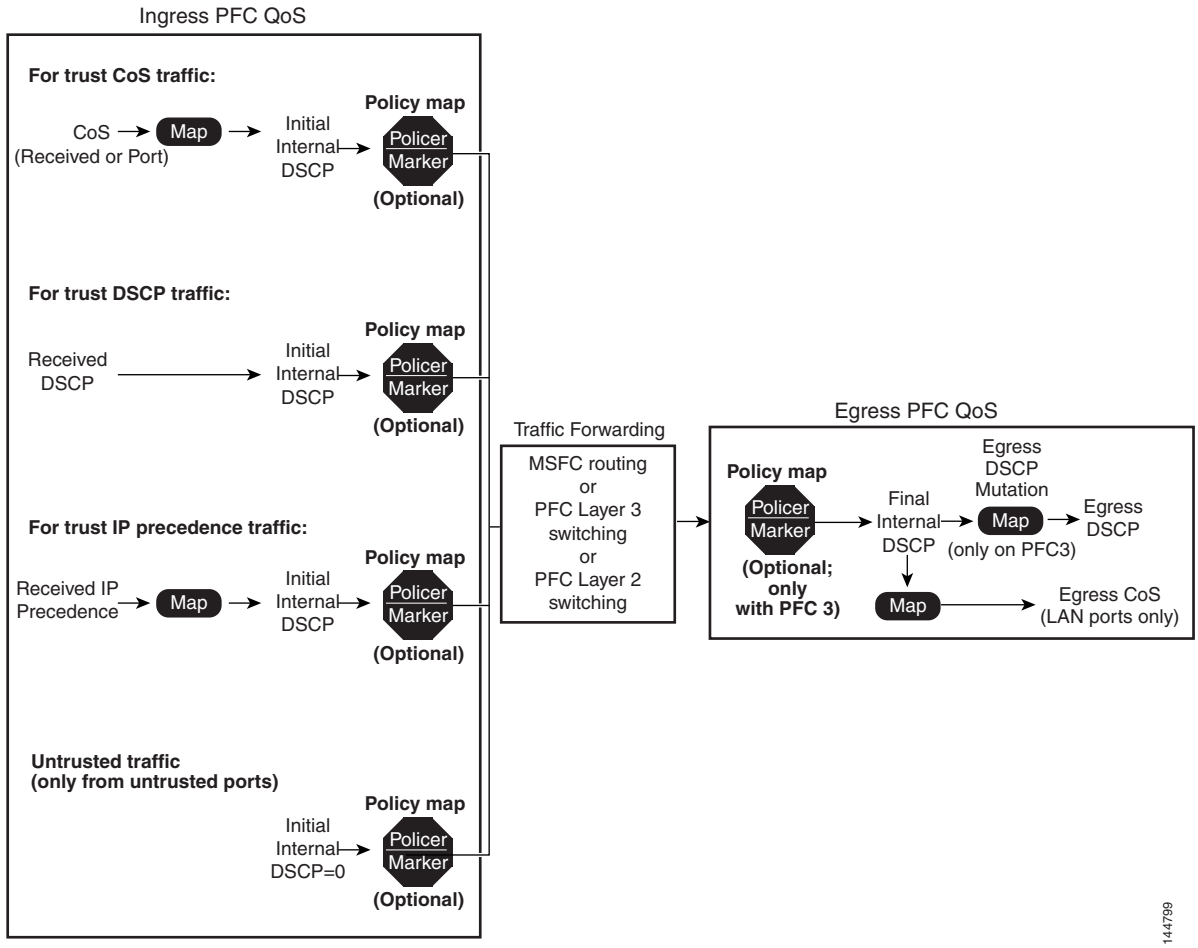
Table 44-1 lists the QoS features supported on the different versions of PFCs and DFCs.

Table 44-1 QoS Features Supported on PFCs and DFCs

Feature	PFC3B/DFC3B	PFC3BXL/DFC3BXL
Support for DFCs	Yes	Yes
Flow granularity	Source Destination	Source Destination
QoS ACLs	IP, MAC	IP, MAC
DSCP transparency **Note** Enabling DSCP transparency disables egress ToS rewrite.	Optional	Optional
Egress ToS rewrite	Optional	Optional
Policing:		
Ingress aggregate policers	Yes	Yes
Egress aggregate policers	Yes	Yes
Number of aggregate policers	1022	1022
Microflow policers	64 rates	64 rates
Number of flows per Microflow policer	110,000	240,000
Unit of measure for policer statistics	Bytes	Bytes
Basis of policer operation	Layer 2 length	Layer 2 length

Figure 44-5 shows how traffic flows through the QoS features on the PFC and DFCs.

Figure 44-5 QoS Features on the PFC and DFCs



Note The DSCP transparency feature makes writing the egress DSCP value into the Layer 3 ToS byte optional.

Internal DSCP Values

During processing, PFC QoS represents the priority of all traffic (including non-IP traffic) with an internal DSCP value. On the PFC, before any marking or policing takes place, PFC QoS derives the initial internal DSCP value as follows:

- From received CoS values or from port CoS values for trust CoS traffic.

  Note Traffic from an untrusted ingress port has the port CoS value. If traffic from an untrusted port matches a trust CoS policer, PFC QoS derives the internal DSCP value from the ingress port CoS value.

- Mapped from received IP precedence values for trust IP precedence traffic.
- From received DSCP values for trust DSCP traffic.

- Mapped from port CoS or DSCP values configured in policy maps for untrusted traffic.

For trust CoS traffic and trust IP precedence traffic, PFC QoS uses configurable maps to derive the internal 6-bit DSCP value from CoS or IP precedence, which are 3-bit values.

Marking and policing (set action) on the PFC can change the initial internal DSCP value to a final internal DSCP value, which is then used for all subsequently applied QoS features.

Port-Based PFC QoS

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:

- Port based PFC Qos 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.

VLAN-Based PFC QoS

On ports configured for port-based VLAN-PFC QoS, you can attach a policy map to the ingress LAN port as follows:

- 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. To enable VLAN based Qos on switchports, use the command **mls qos vlan-based** on the interface.

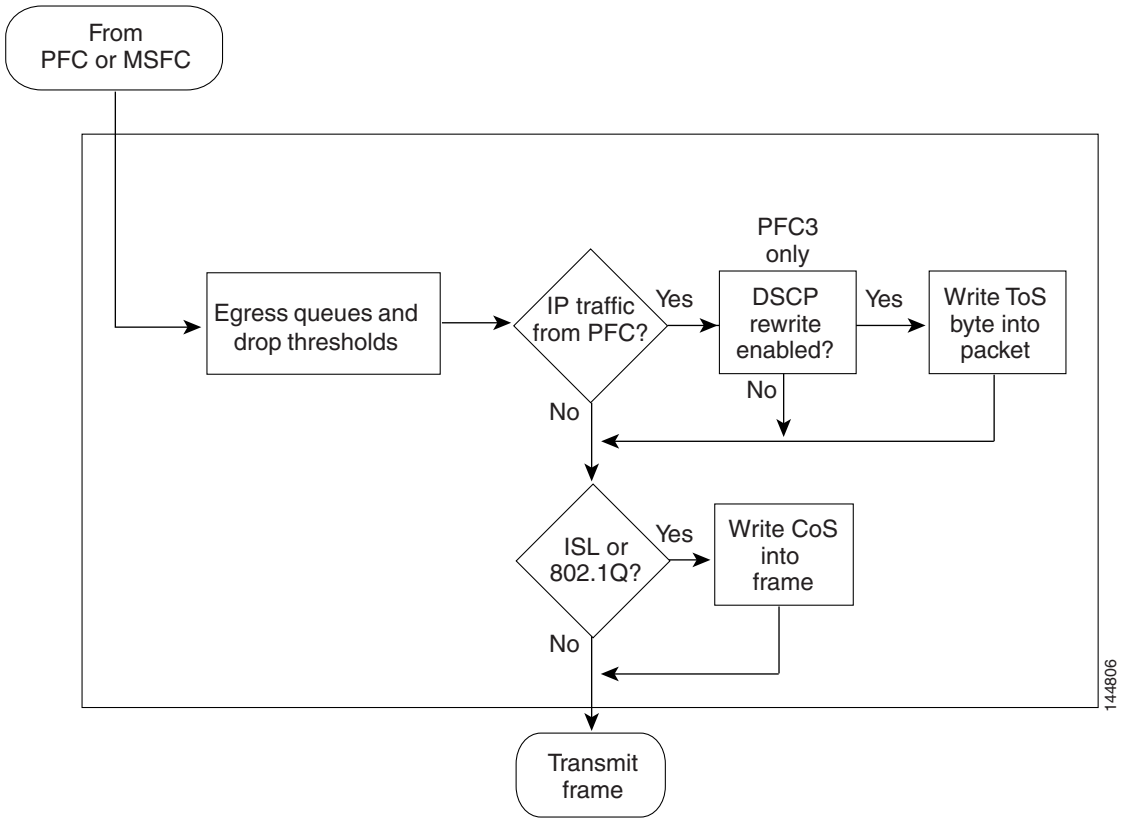
Egress Port QoS Features

These sections describe egress port QoS features:

Flowchart of Egress LAN Port Features

Figure 44-6 shows how traffic flows through the QoS features on egress LAN ports.

Figure 44-6 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 scheduling and to be written into ISL and 802.1Q frames.

Note With Release 12.2(33)SRC and later releases, you can configure WS-X6708-10GE ports to use the final internal DSCP value for egress LAN port classification and congestion avoidance. See Configuring DSCP-Based Queue Mapping, page 44-86.

Egress DSCP Mutation

You can configure 15 egress DSCP mutation maps (per interface assignment is 1 map with a total of 15 tables) 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 44-45).

Egress PFC QoS Interfaces

You can attach an output policy map to a Layer 3 interface (either a LAN port configured as a Layer 3 interface or a VLAN interface) to apply a policy map to egress traffic.

Note

- Output policies do not support microflow policing.
- You cannot apply microflow policing to ARP traffic.
- You cannot set a trust state in an output policy.

Egress ACL Support for Remarked DSCP

Note Egress ACL support for remarked DSCP is also known as packet recirculation.

The PFC supports egress ACL support for remarked DSCP, which enables IP precedence-based or DSCP-based egress QoS filtering to use any IP precedence or DSCP policing or marking changes made by ingress PFC QoS.

Without egress ACL support for remarked DSCP, egress QoS filtering uses received IP precedence or DSCP values; it does not use any IP precedence or DSCP changes made by ingress PFC QoS as the result of policing or marking.

The PFC 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 PFC 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.

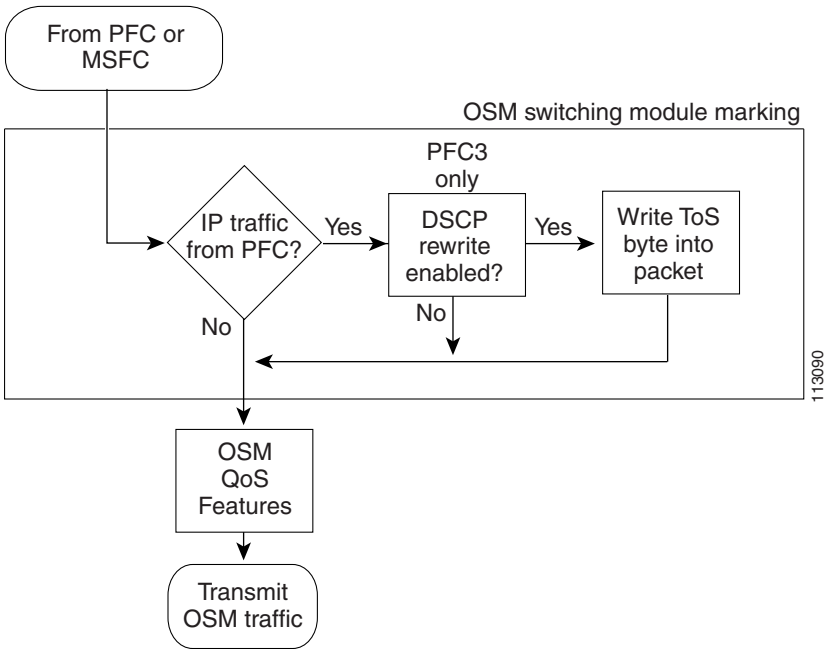
The Layer 3 features configured on an interface where egress ACL support for remarked DSCP is configured might redirect or drop the packets that have been processed by ingress PFC QoS, which would prevent them from being processed by egress PFC QoS.

Marking on Egress OSM Ports

Note Marking on Egress OSM Ports is not supported from 12.2(33) onwards.

Ingress PFC QoS sets DSCP values that can be used by the OSM egress QoS features (see Figure 44-7).

Figure 44-7 Egress WAN Port Marking



Understanding Classification and Marking

The following sections describe where and how classification and marking occur on the Cisco 7600 series routers:

Classification and Marking at Trusted and Untrusted Ingress Ports

The trust state of an ingress port determines how the port marks, schedules, and classifies received Layer 2 frames, and whether or not congestion avoidance is implemented. These are the port trust states:

- Untrusted (default)
- Trust IP precedence
- Trust DSCP
- Trust CoS

Ingress LAN port queuing classification, marking, and congestion avoidance use Layer 2 CoS values only and do not use or set Layer 3 IP precedence or DSCP values.

In Release 12.2(33)SRC and later releases, you can configure WS-X6708-10GE ports to use received DSCP values for ingress LAN port queuing classification and congestion avoidance (See Configuring DSCP-Based Queue Mapping, page 44-86).

The following sections describe classification and marking at trusted and untrusted ingress ports:

Classification and Marking at Untrusted Ingress Ports

PFC QoS Layer 2 remarking marks all frames received through untrusted ingress ports with the port CoS value (the default is zero). PFC Qos classification happens before the port Cos values are reset or remarked.

To map the port CoS value applied to untrusted traffic to the initial internal DSCP value, configure a trust CoS policy map that matches the ingress traffic.

Classification and Marking at Trusted Ingress 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 router, you can apply a different trust state to traffic with a policy map. For example, traffic can enter the router 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:

- Classification and Marking at Trust CoS Ingress LAN Ports, page 44-15
- Classification and Marking at Trust IP precedence Ingress Ports, page 44-15
- Classification and Marking at Trust DSCP Ingress Ports, page 44-15

Classification and Marking at Trust CoS Ingress 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 router 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 router 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.

Note A policy map can change the trust state of the traffic after it enters the router and use received IP precedence or DSCP as the basis of 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 44-21 for more information about ingress queues and thresholds.

Classification and Marking at Trust IP precedence Ingress 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, unless there is a policy map that changes the trust state of the traffic. Because the ingress port queues and thresholds use Layer 2 CoS, PFC QoS does not implement ingress port congestion avoidance on ports configured to trust IP precedence. PFC does not mark any traffic on ingress ports configured to trust IP precedence.

Classification and Marking at Trust DSCP Ingress Ports

You should configure ports to trust DSCP only if they receive traffic that carries valid Layer 3 DSCP. For traffic from trust DSCP ports, PFC QoS uses the received DSCP value as the initial internal DSCP value, unless there is a policy map that changes the trust state of the traffic. 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 DSCP. PFC does not mark any traffic on ingress ports configured to trust received DSCP.

In Release 12.2(33)SRC and later releases, you can enable DSCP-based ingress queues and thresholds on WS-X6708-10GE ports to provide congestion avoidance (See Configuring DSCP-Based Queue Mapping, page 44-86).

Dependence on CoPP

When CoPP is configured, regardless of trust configured on the input port, packets punted to RP arel be treated with trust DSCP action. If CoPP is configured, and you want the punted packets marked or trusted based on input port, then execute the **platform ip features sequential** command on the input port.

Classification and Marking at Ingress OSM Ports

Note OSMs are not supported from 12.2(33) SRE onwards.

PFC QoS associates CoS zero with all traffic received through ingress OSM ports. You can configure ingress OSM port trust states that can be used by the PFC to set IP precedence or DSCP values and the CoS value. You can configure the trust state of each ingress OSM port as follows:

- Untrusted (default)
- Trust IP precedence
- Trust DSCP
- Trust CoS (CoS is always zero for POS and ATM OSM ports because the port CoS value is not configurable on POS and ATM OSM ports.)

Classification and Marking on the PFC Using Service Policies and Policy Maps

PFC QoS supports classification and marking with service policies that attach one policy map to these interface types to apply ingress PFC QoS:

- Each ingress port (except FlexWAN interfaces)
- Each EtherChannel port-channel interface
- Each VLAN interface

You can attach one policy map to each Layer 3 interface (except FlexWAN interfaces) to apply egress PFC QoS.

Each policy map can contain multiple policy-map classes. You can configure a separate policy-map class for each type of traffic handled by the interface. There are two ways to configure filtering in policy-map classes:

- Access control lists (ACLs)
- Class-map **match** commands for IP precedence and DSCP values

Policy-map classes specify actions with the following optional commands:

- Policy-map **set** commands—For untrusted traffic, 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 44-45 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.

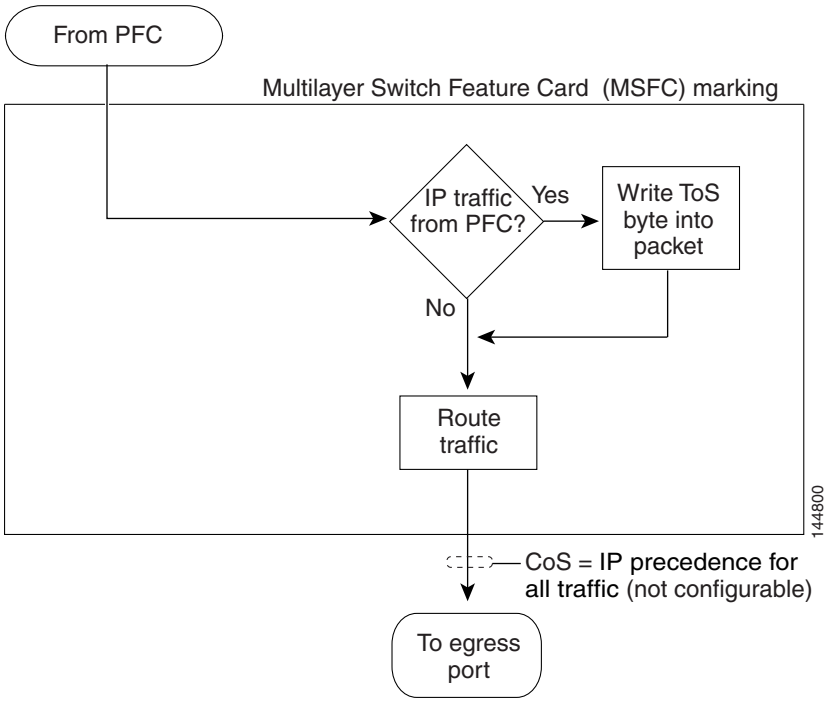
Note A trust CoS policy map cannot restore received CoS in traffic from untrusted ingress LAN ports. Traffic from untrusted ingress LAN ports always has the port CoS value.

- Aggregate and microflow policers—PFC QoS can use policers to either mark or drop both conforming and nonconforming traffic.

Classification and Marking on the MSFC

PFC QoS sends IP traffic to the MSFC with the final internal DSCP values. CoS is equal to IP precedence in all traffic sent from the MSFC to egress ports.

Figure 44-8 Marking with PFC3 and MSFC2A or MSFC3



Note Traffic that is Layer 3 switched on the PFC does not go through the MSFC and retains the CoS value assigned by the PFC.

Policers

These sections describe policers:

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 PFC supports both ingress and egress PFC QoS, which includes ingress and egress policing. Traffic shaping is supported on some WAN modules.

Note Policers can act on ingress traffic per-port or per-VLAN. With a PFC, for egress traffic, the policers can act per-VLAN only.

You can create policers to do the following:

- Mark traffic
- Limit bandwidth utilization and mark traffic

Aggregate Policers

PFC QoS applies the bandwidth limits specified in an aggregate policer cumulatively to all flows in matched traffic. For example, if you configure an aggregate policer to allow 1 Mbps for all TFTP traffic flows on VLAN 1 and VLAN 3, it limits the TFTP traffic for all flows combined on VLAN 1 and VLAN 3 to 1 Mbps.

- You define per-interface aggregate policers in a policy map class with the **police** command. If you attach a per-interface aggregate policer to multiple ingress ports, it polices the matched traffic on each ingress port separately.
- You create named aggregate policers with the **mls qos aggregate-policer** command. If you attach a named aggregate policer to multiple ingress ports, it polices the matched traffic from all the ingress ports to which it is attached.
- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Each PFC or DFC polices independently, which might affect QoS features being applied to traffic that is distributed across the PFC and any DFCs. Examples of these QoS feature are:

- Policers applied to a port channel interface.
- Policers applied to a switched virtual interface.
- Egress policers applied to either a Layer 3 interface or an SVI. Note that PFC QoS performs egress policing decisions at the ingress interface, on the PFC or ingress DFC.

Policers affected by this restriction deliver an aggregate rate that is the sum of all the independent policing rates.

Utilization of Aggregate Policers

PFC3 supports a maximum of 1022 aggregate policers, but some PFC QoS commands other than the **police** command are included in this count. By default, any policy using a **set** or **trust** command are included in the aggregate policer count. Use the **no mls qos marking statistics** command to disable the the **set** or **trust** commands to the aggregate policer count; but you cannot collect statistics for the classmaps associated with these commands. Use the **show platform hardware capacity qos** command to view the aggregate policer count in the QoS Policer resources section of the derived output.

Microflow Policers

PFC QoS applies the bandwidth limit specified in a microflow policer separately to each flow in matched traffic. For example, if you configure a microflow policer to limit the TFTP traffic to 1 Mbps on VLAN 1 and VLAN 3, then 1 Mbps is allowed for each flow in VLAN 1 and 1 Mbps for each flow in VLAN 3. In other words, if there are three flows in VLAN 1 and four flows in VLAN 3, the microflow policer allows each of these flows 1 Mbps.

You can configure PFC QoS to apply the bandwidth limits in a microflow policer as follows:

- You can create microflow policers with up to 63 different rate and burst parameter combinations.
- You create microflow policers in a policy map class with the **police flow** command.
- You can configure a microflow policer to use only source addresses, which applies the microflow policer to all traffic from a source address regardless of the destination addresses.
- You can configure a microflow policer to use only destination addresses, which applies the microflow policer to all traffic to a destination address regardless of the source addresses.
- For MAC-Layer microflow policing, PFC QoS considers MAC-Layer traffic with the same protocol and the same source and destination MAC-Layer addresses to be part of the same flow, including traffic with different EtherTypes. You can configure MAC ACLs to filter IPX traffic.
- For IPX microflow policing, PFC QoS considers IPX traffic with the same source network, destination network, and destination node to be part of the same flow, including traffic with different source nodes or source sockets.
- By default, microflow policers only affect traffic routed by the MSFC. 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 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 Combination of aggregate and microflow policer in each policyclass map is supported only on LAN cards.

Note If traffic is both aggregate and microflow policed, then the aggregate and microflow policers must both be in the same policy-map class and each must use the same **conform-action** and **exceed-action** keyword option: **drop**, **set-dscp-transmit**, **set-prec-transmit**, or **transmit**.

For example, you could create a microflow policer with a bandwidth limit suitable for individuals in a group, and you could create a named aggregate policer with bandwidth limits suitable for the group as a whole. You could include both policers in policy map classes that match the group's traffic. The combination would affect individual flows separately and the group aggregately.

For policy map classes that include both an aggregate and a microflow policer, PFC QoS responds to an out-of-profile status from either policer and, as specified by the policer, applies a new DSCP value or drops the packet. If both policers return an out-of-profile status, then if either policer specifies that the packet is to be dropped, it is dropped; otherwise, PFC QoS applies a marked-down DSCP value.

Note To avoid inconsistent results, ensure that all traffic policed by the same aggregate policer has the same trust state.

Policing uses the Layer 2 frame size. You specify the bandwidth utilization limit as a committed information rate (CIR). You can also specify a higher peak information rate (PIR). Packets that exceed a rate are "out of profile" or "nonconforming."

In each policer, you specify if out-of-profile packets are to be dropped or to have a new DSCP value applied to them (applying a new DSCP value is called "markdown"). Because out-of-profile packets do not retain their original priority, they are not counted as part of the bandwidth consumed by in-profile packets.

If you configure a PIR, the PIR out-of-profile action cannot be less severe than the CIR out-of-profile action. For example, if the CIR out-of-profile action is to mark down the traffic, then the PIR out-of-profile action cannot be to transmit the traffic.

For all policers, PFC QoS uses a configurable global table that maps the internal DSCP value to a marked-down DSCP value. When markdown occurs, PFC QoS gets the marked-down DSCP value from the table. You cannot specify marked-down DSCP values in individual policers.

Note

- Policing with the **conform-action transmit** keywords supersedes the ingress LAN port trust state of matched traffic with trust DSCP or with the trust state defined by a **trust** policy-map class command.
- By default, the markdown table is configured so that no markdown occurs: the marked-down DSCP values are equal to the original DSCP values. To enable markdown, configure the table appropriately for your network.
- When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown.

Understanding Port-Based Queue Types

Port-based queue types are determined by the ASICs that control the ports. The following sections describe the queue types, drop thresholds, and buffers that are supported on the Cisco 7600 series router LAN modules:

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 Cisco 7600 series router LAN modules support the following types of queues:

- Standard queues
- Strict-priority queues

The Cisco 7600 series router LAN modules support the following types of scheduling algorithms between queues:

- 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 allocated to a queue defines the amount of bandwidth allocated to the queue.
- Deficit weighted round robin (DWRR)—In addition to the operation provided by WRR, DWRR keeps track of any low-priority queue under-transmission and compensates in the next round.
- 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 router 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 Cisco 7600 series router 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 of a given CoS value are admitted to the queue based on a random probability designed to avoid buffer congestion. The probability of a frame with a given CoS being admitted to the queue or discarded depends on the weight and threshold assigned to that CoS value.

  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.

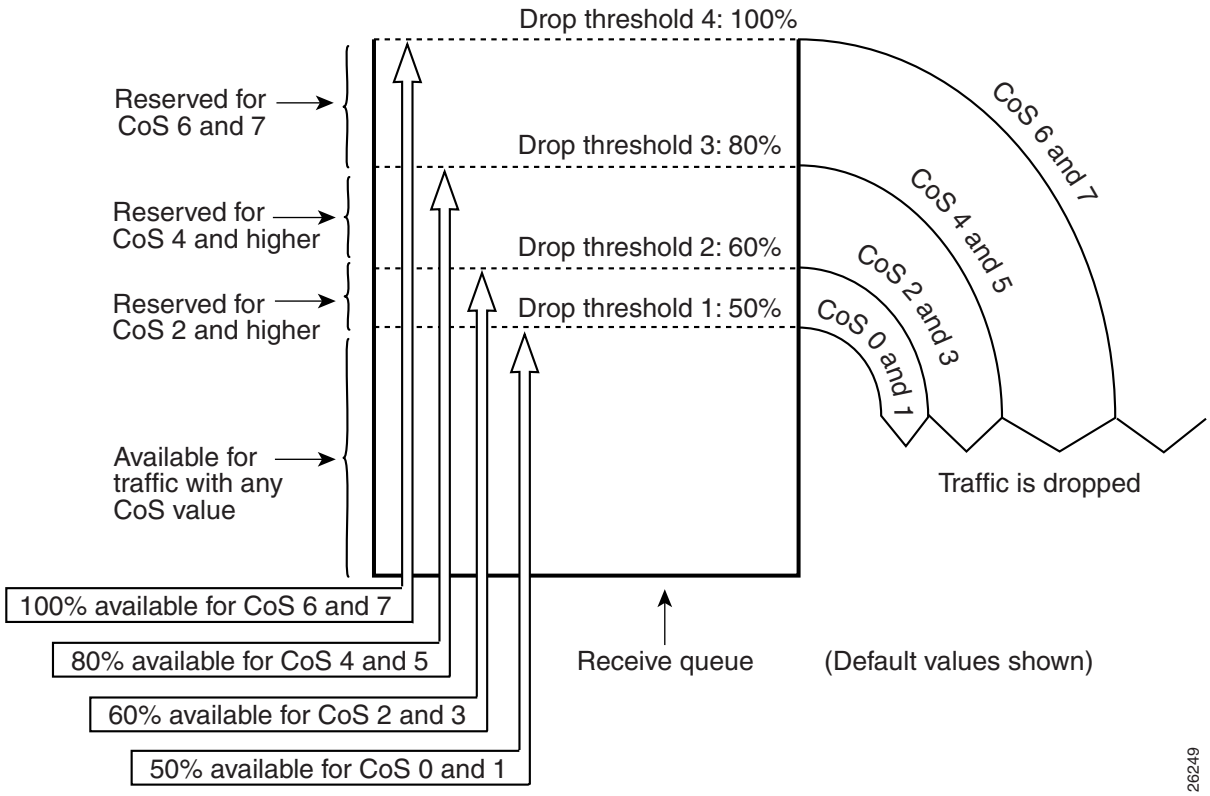
- Tail-drop thresholds—On ports with tail-drop thresholds, frames of a given CoS value are admitted to the queue until the drop threshold associated with that CoS value is exceeded; subsequent frames of that CoS value are discarded until the threshold is no longer exceeded. For example, if CoS 1 is assigned to queue 1, threshold 2, and the threshold 2 watermark is 60 percent, then frames with CoS 1 will not be dropped until queue 1 is 60 percent full. All subsequent CoS 1 frames will be dropped until the queue is less than 60 percent full. With some port types, you can configure the standard receive queue to use both a tail-drop and a WRED-drop threshold by mapping a CoS value to the queue or to the queue and a threshold. The switch uses the tail-drop threshold for traffic carrying CoS values mapped only to the queue. The switch uses WRED-drop thresholds for traffic carrying CoS values mapped to the queue and a threshold. All LAN ports of the same type use the same drop-threshold configuration.

Note In Release 12.2(33)SRC and later releases, you can enable DSCP-based queues and thresholds onWS-X6708-10GE ports. See Configuring DSCP-Based Queue Mapping, page 44-86.

The combination of multiple queues and the scheduling algorithms associated with each queue allows the switch to provide congestion avoidance.

Figure 44-9 illustrates the drop thresholds for a **1q4t** ingress LAN port. Drop thresholds in other configurations function similarly.

Figure 44-9 Receive Queue Drop Thresholds



Ingress Queue Types

To see the queue structure of a LAN port, enter the **show mls qos queuing 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 non configurable (100 percent) tail-drop threshold

Egress Queue Types

To see the queue structure of an egress LAN port, enter the **show mls qos queuing 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:

- Supervisor Engine Module QoS Queue Structures
- Ethernet and Fast Ethernet Module Queue Structures
- Gigabit and 10/100/1000 Ethernet Modules
- 10 Gigabit Ethernet Modules

Table 44-2 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-SUP720	1p1q4t	WRR	1p2q2t	WRR	512 KB	73 KB	439 KB
WS-SUP720-3B							
WS-SUP720-3BXL							
WS-SUP32-10GE	2q8t	WRR	1p3q8t	DWRR	1.3 MB	166 KB	1.2 MB
WS-SUP32-GE							

Table 44-3 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	WRR	1p3q1t	DWRR	1,116 KB	28 KB	1,088 KB
WS-X6548-RJ-21							
WS-X6548-RJ-45							

Table 44-3 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	WRR	2q2t	WRR	128 KB	16 KB	112 KB
WS-X6324-100FX-SM							
WS-X6348-RJ-45							
WS-X6348-RJ-45V							
WS-X6348-RJ-21V							
WS-X6224-100FX-MT					64 KB	8 KB	56 KB
WS-X6248-RJ-45							
WS-X6248-TEL							
WS-X6248A-TEL					128 KB	16 KB	112 KB
WS-X6148-RJ-45							
WS-X6148-RJ-45V							
WS-X6148-45AF							
WS-X6148-RJ-21							
WS-X6148-RJ-21V							
WS-X6148-21AF							
WS-X6148X2-RJ-45	1p1q0t	WRR	1p3q1t	DWRR	1,116 KB	28 KB	1,088 KB
WS-X6148X2-45AF							
WS-X6024-10FL-MT	1q4t	WRR	2q2t	WRR	64 KB	8 KB	56 KB

Table 44-4 Gigabit and 10/100/1000 Ethernet Modules

Modules	Ingress Queue and Drop Thresholds	Ingress Queue Scheduler	Egress Queue and Drop Thresholds	Egress Queue Scheduler	Total Buffer Size	Ingress Buffer Size	Egress Buffer Size
WS-X6816-GBIC	1p1q4t	WRR	1p2q2t	WRR	512 KB	73 KB	439 KB
WS-X6748-GE-TX with DFC3	2q8t	WRR	1p3q8t	DWRR	1.3 MB	166 KB	1.2 MB
WS-X6748-GE-TX with CFC	1q8t	WPR					
WS-X6748-SFP with DFC3	2q8t	WRR					
WS-X6748-SFP with CFC	1q8t	WRR					
WS-X6724-SFP with DFC3	2q8t	WRR					
WS-X6724-SFP with CFC	1q8t	WRR					

Table 44-4 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	WRR	1p2q2t	WRR	1.4 MB	185 KB	1.2 MB
WS-X6548V-GE-TX							
WS-X6548-GE-45AF							
WS-X6516-GBIC	1p1q4t	WRR	1p2q2t	WRR	512 KB	73 KB	439 KB
WS-X6516A-GBIC		WRR		WRR	1 MB	135 KB	946 KB
WS-X6516-GE-TX		WRR		WRR	512 KB	73 KB	439 KB
WS-X6408-GBIC	1q4t	WRR	2q2t	WRR		80 KB	432 KB
WS-X6408A-GBIC	1p1q4t	WRR	1p2q2t	WRR		73 KB	439 KB
WS-X6416-GBIC		WRR					
WS-X6416-GE-MT		WRR					
WS-X6316-GE-TX		WRR					
WS-X6148-GE-TX	1q2t	WRR			1.4 MB	185 KB	1.2 MB
WS-X6148V-GE-TX							
WS-X6148-GE-45AF							

Table 44-5 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-X6708-10GE	8q4t	DWRR	1p7q4t	DWRR SRR	198 MB	108 MB	90 MB
WS-X6704-10GE with DFC3	8q8t	WRR	1p7q8t	DWRR	16 MB	2 MB	14 MB
WS-X6704-10GE with CFC	1q8t	WRR					
WS-X6502-10GE	1p1q8t	WRR	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

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:

Note The ingress LAN port trust state defaults to untrusted with QoS enabled.

Receive-queue Size Percentages

Feature	Default Value
2q8t	Low priority: 80%
	High priority: 20%
8q8t	Lowest priority: 80%
	Intermediate queues: 0%
	Highest priority: 20%

Transmit-Queue Size Percentages

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	10:90
8q8t	10:0:0:0:0:0:0:90
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:

- 1q4t Receive Queues, page 44-30
- 1p1q4t Receive Queues, page 44-31
- 1p1q0t Receive Queues, page 44-31
- 1p1q8t Receive Queues, page 44-32
- 1q8t Receive Queues, page 44-33
- 2q8t Receive Queues, page 44-34
- 8q8t Receive Queues, page 44-35
- 2q2t Transmit Queues, page 44-35
- 1p2q2t Transmit Queues, page 44-36
- 1p3q8t Transmit Queues, page 44-37
- 1p7q8t Transmit Queues, page 44-38
- 1p3q1t Transmit Queues, page 44-39
- 1p2q1t Transmit Queues, page 44-39

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
		Tail-drop	80%
		WRED-drop	Not supported
	Threshold 4	CoS	6 and 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 CoS values mapped to the low-priority queue.

PFC QoS Configuration Guidelines and Restrictions

When configuring PFC QoS, follow these guidelines and restrictions:

General Guidelines

- 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 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**

PFC Guidelines

- All versions of the PFC support QoS for IPv6 unicast and multicast traffic.
- To display information about IPv6 PFC QoS, enter the **show mls qos ipv6** command.
- The QoS features implemented in the port ASICs (queue architecture and dequeuing algorithms) support IPv4 and IPv6 traffic.
- The PFC supports IPv6 named extended ACLs and named standard ACLs.
- The PFC 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.
- With egress ACL support for remarked DSCP configured, the PFC 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 PFC does not apply egress policing to traffic that is being bridged to the MSFC.
- The PFC does not apply egress policing or egress DSCP mutation to multicast traffic from the MSFC.
- PFC QoS does not rewrite the ToS byte in bridged multicast traffic.

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 classmap**
  - **match destination-address**
  - **match input-interface**
  - **match qos-group**

– **match source-address**

Policy Map Command Restrictions

PFC QoS does not support these policy map commands:

- **class** *class_name* **destination-address**
- **class** *class_name* **input-interface**
- **class** *class_name* **protocol**
- **class** *class_name* **qos-group**
- **class** *class_name* **source-address**

Policy Map Class Command Restrictions

PFC QoS does not support these policy map class commands:

- **bandwidth**
- **priority**
- **queue-limit**
- **random-detect**
- **set qos-group**
- **service-policy**

Supported Granularity for CIR and PIR Rate Values

PFC QoS has the following hardware granularity for CIR and PIR rate values:

CIR and PIR Rate Value Range	Granularity
32768 to 2097152 (2 Mbs)	32768 (32 Kb)
2097153 to 4194304 (4 Mbs)	65536 (64 Kb)
4194305 to 8388608 (8 Mbs)	131072 (128 Kb)
8388609 to 16777216 (16 Mbs)	262144 (256 Kb)
16777217 to 33554432 (32 Mbs)	524288 (512 Kb)
33554433 to 67108864 (64 Mbs)	1048576 (1 Mb)
67108865 to 134217728 (128 Mbs)	2097152 (2 Mb)
134217729 to 268435456 (256 Mbs)	4194304 (4 Mb)
268435457 to 536870912 (512 Mbs)	8388608 (8 Mb)
536870913 to 1073741824 (1 Gps)	16777216 (16 Mb)
1073741825 to 2147483648 (2 Gps)	33554432 (32 Mb)
2147483649 to 4294967296 (4 Gps)	67108864 (64 Mb)

Within each range, PFC QoS programs the PFC with rate values that are multiples of the granularity values.

Supported Granularity for CIR and PIR Token Bucket Sizes

PFC QoS has the following hardware granularity for CIR and PIR token bucket (burst) sizes:

CIR and PIR Token Bucket Size Range	Granularity
1 to 32768 (32 KB)	1024 (1 KB)
32769 to 65536 (64 KB)	2048 (2 KB)
65537 to 131072 (128 KB)	4096 (4 KB)
131073 to 262144 (256 KB)	8196 (8 KB)
262145 to 524288 (512 KB)	16392 (16 KB)
524289 to 1048576 (1 MB)	32768 (32 KB)
1048577 to 2097152 (2 MB)	65536 (64 KB)
2097153 to 4194304 (4 MB)	131072 (128 KB)
4194305 to 8388608 (8 MB)	262144 (256 KB)
8388609 to 16777216 (16 MB)	524288 (512 KB)
16777217 to 33554432 (32 MB)	1048576 (1 MB)

Within each range, PFC QoS programs the PFC with token bucket sizes that are multiples of the granularity values.

IP Precedence and DSCP Values

3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP
	8	7	6	5	4	3	
0	0	0	0	0	0	0	0
	0	0	0	0	0	1	1
	0	0	0	0	1	0	2
	0	0	0	0	1	1	3
	0	0	0	1	0	0	4
	0	0	0	1	0	1	5
	0	0	0	1	1	0	6
	0	0	0	1	1	1	7
1	0	0	1	0	0	0	8
	0	0	1	0	0	1	9
	0	0	1	0	1	0	10
	0	0	1	0	1	1	11
	0	0	1	1	0	0	12
	0	0	1	1	0	1	13
	0	0	1	1	1	0	14
	0	0	1	1	1	1	15
2	0	1	0	0	0	0	16
	0	1	0	0	0	1	17
	0	1	0	0	1	0	18
	0	1	0	0	1	1	19
	0	1	0	1	0	0	20
	0	1	0	1	0	1	21
	0	1	0	1	1	0	22
	0	1	0	1	1	1	23
3	0	1	1	0	0	0	24
	0	1	1	0	0	1	25
	0	1	1	0	1	0	26
	0	1	1	0	1	1	27
	0	1	1	1	0	0	28
	0	1	1	1	0	1	29
	0	1	1	1	1	0	30
	0	1	1	1	1	1	31

3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP
	8	7	6	5	4	3	
4	1	0	0	0	0	0	32
	1	0	0	0	0	1	33
	1	0	0	0	1	0	34
	1	0	0	0	1	1	35
	1	0	0	1	0	0	36
	1	0	0	1	0	1	37
	1	0	0	1	1	0	38
	1	0	0	1	1	1	39
5	1	0	1	0	0	0	40
	1	0	1	0	0	1	41
	1	0	1	0	1	0	42
	1	0	1	0	1	1	43
	1	0	1	1	0	0	44
	1	0	1	1	0	1	45
	1	0	1	1	1	0	46
	1	0	1	1	1	1	47
6	1	1	0	0	0	0	48
	1	1	0	0	0	1	49
	1	1	0	0	1	0	50
	1	1	0	0	1	1	51
	1	1	0	1	0	0	52
	1	1	0	1	0	1	53
	1	1	0	1	1	0	54
	1	1	0	1	1	1	55
7	1	1	1	0	0	0	56
	1	1	1	0	0	1	57
	1	1	1	0	1	0	58
	1	1	1	0	1	1	59
	1	1	1	1	0	0	60
	1	1	1	1	0	1	61
	1	1	1	1	1	0	62
	1	1	1	1	1	1	63

1. MSb = most significant bit

Configuring PFC QoS

These sections describe how to configure PFC QoS on the Cisco 7600 series routers:

Note PFC QoS processes both unicast and multicast traffic.

Enabling PFC QoS Globally

To enable PFC QoS globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos`	Enables PFC QoS globally on the router.
	`Router(config)# no mls qos`	Disables PFC QoS globally on the router.
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#
```

Configuring DSCP Transparency

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 slot slot number`	Disables egress ToS-byte rewrite globally on the router.
	`Router(config)# mls qos rewrite ip dscp slot slot number`	Enables egress ToS-byte rewrite globally on the router.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos [ipv6]`	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:

```
Router# configure terminal
Router(config)# no mls qos rewrite ip dscp
Router(config)# end
Router#
```

Configuring Trust State

To enable or disable the trust state over the internal recirculate path, use the **mls qos recirc untrust** command in the global configuration mode.

	Command	Purpose
Step 1	`Router(config)# no mls qos recirc untrust slot slot number`	Disables the trust state over the internal recirculate path.
	`Router(config)# mls qos recirc untrust slot slot number`	Enables the trust state over the internal recirculate path.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos [ipv6]`	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:

```
Router# configure terminal
Router(config)# no mls qos recirc untrust
Router(config)# end
Router#
```

Enabling Queueing-Only Mode

To enable queueing-only mode on the router, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos queueing-only`	Enables queueing-only mode on the router.
	`Router(config)# no mls qos queueing-only`	Disables PFC QoS globally on the router. **Note** You cannot disable queueing-only mode separately.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos [ipv6]`	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 router 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

- PFC QoS supports VLAN-based QoS with DFC3s installed.
- You can attach policy maps to Layer 3 interfaces for application of PFC QoS to egress traffic. VLAN-based or port-based PFC QoS on Layer 2 ports is not relevant to application of PFC QoS to egress traffic on Layer 3 interfaces.

By default, PFC QoS uses policy maps attached to LAN ports. For ports configured as Layer 2 LAN ports with the **switchport** keyword, you can configure PFC QoS to use policy maps attached to a VLAN. Ports not configured with the **switchport** keyword are not associated with a VLAN.

To enable VLAN-based PFC QoS on a Layer 2 LAN port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{`*type*[1] *slot/port*`} | {`**port-channel** *number*`}}`	Selects the interface to configure.
Step 2	`Router(config-if)# mls qos vlan-based`	Enables VLAN-based PFC QoS on a Layer 2 LAN port or a Layer 2 EtherChannel.
	`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:

- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Each PFC or DFC polices independently, which might affect QoS features being applied to traffic that is distributed across the PFC and any DFCs. Examples of these QoS feature are:
  - Policers applied to a port channel interface.
  - Policers applied to a switched virtual interface.
  - Egress policers applied to either a Layer 3 interface or an SVI. Note that PFC QoS performs egress policing decisions at the ingress interface, on the PFC or ingress DFC.

  Policers affected by this restriction deliver an aggregate rate that is the sum of all the independent policing rates.
- You can apply aggregate policers to IPv6 traffic.
- Policing uses the Layer 2 frame size.
- See the "PFC QoS Configuration Guidelines and Restrictions" section on page 44-40 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 PFC supports IPv6 named extended ACLs and named standard ACLs.
  - The PFC supports ARP ACLs.

    Note —The PFC does not apply IP ACLs to ARP traffic.

    —You cannot apply microflow policing to ARP traffic.

  - The PFC does not support IPX ACLs. With a PFC, 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 32, "Configuring Network Security," for additional information about ACLs on the Cisco 7600 series routers.
- **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 44-55
- Enabling VLAN-Based MAC QoS Filtering, page 44-56
- Configuring MAC ACLs, page 44-57

Note You can use MAC ACLs with VLAN ACLs (VACLs). For more information, see Chapter 34, "Configuring VLAN ACLs."

Configuring Protocol-Independent MAC ACL Filtering

The PFC supports 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 PFC3BXL or PFC3CXL.
- Protocol-independent MAC ACL filtering supports microflow policing when the permitted traffic is routed in software by the MSFC3 or MSFC4 (part of the RSP720).

This example shows how to configure VLAN interface 4018 for protocol-independent MAC ACL filtering and how to verify the configuration:

```
Router(config)# interface vlan 4018
Router(config-if)# mac packet-classify
Router(config-if)# end
Router# show running-config interface vlan 4018 | begin 4018
interface Vlan4018
mtu 9216
ipv6 enable
mac packet-classify
end
```

This example shows how to configure Gigabit Ethernet interface 6/1 for protocol-independent MAC ACL filtering and how to verify the configuration:

```
Router(config)# interface gigabitethernet 6/1
Router(config-if)# mac packet-classify
Router(config-if)# end
Router# show running-config interface gigabitethernet 6/1 | begin 6/1
interface GigabitEthernet6/1
mtu 9216
no ip address
mac packet-classify
mpls l2transport route 4.4.4.4 4094
end
```

This example shows how to configure Gigabit Ethernet interface 3/24, subinterface 4000, for protocol-independent MAC ACL filtering and how to verify the configuration:

```
Router(config)# interface gigabitethernet 3/24.4000
Router(config-if)# mac packet-classify
Router(config-if)# end
Router# show running-config interface gigabitethernet 3/24.4000 | begin 3/24.4000
interface GigabitEthernet3/24.4000
encapsulation dot1Q 4000
mac packet-classify
mpls l2transport route 4.4.4.4 4000
end
```

Enabling VLAN-Based MAC QoS Filtering

You can globally enable or disable VLAN-based QoS filtering in MAC ACLs. VLAN-based QoS filtering in MAC ACLs is disabled by default.

To enable VLAN-based QoS filtering in MAC ACLs, perform this task:

Command	Purpose
`Router(config)# mac packet-classify use vlan`	Enables VLAN-based QoS filtering in MAC ACLs.

To disable VLAN-based QoS filtering in MAC ACLs, perform this task:

Command	Purpose
`Router(config)# no mac packet-classify use vlan`	Disables VLAN-based QoS filtering in MAC ACLs.

Configuring MAC ACLs

You can configure named ACLs that filter IPX, DECnet, AppleTalk, VINES, or XNS traffic based on MAC addresses.

You can configure MAC ACLs that perform 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 **ipx-arpa** and **ipx-non-arpa** keywords are supported.
- 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 PFC does not apply IP ACLs to ARP traffic.
- With a PFC, 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 PFC. 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 44-59
- Class Map Filtering Guidelines and Restrictions, page 44-59
- Configuring Filtering in a Class Map, page 44-60

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.
- The PFC 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.
- PFC QoS supports the **match protocol ip** command for IPv4 traffic.

- PFC QoS does not support the **match cos**, **match any**, **match classmap**, **match destination-address**, **match input-interface**, **match qos-group**, and **match source-address** class map commands.
- Cisco 7600 series routers 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 44-55
- Configuring ARP ACLs for QoS Filtering, page 44-58

Other ACLs are not documented in this publication. See the references under **access-list** in the "PFC QoS Policy Configuration Overview" section on page 44-53.

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.
`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.

Command	Purpose
`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 44-62
- Policy Map Class Configuration Guidelines and Restrictions, page 44-62
- Creating a Policy Map Class and Configuring Filtering, page 44-62
- Configuring Policy Map Class Actions, page 44-62

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 mpls** or **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 untrusted 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 44-63
- Configuring the Policy Map Class Trust State, page 44-63
- Configuring Policy Map Class Policing, page 44-64

Configuring Policy Map Class Marking

PFC QoS supports policy map class marking for untrusted traffic with **set** policy map class commands.

To configure policy map class marking for untrusted traffic, 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 44-64
- Configuring a Per-Interface Policer, page 44-65

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:

- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Each PFC or DFC polices independently, which might affect QoS features being applied to traffic that is distributed across the PFC and any DFCs. Examples of these QoS feature are:
  - Policers applied to a port channel interface.
  - Policers applied to a switched virtual interface.
  - Egress policers applied to either a Layer 3 interface or an SVI. Note that PFC QoS performs egress policing decisions at the ingress interface, on the PFC or ingress DFC.

  Policers affected by this restriction deliver an aggregate rate that is the sum of all the independent policing rates.
- When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown.
- You can apply aggregate and microflow policers to IPv6 traffic.
- Policing uses the Layer 2 frame size.
- See the "PFC QoS Configuration Guidelines and Restrictions" section on page 44-40 for information about rate and burst size granularity.

- You can enter the **flow** keyword to define a microflow policer (you cannot apply microflow policing to ARP traffic). When configuring a microflow policer, note the following information:
  - You can enter the **mask src-only** keywords to base flow identification only on source addresses, which applies the microflow policer to all traffic from each source address. The **mask src-only** keywords are supported for both IP traffic and MAC traffic.
  - You can enter the **mask dest-only** keywords to base flow identification only on destination addresses, which applies the microflow policer to all traffic to each source address. The **mask dest-only** keywords are supported 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`	Attaches a policy map to the interface.
	`Router(config-if)# no service-policy [input | output] policy_map_name`	Removes the policy map from the interface.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show policy-map interface {{vlan vlan_ID} | {type[1] slot/port} | {port-channel number}}`	Verifies the configuration.

1. *type* = **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 on Layer 3 interfaces (either LAN ports configured as Layer 3 interfaces or VLAN interfaces). You can attach both an input and an output policy map to a Layer 3 interface.
- VLAN-based or port-based PFC QoS on Layer 2 ports is not relevant to policies attached to Layer 3 interfaces with the **output** keyword.

- Policies attached with the **output** keyword do not support microflow policing.
- You cannot attach a policy map that configures a trust state with the **service-policy output** command.
- Filtering based on IP precedence or DSCP in policies attached with the **output** keyword uses the received IP precedence or DSCP values. Filtering based on IP precedence or DSCP in policies attached with the **output** keyword is not based on any IP precedence or DSCP changes made by ingress QoS.
- Aggregate policing works independently on each DFC-equipped switching module and independently on the PFC, which supports any non-DFC-equipped switching modules. Aggregate policing does not combine flow statistics from different DFC-equipped switching modules. You can display aggregate policing statistics for each DFC-equipped switching module and for the PFC and any non-DFC-equipped switching modules supported by the PFC.
- Each PFC or DFC polices independently, which might affect QoS features being applied to traffic that is distributed across the PFC and any DFCs. Examples of these QoS feature are:
  - Policers applied to a port channel interface.
  - Policers applied to a switched virtual interface.
  - Egress policers applied to either a Layer 3 interface or an SVI. Note that PFC QoS performs egress policing decisions at the ingress interface, on the PFC or ingress DFC.

  Policers affected by this restriction deliver an aggregate rate that is the sum of all the independent policing rates.
- When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown.

This example shows how to attach the policy map named **pmap1** to Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# service-policy input pmap1
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show policy-map interface fastethernet 5/36
 FastEthernet5/36
  service-policy input: pmap1
    class-map: cmap1 (match-all)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 5
  class cmap1
    police 8000 8000 conform-action transmit exceed-action drop
    class-map: cmap2 (match-any)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 2
        0 packets, 0 bytes
        5 minute rate 0 bps
  class cmap2
    police 8000 10000 conform-action transmit exceed-action drop
Router#
```

Configuring Egress DSCP Mutation on a PFC

These sections describe how to configure egress DSCP mutation on a PFC:

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 output_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`	Attaches an egress DSCP mutation map to the interface.
	`Router(config-if)# no mls qos dscp-mutation mutation_map_name`	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

Ingress CoS mutation on IEEE 802.1Q tunnel ports configured to trust received CoS is supported (see the "Applying Ingress CoS Mutation Maps to IEEE 802.1Q Tunnel Ports" section on page 44-73 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.
- Ingress CoS mutation on WS-X6704-10GE, WS-X6748-SFP, WS-X6724-SFP, and WS-X6748-GE-TX switching modules is supported.
- 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 44-74
- Mapping Received IP Precedence Values to Internal DSCP Values, page 44-75
- Configuring DSCP Markdown Values, page 44-75
- Mapping Internal DSCP Values to Egress CoS Values, page 44-77

Mapping Received CoS Values to Internal DSCP Values

To configure the mapping of received CoS values to the DSCP value that PFC QoS uses internally on the PFC, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos map cos-dscp` *dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8*	Configures the received CoS to internal DSCP map. You must enter 8 DSCP values to which PFC QoS maps CoS values 0 through 7.
	`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 PFC, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos map ip-prec-dscp dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8`	Configures the received IP precedence to internal DSCP map. You must enter 8 internal DSCP values to which PFC QoS maps received IP precedence values 0 through 7.
	`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.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos maps`	Verifies the configuration.

When configuring a DSCP markdown map, note the following information:

- You can enter the **normal-burst** keyword to configure the markdown map used by the **exceed-action policed-dscp-transmit** keywords.
- You can enter the **max-burst** keyword to configure the markdown map used by the **violate-action policed-dscp-transmit** keywords.

Note When you create a policer that does not use the **pir** keyword, and the *maximum_burst_bytes* parameter is equal to the *normal_burst_bytes* parameter (which occurs if you do not enter the *maximum_burst_bytes* parameter), the **exceed-action policed-dscp-transmit** keywords cause PFC QoS to mark traffic down as defined by the **policed-dscp max-burst** markdown map.

- To avoid out-of-sequence packets, configure the markdown maps so that conforming and nonconforming traffic uses the same queue.
- You can enter up to 8 DSCP values that map to a marked-down DSCP value.
- You can enter multiple commands to map additional DSCP values to a marked-down DSCP value.
- You can enter a separate command for each marked-down DSCP value.

Note Configure marked-down DSCP values that map to CoS values consistent with the markdown penalty.

This example shows how to map DSCP 1 to marked-down DSCP value 0:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# mls qos map policed-dscp normal-burst 1 to 0
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos map
   Normal Burst Policed-dscp map:                                    (dscp= d1d2)
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :    00 01 02 03 04 05 06 07 08 09
      1 :    10 11 12 13 14 15 16 17 18 19
      2 :    20 21 22 23 24 25 26 27 28 29
      3 :    30 31 32 33 34 35 36 37 38 39
      4 :    40 41 42 43 44 45 46 47 48 49
      5 :    50 51 52 53 54 55 56 57 58 59
      6 :    60 61 62 63

   Maximum Burst Policed-dscp map:                                   (dscp= d1d2)
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :    00 01 02 03 04 05 06 07 08 09
      1 :    10 11 12 13 14 15 16 17 18 19
      2 :    20 21 22 23 24 25 26 27 28 29
      3 :    30 31 32 33 34 35 36 37 38 39
      4 :    40 41 42 43 44 45 46 47 48 49
      5 :    50 51 52 53 54 55 56 57 58 59
      6 :    60 61 62 63
<...Output Truncated...>
Router#
```

Note In the Policed-dscp displays, the marked-down DSCP values are shown in the body of the matrix; the first digit of the original DSCP value is in the column labeled d1 and the second digit is in the top row. In the example shown, DSCP 41 maps to DSCP 41.

Mapping Internal DSCP Values to Egress CoS Values

To configure the mapping of the DSCP value that PFC QoS uses internally on the PFC to the CoS value used for egress LAN port scheduling and congestion avoidance, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos map dscp-cos` *dscp1* [*dscp2* [*dscp3* [*dscp4* [*dscp5* [*dscp6* [*dscp7* [*dscp8*]]]]]]] `to` *cos_value*	Configures the internal DSCP to egress CoS map.
	`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 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 and OSM Ingress Ports

By default, all ingress ports are untrusted. You can configure the ingress port trust state on all Ethernet LAN ports and OSM 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 an ingress 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 mls qos [ipv6]`	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 an ingress port, note the following information:

- With no other keywords, the **mls qos trust** command is equivalent to **mls qos trust dscp**.
- With Release 12.2(33)SRC and later releases, you can use the **mls qos trust dscp** command to enable DSCP-based receive-queue drop thresholds on WS-X6708-10GE ports (See Configuring DSCP-Based Queue Mapping, page 44-86). To avoid dropping traffic because of inconsistent DSCP values when DSCP-based queue mapping is enabled, configure ports with the **mls qos trust dscp** command only when the received traffic carries DSCP values that you know to be consistent with network policy.
- The **mls qos trust cos** command enables 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 44-71).
- 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 mls qos queuing 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*	Configures the ingress LAN port CoS value.
	`Router(config-if)# no mls qos cos` *port_cos*	Reverts to the default port CoS value.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos 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 mls qos queuing 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 44-80
- Configuring a WRED-Drop Transmit Queue, page 44-81
- Configuring a WRED-Drop and Tail-Drop Receive Queue, page 44-82
- Configuring a WRED-Drop and Tail-Drop Transmit Queue, page 44-82

- Configuring 1q4t/2q2t Tail-Drop Threshold Percentages, page 44-84

Note

- Enter the **show mls QoS queuing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command to see the queue structure of a port on a LAN card.
- **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 mls qos queuing 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 mls qos queuing 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.

	Command	Purpose
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 mls qos queuing interface {fastethernet | gigabitethernet} 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.
Step 6	`Router(config-if)# end`	Exits configuration mode.
Step 7	`Router# show mls qos queuing interface {fastethernet | gigabitethernet} 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 mls qos queuing interface {fastethernet | gigabitethernet} 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 mls qos queuing interface gigabitethernet 1/1 | begin Transmit queues
  Transmit queues [type = 1p2q2t]:
    Queue Id    Scheduling  Num of thresholds
    -----------------------------------------
       1         WRR low             2
       2         WRR high            2
       3         Priority            1

    queue random-detect-max-thresholds
    ----------------------------------
      1    40[1] 70[2]
      2    40[1] 70[2]
<...Output Truncated...>
Router#
```

Configuring 1q4t/2q2t Tail-Drop Threshold Percentages

On **1q4t/2q2t** ports, the receive- and transmit-queue drop thresholds have this relationship:

- Receive queue 1 (standard) threshold 1 = transmit queue 1 (standard low priority) threshold 1
- Receive queue 1 (standard) threshold 2 = transmit queue 1 (standard low priority) threshold 2
- Receive queue 1 (standard) threshold 3 = transmit queue 2 (standard high priority) threshold 1
- Receive queue 1 (standard) threshold 4 = transmit queue 2 (standard high priority) threshold 2

To configure tail-drop threshold percentages for the standard receive and transmit queues on **1q4t/2q2t** LAN ports, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {ethernet | fastethernet | gigabitethernet} slot/port`	Selects the interface to configure.
Step 2	`Router(config-if)# wrr-queue threshold queue_id thr1% thr2%`	Configures the receive- and transmit-queue tail-drop thresholds.
	`Router(config-if)# no wrr-queue threshold [queue_id]`	Reverts to the default receive- and transmit-queue tail-drop thresholds.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos queuing 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 mls qos queuing 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 From Cisco IOS Software Release 15.0(1)S enter the **show mls qos queuing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command to see the queue structure of a LAN port.

These sections describe how to map QoS labels to queues and drop thresholds:

Queue and Drop Threshold Mapping Guidelines and Restrictions

When mapping QoS labels to queues and thresholds, note the following information:

- When SRR is enabled, you cannot map any CoS values or DSCP values to strict-priority queues.
- Queue number 1 is the lowest-priority standard queue.
- Higher-numbered queues are higher priority standard queues.
- You can map up to 8 CoS values to a threshold.
- You can map up to 64 DSCP values to a threshold.
- Threshold 0 is a nonconfigurable 100-percent tail-drop threshold on these port types:
  - **1p1q0t** (receive)
  - **1p1q8t** (receive)
  - **1p3q1t** (transmit)
  - **1p2q1t** (transmit)
- The standard queue thresholds can be configured as either tail-drop or WRED-drop thresholds on these port types:
  - **1p1q8t** (receive)
  - **1p3q1t** (transmit)
  - **1p3q8t** (transmit)
  - **1p7q1t** (transmit)

Configuring DSCP-Based Queue Mapping

These sections describe how to configure DSCP-based queue mapping:

- Configuring Ingress DSCP-Based Queue Mapping, page 44-86
- Mapping DSCP Values to Standard Transmit-Queue Thresholds, page 44-88
- Mapping DSCP Values to the Transmit Strict-Priority Queue, page 44-90

Note DSCP-based queue mapping is supported on WS-X6708-10GE ports.

Enabling DSCP-Based Queue Mapping

To enable DSCP-based queue mapping, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface tengigabitethernet slot/port`	Selects the interface to configure.
Step 2	`Router(config-if)# mls qos queue-mode mode-dscp`	Enables DSCP-based queue mapping.
	`Router(config-if)# no mls qos queue-mode mode-dscp`	Reverts to CoS-based queue mapping.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos queuing interface tengigabitethernet slot/port | include Queueing Mode`	Verifies the configuration.

This example shows how to enable DSCP-based queue mapping on 10-Gigabit Ethernet port 6/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 6/1
Router(config-if)# mls qos queue-mode mode-dscp
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show mls qos queuing interface tengigabitethernet 6/1 | include Queueing Mode
    Queueing Mode In Tx direction: mode-dscp
    Queueing Mode In Rx direction: mode-dscp
```

Configuring Ingress DSCP-Based Queue Mapping

Ingress DSCP-to-queue mapping is supported only on ports configured to trust DSCP.

These sections describe how to configure ingress DSCP-based queue mapping:

- Enabling DSCP-Based Queue Mapping, page 44-86
- Mapping DSCP Values to Standard Receive-Queue Thresholds, page 44-87

Configuring the Port to Trust DSCP

To configure the port to trust DSCP perform this task:

	Command	Purpose
Step 1	`Router(config)# interface tengigabitethernet slot/port`	Selects the interface to configure.
Step 2	`Router(config-if)# mls qos trust dscp`	Configures the port to trust received DSCP values.
	`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 mls qos queuing interface tengigabitethernet slot/port | include Trust state`	Verifies the configuration.

This example shows how to configure 10-Gigabit Ethernet port 6/1 port 6/1 to trust received DSCP values:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 6/1
Router(config-if)# mls qos trust dscp
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos queuing interface gigabitethernet 6/1 | include Trust state
  Trust state: trust DSCP
```

Mapping DSCP Values to Standard Receive-Queue Thresholds

To map DSCP values to the standard receive-queue thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface tengigabitethernet slot/port`	Selects the interface to configure.
Step 2	`Router(config-if)# rcv-queue dscp-map queue_# threshold_# dscp1 [dscp2 [dscp3 [dscp4 [dscp5 [dscp6 [dscp7 [dscp8]]]]]]]`	Maps DSCP values to the standard receive queue thresholds.
	`Router(config-if)# no rcv-queue dscp-map`	Reverts to the default mapping.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos queuing interface tengigabitethernet slot/port`	Verifies the configuration.

When mapping DSCP values, note the following information:

- You can enter up to 8 DSCP values that map to a queue and threshold.
- You can enter multiple commands to map additional DSCP values to the queue and threshold.
- You must enter a separate command for each queue and threshold.

This example shows how to map the DSCP values 0 and 1 to threshold 1 in the standard receive queue for 10-Gigabit Ethernet port 6/1 port 6/1:

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

```
Router(config)# interface tengigabitethernet 6/1
Router(config-if)# rcv-queue dscp-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos queuing interface tengigabitethernet 1/1 | begin queue thresh
dscp-map
<...Output Truncated...>
queue thresh dscp-map
----------------------------------------
1     1       0 1 2 3 4 5 6 7 8 9 11 13 15 16 17 19 21 23 25 27 29 31 33 39 41 42 43 44 45
47
1     2
1     3
1     4
2     1       14
2     2       12
2     3       10
2     4
3     1       22
3     2       20
3     3       18
3     4
4     1       24 30
4     2       28
4     3       26
4     4
5     1       32 34 35 36 37 38
5     2
5     3
5     4
6     1       48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
6     2
6     3
6     4
7     1
7     2
7     3
7     4
8     1       40 46
8     2
8     3
8     4
<...Output Truncated...>
Router#
```

Mapping DSCP Values to Standard Transmit-Queue Thresholds

To map DSCP values to standard transmit-queue thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface tengigabitethernet` *slot/port*	Selects the interface to configure.
Step 2	`Router(config-if)# wrr-queue dscp-map` *transmit_queue_# threshold_# dscp1* [*dscp2* [*dscp3* [*dscp4* [*dscp5* [*dscp6* [*dscp7* [*dscp8*]]]]]]]	Maps DSCP values to a standard transmit-queue threshold.
	`Router(config-if)# no wrr-queue dscp-map`	Reverts to the default mapping.

	Command	Purpose
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos queuing interface tengigabitethernet slot/port`	Verifies the configuration.

When mapping DSCP values, note the following information:

- You can enter up to 8 DSCP values that map to a queue and threshold.
- You can enter multiple commands to map additional DSCP values to the queue and threshold.
- You must enter a separate command for each queue and threshold.

This example shows how to map the DSCP values 0 and 1 to standard transmit queue 1/threshold 1 for 10-Gigabit Ethernet port 6/1 port 6/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 6/1
Router(config-if)# wrr-queue dscp-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos queuing interface tengigabitethernet 6/1 | begin queue thresh
dscp-map
queue thresh dscp-map
----------------------------------------
1     1      0 1 2 3 4 5 6 7 8 9 11 13 15 16 17 19 21 23 25 27 29 31 33 39 41 42 43 44 45
47
1     2
1     3
1     4
2     1      14
2     2      12
2     3      10
2     4
3     1      22
3     2      20
3     3      18
3     4
4     1      24 30
4     2      28
4     3      26
4     4
5     1      32 34 35 36 37 38
5     2
5     3
5     4
6     1      48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
6     2
6     3
6     4
7     1
7     2
7     3
7     4
8     1      40 46
<...Output Truncated...>
Router#
```

Mapping DSCP Values to the Transmit Strict-Priority Queue

To map DSCP values to the transmit strict-priority queue, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface tengigabitethernet slot/port`	Selects the interface to configure.
Step 2	`Router(config-if)# priority-queue dscp-map queue_# dscp1 [dscp2 [dscp3 [dscp4 [dscp5 [dscp6 [dscp7 [dscp8]]]]]]]`	Maps DSCP values to the transmit strict-priority queue. You can enter multiple **priority-queue dscp-map** commands to map more than 8 DSCP values to the strict-priority queue.
	`Router(config-if)# no priority-queue dscp-map`	Reverts to the default mapping.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos queuing interface tengigabitethernet slot/port`	Verifies the configuration.

When mapping DSCP values to the strict-priority queue, note the following information:

- The queue number is always 1.
- You can enter up to 8 DSCP values to map to the queue.
- You can enter multiple commands to map additional DSCP values to the queue.

This example shows how to map DSCP value 7 to the strict-priority queue on 10 Gigabit Ethernet port 6/1:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tengigabitethernet 6/1
Router(config-if)# priority-queue dscp-map 1 7
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
 Router# show mls qos queuing interface tengigabitethernet 6/1 | begin queue thresh
dscp-map
    queue thresh dscp-map
    ----------------------------------------
<...Output Truncated...>
    8     1      7 40 46
<...Output Truncated...>
Router#
```

Configuring CoS-Based Queue Mapping

These sections describe how to configure CoS-based queue mapping:

- Mapping CoS Values to Standard Receive-Queue Thresholds, page 44-91
- Mapping CoS Values to Standard Transmit-Queue Thresholds, page 44-91
- Mapping CoS Values to Strict-Priority Queues, page 44-92
- Mapping CoS Values to Tail-Drop Thresholds on 1q4t/2q2t LAN Ports, page 44-93

Mapping CoS Values to Standard Receive-Queue Thresholds

To map CoS values to the standard receive-queue thresholds, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Selects the interface to configure.
Step 2	`Router(config-if)# rcv-queue cos-map` *queue_# threshold_# cos1* [*cos2* [*cos3* [*cos4* [*cos5* [*cos6* [*cos7* [*cos8*]]]]]]]	Maps CoS values to the standard receive queue thresholds.
	`Router(config-if)# no rcv-queue cos-map`	Reverts to the default mapping.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos queuing 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 mls qos queuing 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 mls qos queuing 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 mls qos queuing 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 mls qos queuing 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.

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 mls qos queuing 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
    1     4      6
    2     1      5 7
<...Output Truncated...>
Router#
```

Mapping CoS Values to Tail-Drop Thresholds on 1q4t/2q2t LAN Ports

Note Enter the **show mls qos queuing 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 mls qos queuing 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 mls qos queuing 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 router 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:

- Deficit weighted round robin (DWRR)—Supported on **1p3q1t**, **1p2q1t**, **1p3q8t**, and **1p7q8t** ports. DWRR keeps track of any low-priority queue under-transmission and compensates in the next round.
- Weighted round robin (WRR)—Supported on all other ports. 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.

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 division 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 low_priority_queue_percentage [intermediate_priority_queue_percentages] high_priority_queue_percentage`
	Or:
	`Router(config-if)# wrr-queue bandwidth low_priority_queue_weight [intermediate_priority_queue_weights] high_priority_queue_weight`
		Allocates bandwidth between standard transmit queues:
		• 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`	Reverts to the default bandwidth allocation.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos queuing 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 mls qos queuing interface gigabitethernet 1/2 | include bandwidth
WRR bandwidth ratios:    3[queue 1]   1[queue 2]
Router#
```

Setting the Receive-Queue Size Ratio on 1p1q0t and 1p1q8t Ports

To set the size ratio between the strict-priority and standard receive queues on a **1p1q0t** or **1p1q8t** port, 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 standard_queue_weight strict_priority_queue_weight`	Sets the size ratio between the strict-priority and standard 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 mls qos queuing 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 strict priority-to-standard 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 mls qos queuing interface fastethernet 2/2 | include queue-limit
    queue-limit ratios:     75[queue 1]  15[queue 2]
Router#
```

Setting the LAN-Port Transmit-Queue Size Ratio

To set 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*	Sets the transmit-queue size ratio between transmit queues.
	`Router(config-if)# no wrr-queue queue-limit`	Reverts to the default transmit-queue size ratio.
Step 3	`Router(config-if)# end`	Exits configuration mode.
Step 4	`Router# show mls qos queuing interface` *type*[1] *slot/port*	Verifies the configuration.

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

When setting the transmit-queue size ratio between transmit queues, note the following information:

- 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).
- For ports that have an egress strict priority queue:
  - With Release 12.2SR and later releases, you can enter the priority-queue queue-limit interface command to set the size of the egress strict priority queue on these switching modules:
    - WS-X6502-10GE (1p2q1t)
    - WS-X6148A-GE-TX (1p3q8t)
    - WS-X6148-RJ-45 (1p3q8t)

  - WS-X6148-FE-SFP (1p3q8t)
  - WS-X6748-SFP (1p3q8t)
  - WS-X6724-SFP (1p3q8t)
  - WS-X6748-GE-TX (1p3q8t)
  - WS-X6704-10GE (1p7q4t)
  - WS-SUP32-10GE-3B (1p3q8t)
  - WS-SUP32-GE-3B (1p3q8t)
  - WS-X6708-10GE (1p7q4t)
  - With releases earlier than Release 12.2SR and for other modules, PFC QoS sets the egress strict-priority queue size equal to the high-priority queue size.
- 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 mls qos queuing 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 44-98. 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 router 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 routers 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 routers.

The port on the access router 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 an output service policy can only be applied to VLAN interfaces. 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 mls qos queuing 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 router 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 mls qos queuing interface gigabitethernet 5/1 | include Trust
Trust state:trust DSCP
```

Prioritizing Traffic on Interswitch Links

This section describes how the routers 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 44-99 and the "Accepting the Traffic Priority Value on Interswitch Links" section on page 44-101 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 Cisco 7600 series router 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)

Cisco 7600 series router 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 router, 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 44-99) or to trust the packet's incoming DSCP value (as in the "Accepting the Traffic Priority Value on Interswitch Links" section on page 44-101). These options determine the packet's priority as it leaves the router.

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 mls qos queuing 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>
```

Note From Cisco IOS Software Release 15.0(1)S use the **show mls qos queuing interface** command to verify the configuration on LAN ports.

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 mls QoS queuing 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 router 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 44-91.

Now you understand how traffic is assigned to the available queues on the output ports of the router. The next concept to understand is how the queue weights operate, which is called the queue scheduling algorithm.

On the Cisco 7600 series router, 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 44-94).

Note From Cisco IOS Software Release 15.0(1)S use the **show mls qos queuing interface** command to determine where packet loss is happening on LAN ports.

The best way to verify that the router is handling oversubscription is to ensure that there is minimal packet drop. Use the **show mls QoS queuing 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 Cisco 7600 series router Ethernet ports, the supported rate-limiting method is called policing. Policing is implemented in the PFC hardware with no performance impact. A policer operates by allowing the traffic to flow freely as long as the traffic rate remains below the configured transmission rate. Traffic bursts are allowed, provided that they are within the configured burst size. Any traffic that exceeds the configured rate and burst can be either dropped or marked down to a lower priority. The benefit of policing is that it can constrain the amount of bandwidth that a particular application consumes, which helps ensure quality of service on the network, especially during abnormal network conditions such as a virus or worm attack.

This example focuses on a basic per-interface aggregate policer applied to a single interface in the inbound direction, but you can use other policing options to achieve this same result.

The configuration of a policer is similar to the marking example provided in the "Classifying Traffic from PCs and IP Phones in the Access Layer" section on page 44-99 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 Supervisor Engine 720 forwarding engine also 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
```

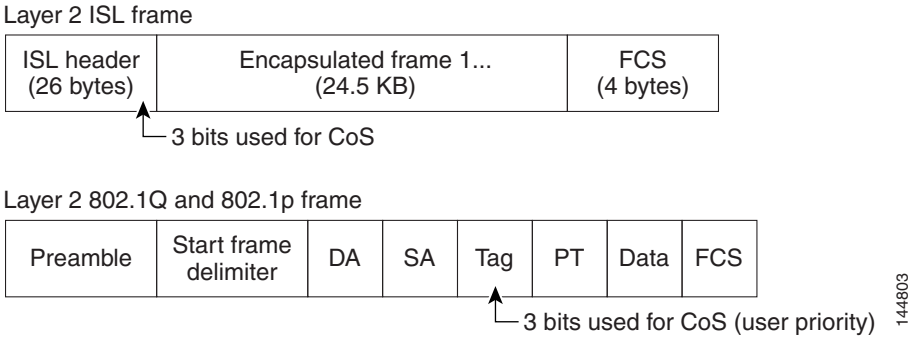
To monitor the policing operation, use these commands:

```
show policy-map interface fastethernet 5/1
show class-map
show mls qos ip fastethernet 5/1
```

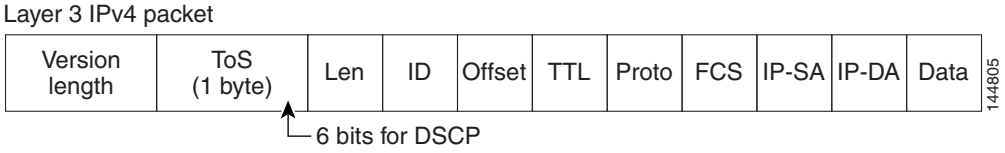
PFC QoS Glossary

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.

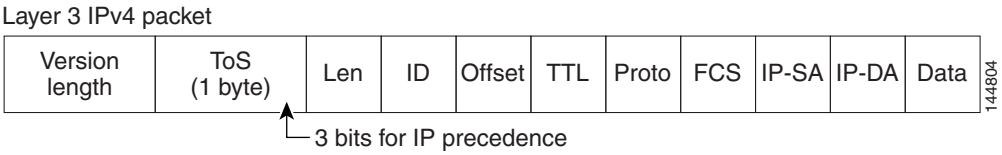


- *Classification* is the process used for selecting traffic to be marked for QoS.
- *Congestion avoidance* is the process by which PFC QoS reserves ingress and egress LAN port capacity for Layer 2 frames with high-priority Layer 2 CoS values. PFC QoS implements congestion avoidance with Layer 2 CoS 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 PFC and Distributed Forwarding Cards (DFCs). Policing can mark or drop traffic.
- *Queues*—Queues are allocations of buffer space used to temporarily store data on a port.
- *QoS labels*—PFC QoS uses CoS, DSCP, and IP Precedence as QoS labels. QoS labels are prioritization values carried in Layer 3 packets and Layer 2 frames.
- *Scheduling* is the assignment of Layer 2 frames to a queue. PFC QoS assigns frames to a queue based on Layer 2 CoS values.
- *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.

Troubleshooting MLS QoS

Table 44-5 lists some of the troubleshooting scenarios for MLS QoS.

Table 44-6 Troubleshooting MLS QoS

Problem	Solution
MLS QoS issues	You can configure mls qos commands such as **mls qos trust dscp/cos/ip prec** on the interface. To confirm issues with the mls qos commands, execute the following commands to view the user configuration of the policy-map. • **show run policy-map lan-police-ingress** • **show mls qos** • **show mls qos ip** • **show mls qos map**
The hardware configuration of Control Plane service-policy and Control Plan Policy not working as expected.	Execute the command **sh policy-map control-plane** to display the QoS statistics such as number of packets matched, dropped and, offered and drop rates for the policy-map applied on the control-plane interface.

Problem	Solution

Traffic classified incorrectly	• If trust is configured onthe input port, use the **show mls qos maps cos-dscp/dscp-cos/ip-prec-dscp/exp-dscp** command to check the mls qos mapping tables.
	• Use **show tcam interface interface qos type1/type2 ip** *detail* (type1 is for input policy, type2 for output policy) command to verify that the classification hardware parameters are configured correctly and packets are relayed to the right class as in this example:
	`Router#sh tcam interface gig10/1 qos type1 ip detail`
	`* Global Defaults shared`
	`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: 513 lookup_type: 1`
	`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 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 X--- 0 0 <-`
	`R rslt: 142811A8 <-`
	`V 36829 0.0.0.0 0.0.0.0 P=0 P=0 ------ 0 M--- 0 0 -- --- 0-0`
	`M 36836 0.0.0.0 0.0.0.0 0 0 ------ 0 X--- 0 0`

Problem	Solution
	`R rslt: 142811A8` **Note** **<-** indicates the class where the packets are being classified. • Use the **show run class-map** command to check the class-map definition. • Use the **show policy-map interface** *interface* command to check the classification statistics.
Packets wrongly marked or not being marked	• Use the **show policy-map interface** *interface-name* command to check the classification and marking counters. **Note** Marking statistics are not available for ES+ line cards. • Use the procedures at Traffic classified incorrectly, page 4-4 to confirm if the packet classification is performed correctly. • Use the Embedded Logic Analyzer Module (ELAM) capture tool to study the impacted packets on a c7600 router.
Queuing issues	Use the **show mls qos queuing interface** command to display the queuing details, trust state, default CoS, queue ID, scheduling, threshold value and the queuing mode. Study the details from the output to identify the issue. If the issue persists, contact TAC.

CHAPTER 42

Configuring PFC QoS Statistics Data Export

This chapter describes how to configure PFC QoS statistics data export on Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter contains these sections:

Understanding PFC QoS Statistics Data Export

The PFC QoS statistics data export feature generates per-LAN-port and per-aggregate policer utilization information and forwards this information in UDP packets to traffic monitoring, planning, or accounting applications. You can enable PFC QoS statistics data export on a per-LAN-port or on a per-aggregate policer basis. The statistics data generated per port consists of counts of the input and output packets and bytes. The aggregate policer statistics consist of counts of allowed packets and counts of packets exceeding the policed rate.

The PFC QoS statistics data collection occurs periodically at a fixed interval, but you can configure the interval at which the data is exported. PFC QoS statistics collection is enabled by default, and the data export feature is disabled by default for all ports and all aggregate policers configured on the Cisco 7600 series router.

Note The PFC QoS statistics data export feature is completely separate from NetFlow Data Export and does not interact with it.

PFC QoS Statistics Data Export Default Configuration

Table 42-1 shows the PFC QoS statistics data export default configuration.

Table 42-1 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 42-2
- Enabling PFC QoS Statistics Data Export for a Port, page 42-3
- Enabling PFC QoS Statistics Data Export for a Named Aggregate Policer, page 42-4
- Enabling PFC QoS Statistics Data Export for a Class Map, page 42-5
- Setting the PFC QoS Statistics Data Export Time Interval, page 42-6
- Configuring PFC QoS Statistics Data Export Destination Host and UDP Port, page 42-7
- Setting the PFC QoS Statistics Data Export Field Delimiter, page 42-9

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”)
- PFC or DFC slot number
- Number of in-profile bytes
- Number of bytes that exceed the CIR
- Number of bytes that exceed the PIR
- Time stamp

Enabling PFC QoS Statistics Data Export for a Class Map

To enable PFC QoS statistics data export for a class map, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos statistics-export class-map` *classmap_name*	Enables PFC QoS statistics data export for a class map.
	`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”)
  - PFC or DFC slot number
  - VLAN ID
  - Number of in-profile bytes
  - Number of bytes that exceed the CIR
  - Number of bytes that exceed the PIR
  - Time stamp
- For data from a port channel interface:
  - Export type (“6” for a class map and port channel)
  - Class map name
  - Direction (“in”)
  - PFC or DFC slot number
  - Port channel ID
  - Number of in-profile bytes
  - Number of bytes that exceed the CIR
  - Number of bytes that exceed the PIR
  - Time stamp

Setting the PFC QoS Statistics Data Export Time Interval

To set the time interval for the PFC QoS statistics data export, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos statistics-export interval` *interval_in_seconds*	Sets the time interval for the PFC QoS statistics data export. **Note** The interval needs to be short enough to avoid counter wraparound with the activity in your configuration, but because exporting PFC QoS statistic creates a significant load on the router, 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 42-2 lists the supported PFC QoS data export facility and severity parameter values.

Table 42-2 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 42-2 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 42-3 lists the supported PFC QoS data export severity parameter values.

Table 42-3 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 43

Configuring MPLS QoS on the PFC

This chapter describes how to configure Multiprotocol Label Switching (MPLS) quality of service (QoS) on a Cisco 7600 PFC3B, PFC3BXL, PFC3C, and PFC3CXL card. Unless otherwise noted, MPLS QoS operation is the same on all of these PFC cards.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- PFC-mode MPLS QoS provides MPLS traffic with the PFC QoS features described in Chapter 44, "Configuring PFC QoS."
- This chapter provides supplemental information about MPLS QoS features on the PFC. Be sure that you understand the PFC QoS features before you read this chapter.
- All policing and marking available for MPLS QoS on the PFC 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/qcfintro.html#wp998197

This chapter contains these sections:

Terminology

This section defines some MPLS QoS terminology:

- *Class of Service* (CoS) refers to three bits in either an Inter-Switch Link (ISL) header or an 802.1Q header that are used to indicate the priority of the Ethernet frame as it passes through a switched network. The CoS bits in the 802.1Q header are commonly referred to as the 802.1p bits. To maintain QoS when a packet traverses both Layer 2 and Layer 3 domains, the type of service (ToS) and CoS values can be mapped to each other.
- *Classification* is the process used for selecting traffic to be marked for QoS.
- *Differentiated Services Code Point* (DSCP) is the first six bits of the ToS byte in the IP header. DSCP is only present in an IP packet.
- *E-LSP* is a label switched path (LSP) on which nodes infer the QoS treatment for MPLS packets exclusively from the experimental (EXP) bits in the MPLS header. Because the QoS treatment is inferred from the EXP (both class and drop precedence), several classes of traffic can be multiplexed onto a single LSP (use the same label). A single LSP can support up to eight classes of traffic because the EXP field is a 3-bit field. The maximum number of classes would be less after reserving some values for control plane traffic or if some of the classes have a drop precedence associated with them.
- *EXP bits* define the QoS treatment (per-hop behavior) that a node should give to a packet. It is the equivalent of the DiffServ Code Point (DSCP) in the IP network. A DSCP defines a class and drop precedence. The EXP bits are generally used to carry all the information encoded in the IP DSCP. In some cases, however, the EXP bits are used exclusively to encode the dropping precedence.
- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *IP precedence* is the three most significant bits of the ToS byte in the IP header.
- *QoS tags* are prioritization values carried in Layer 3 packets and Layer 2 frames. A Layer 2 CoS label can have a value ranging between zero for low priority and seven for high priority. A Layer 3 IP precedence label can have a value ranging between zero for low priority and seven for high priority. IP precedence values are defined by the three most significant bits of the 1-byte ToS byte. A Layer 3 DSCP label can have a value between 0 and 63. DSCP values are defined by the six most significant bits of the 1-byte IP ToS field.
- *LERs* (label edge routers) are devices that impose and dispose of labels upon packets; also referred to as Provider Edge (PE) routers.
- *LSRs* (label switching routers) are devices that forward traffic based upon labels present in a packet; also referred to as Provider (P) routers.
- *Marking* is the process of setting a Layer 3 DSCP value in a packet. Marking is also the process of choosing different values for the MPLS EXP field to mark packets so that they have the priority that they require during periods of congestion.
- *Packets* carry traffic at Layer 3.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.

PFC-Mode MPLS QoS Features

QoS enables a network to provide improved service to selected network traffic. This section explains the following PFC-mode 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 PFC-mode MPLS QoS policy.

Trust

For received Layer 3 MPLS packets, the PFC usually trusts the EXP value in the received topmost label. None of the following have any effect on MPLS packets:

- Interface trust state
- Port CoS value
- Policy-map **trust** command

For received Layer 2 MPLS packets, the PFC can either trust the EXP value in the received topmost label or apply port trust or policy trust to the MPLS packets for CoS and egress queueing purposes.

Classification

Classification is the process that selects the traffic to be marked. Classification accomplishes this by partitioning traffic into multiple priority levels, or classes of service. Traffic classification is the primary component of class-based QoS provisioning. The PFC makes classification decisions based on the EXP bits in the received topmost label of received MPLS packets (after a policy is installed). See the "Configuring a Class Map to Classify MPLS Packets" section on page 43-20 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 PFC-mode 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 43-23 for information.

Preserving IP ToS

The PFC automatically preserves the IP ToS during all MPLS operations including imposition, swapping, and disposition.You do not need to enter a command to save the IP ToS.

EXP Mutation

You can configure 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 PFC-Mode MPLS QoS Egress EXP Mutation" section on page 43-28.

MPLS DiffServ Tunneling Modes

The PFC uses MPLS DiffServ tunneling modes. Tunneling provides QoS transparency from one edge of a network to the other edge of the network. See the "MPLS DiffServ Tunneling Modes" section on page 43-31 for information.

PFC-Mode MPLS QoS Overview

PFC-mode 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.

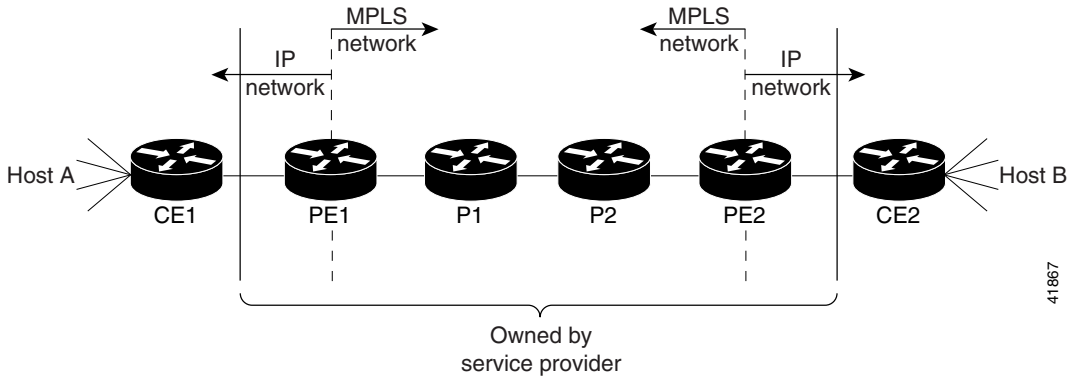
For more information, see the "MPLS DiffServ Tunneling Modes" section on page 43-31.

PFC-Mode MPLS QoS

This section describes how PFC-mode MPLS QoS works.

Figure 43-1 shows an MPLS network of a service provider that connects two sites of a customer network.

Figure 43-1 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 43-1, 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.

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 PFC receives the traffic from the input interface and uses the 802.1p bits or the IP ToS bits to determine the EXP bits and to perform any classification, marking, and policing. For classification of incoming IP packets, the input service policy can also use access control lists (ACLs).
3. For each incoming packet, the PFC performs a lookup on the IP address to determine the next-hop router.
4. The appropriate label is pushed (imposition) into the packet, and the EXP value resulting from the QoS decision is copied into the MPLS EXP field in the label header.
5. The PFC forwards the labeled packets to the appropriate output interface for processing.
6. The PFC also forwards the 802.1p bits or the IP ToS bits to the output interface.
7. At the output interface, the labeled packets are differentiated by class for marking or policing. For LAN interfaces, egress classification is still based on IP, not on MPLS.
8. The labeled packets (marked by EXP) are sent to the core MPLS network.

LSRs in the Core of an MPLS Network

This section describes how LSRs used at the core of an MPLS network process packets:

1. Incoming MPLS-labeled packets (and 802.1p bits or IP ToS bits) from an edge LER (or other core device) arrive at the core LSR.
2. The PFC receives the traffic from the input interface and uses the EXP bits to perform classification, marking, and policing.

3. The PFC performs a table lookup to determine the next-hop LSR.
4. An appropriate label is placed (swapped) into the packet and the MPLS EXP bits are copied into the label header.
5. The PFC forwards the labeled packets to the appropriate output interface for processing.
6. The PFC also forwards the 802.1p bits or the IP ToS bits to the output interface.
7. The outbound packet is differentiated by the MPLS EXP field for marking or policing.
8. The labeled packets (marked with EXP) are sent to another LSR in the core MPLS network or to an LER at the output edge.

Note Within the service provider network, there is no IP precedence field for the queueing algorithm to use because the packets are MPLS packets. The packets remain MPLS packets until they arrive at PE2, the provider edge router.

LERs at the Output Edge of an MPLS Network

At the egress side of an MPLS network, LERs process packets as follows:

1. MPLS-labeled packets (and 802.1p bits or IP ToS bits) from a core LSR arrive at the egress LER (PE2) from the MPLS network backbone.
2. The PFC pops the MPLS labels (disposition) from the packets. Aggregate labels are classified using the original 802.1p bits or the IP ToS bits. Nonaggregate labels are classified with the EXP value by default.
3. For aggregate labels, the PFC performs a lookup on the IP address to determine the packet's destination; the PFC then forwards the packet to the appropriate output interface for processing. For non-aggregate labels, forwarding is based on the label. By default, non-aggregate labels are popped at the penultimate-hop router (next to last), not the egress PE router.
4. The PFC also forwards the 802.1p bits or the IP ToS bits to the output interface.
5. The packets are differentiated according to the 802.1p bits or the IP ToS bits and treated accordingly.

Note The MPLS EXP bits allow you to specify the QoS for an MPLS packet. The IP precedence and DSCP bits allow you to specify the QoS for an IP packet.

Understanding PFC-Mode MPLS QoS

PFC-mode MPLS QoS supports IP QoS. For MPLS packets, the EXP value is mapped into an internal DSCP so that the PFC can apply non-MPLS QoS marking and policing.

For both the ingress and egress policies, PFC-mode MPLS QoS marking and policing decisions are made on a per-interface basis at an ingress PFC. The ingress interfaces are physical ports, subinterfaces, or VLANs.

The QoS policy ACLs are programmed in QoS TCAM separately for ingress and egress lookup. The ternary content addressable memory (TCAM) egress lookup takes place after the IP forwarding table (FIB) and NetFlow lookups are completed.

The results of each QoS TCAM lookup yield an index into RAM that contains policer configuration and policing counters. Additional RAM contains the microflow policer configuration; the microflow policing counters are maintained in the respective NetFlow entries that match the QoS ACL.

The results of ingress and egress aggregate and microflow policing are combined into a final policing decision. The out-of-profile packets can be either dropped or marked down in the DSCP.

This section describes PFC-mode MPLS QoS for the following:

- LERs at the EoMPLS Edge, page 43-8
- LERs at the IP Edge (MPLS, MPLS VPN), page 43-9
- LSRs at the MPLS Core, page 43-13

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, PFC-mode 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 43-9
- Classification for IP-to-MPLS PFC-Mode MPLS QoS, page 43-10
- Classification at IP-to-MPLS Ingress Port, page 43-10
- Classification at IP-to-MPLS Egress Port, page 43-10

MPLS to Ethernet

For MPLS to Ethernet, the ingress interface, PFC-mode 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 43-11
- Classification for MPLS-to-IP PFC3BXL or PFC3B Mode MPLS QoS, page 43-11
- Classification at MPLS-to-IP Ingress Port, page 43-11
- Classification at MPLS-to-IP Egress Port, page 43-12.

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 43-12 for additional MPLS VPN-specific QoS information.

IP to MPLS

The PFC provides the following MPLS QoS capabilities at the IP-to-MPLS edge:

- Assigning an EXP value based on the **mls qos trust** or **policy-map** command
- Marking an EXP value using a policy
- Policing traffic using a policy

This section provides information about the MPLS QoS classification that the PFC3BXL or 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 PFC ingress and egress policies for IP traffic classify traffic on the original received IP using **match** commands for IP precedence, IP DSCP, and IP ACLs. Egress policies do not classify traffic on the imposed EXP value nor on a marking done by an ingress policy.

After the PFC applies the port trust and QoS policies, it assigns the internal DSCP. The PFC then assigns the EXP value based on the internal DSCP-to-EXP global map for the labels that it imposes. If more than one label is imposed, the EXP value is the same in each label. The PFC' preserves the original IP ToS when the MPLS labels are imposed.

The PFC assigns the egress CoS based on the internal DSCP-to-CoS global map. If the default internal DSCP-to-EXP and the internal DSCP-to-CoS maps are consistent, then the egress CoS has the same value as the imposed EXP.

If the ingress port receives both IP-to-IP and IP-to-MPLS traffic, classification should be used to separate the two types of traffic. For example, if the IP-to-IP and IP-to-MPLS traffic have different destination address ranges, you can classify traffic on the destination address, and then apply IP ToS policies to the IP-to-IP traffic and apply a policy (that marks or sets the EXP value in the imposed MPLS header) to the IP-to-MPLS traffic. See the following two examples:

- A policy to mark IP ToS sets the internal DSCP—If it is applied to all traffic, then for IP-to-IP traffic, the egress port will rewrite the CoS (derived from the internal DSCP) to the IP ToS byte in the egress packet. For IP-to-MPLS traffic, the PFC maps the internal DSCP to the imposed EXP value.
- A policy to mark MPLS EXP sets the internal DSCP—If it is applied to all traffic, then for IP-to-IP traffic, the egress port rewrites the IP ToS according to the ingress IP policy (or trust). The CoS is mapped from the ToS. For IP-to-MPLS traffic, the PFC maps the internal DSCP to the imposed EXP value.

Classification for IP-to-MPLS PFC-Mode MPLS QoS

PFC-mode 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

PFC-mode 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

PFC-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 PFC assigns the internal DSCP (internal priority that is assigned to each frame) based on the QoS result. The QoS result is affected by the following:

- Default trust EXP value
- Label type (per-prefix or aggregate)
- Number of VPNs
- Explicit NULL use
- QoS policy

There are three different classification modes:

- Regular MPLS classification—For nonaggregate labels, in the absence of MPLS recirculation, the PFC classifies the packet based on MPLS EXP ingress or egress policy. The PFC3BXL and PFC3CXL queue the packet based on COS derived from EXP-to-DSCP-to-CoS mapping. The underlying IP DSCP is either preserved after egress decapsulation, or overwritten from the EXP (through the EXP-to-DSCP map).
- IP classification for aggregate label hits in VPN CAM—The PFC does one of the following:
  - Preserves the underlying IP ToS
  - Rewrites the IP ToS by a value derived from the EXP-to-DSCP global map
  - Changes the IP ToS to any value derived from the egress IP policy

  In all cases, egress queueing is based on the final IP ToS from the DSCP-to-CoS map.
- IP classification with aggregate labels not in VPN CAM—After recirculation, the PFC differentiates the MPLS-to-IP packets from the regular IP-to-IP packets based on the ingress reserved VLAN specified in the MPLS decapsulation adjacency. The reserved VLAN is allocated per VRF both for VPN and non-VPN cases. The ingress ToS after recirculation can be either the original IP ToS value, or derived from the original EXP value. The egress IP policy can overwrite this ingress ToS to an arbitrary value.

Note For information about recirculation, see the "Recirculation" section on page 24-4.

For incoming MPLS packets on the PE-to-CE ingress, the PFC supports MPLS classification only. Ingress IP policies are not supported. PE-to-CE traffic from the MPLS core is classified or policed on egress as IP.

Classification for MPLS-to-IP PFC3BXL or PFC3B Mode MPLS QoS

PFC-mode MPLS QoS at the ingress to PE2 supports matching on the EXP value and the **police** command.

PFC-mode 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 PFC 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 PFC3BXLs or PFC3Bs.

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 24-4.

The PFC propagates the EXP value if all interfaces in the VPN have EXP propagation enabled.

The following PE MPLS QoS features are supported:

- General MPLS QoS features for IP packets
- Classification, policing, or marking of CE-to-PE IP traffic through the VPN subinterface
- Per-VPN QoS (per-port, per-VLAN, or per-subinterface)

LSRs at the MPLS Core

This section provides information about MPLS QoS features for LSRs at the core (MPLS-to-MPLS) for MPLS and MPLS VPN networks. Ingress features, egress interface, and PFC features for Carrier Supporting Carrier (CsC) QoS features are similar to those used with MPLS to MPLS described in the next section. A difference between CsC and MPLS to MPLS is that with CsC labels can be imposed inside the MPLS domain.

MPLS to MPLS

PFC-mode 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 PFC-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 PFC ignores the port trust state, the ingress CoS, and any policy-map **trust** commands. Instead, the PFC trusts the EXP value in the topmost label.

Note PFC-mode 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.

PFC-mode MPLS QoS maps the EXP value to the internal DSCP using the EXP-to-DSCP global map. What the PFC does next depends on whether it is swapping labels, imposing a new label, or popping a label:

- Swapping labels—When swapping labels, the PFC' preserves the EXP value in the received topmost label and copies it to the EXP value in the outgoing topmost label. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP global maps are consistent, then the egress CoS is based on the EXP in the outgoing topmost label.

  The PFC can mark down out-of-profile traffic using the **police** command's **exceed** and **violate** actions. It does not mark in-profile traffic, so the **conform** action must be transmitted and the **set** command cannot be used. If the PFC is performing a markdown, it uses the internal DSCP as an index into the internal DSCP markdown map. The PFC maps the result of the internal DSCP markdown to an EXP value using the internal DSCP-to-EXP global map. The PFC rewrites the new EXP value to the topmost outgoing label and does not copy the new EXP value to the other labels in the stack. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, then the egress CoS is based on the EXP value in the topmost outgoing label.

- Imposing an additional label—When imposing a new label onto an existing label stack, the PFC maps the internal DSCP to the EXP value in the imposed label using the internal DSCP-to-EXP map. It then copies the EXP value in the imposed label to the underlying swapped label. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, the egress CoS is based on the EXP value in the imposed label.

  The PFC can mark in-profile and mark down out-of-profile traffic. After it marks the internal DSCP, the PFC uses the internal DSCP-to-EXP global map to map the internal DSCP to the EXP value in the newly imposed label. The PFC then copies the EXP in the imposed label to the underlying swapped label. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. Therefore, the egress CoS is based on the EXP in the imposed label.
- Popping a label—When popping a label from a multi-label stack, the PFC preserves the EXP value in the exposed label. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, then the egress CoS is based on the EXP value in the popped label.
- If EXP propagation is configured for the egress interface, the PFC maps the internal DSCP to the EXP value in the exposed label using the DSCP-to-EXP global map. The PFC assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, the egress CoS is based on the EXP value in the exposed label.

Classification for MPLS-to-MPLS PFC-Mode MPLS QoS

PFC-mode 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

PFC-mode 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 PFC. 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 PFC. 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.

PFC MPLS QoS Default Configuration

This section describes the MPLS QoS default configuration on the PFC. The PFC has the following global MPLS QoS settings:

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

The Cisco 7600 PFC 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 44-45.

The following commands are not supported:

- **set qos-group**
- **set discard-class**

PFC-Mode MPLS QoS Restrictions and Guidelines

When configuring MPLS QoS on the PFC, follow these guidelines and restrictions:

- For IP-to-MPLS or EoMPLS imposition when the received packet is an IP packet:
  - When QoS is disabled, the EXP value is based on the received IP ToS.
  - When QoS is queuing only, the EXP value is based on the received IP ToS.
- For EoMPLS imposition when the received packet is a non-IP packet:
  - When QoS is disabled, the EXP value is based on the ingress CoS.
  - When QoS is queuing only, the EXP value is based on the received IP ToS.
- For MPLS-to-MPLS operations:
  - Swapping when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Swapping when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Imposing additional label when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Imposing an additional label when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Popping one label when QoS is disabled, the EXP value is based on the underlying EXP value.
  - Popping one label when QoS is queuing only, the EXP value is based on the underlying EXP value.
- EXP value is irrelevant to MPLS-to-IP disposition.
- The **no mls qos rewrite ip dscp** command is incompatible with MPLS. The default **mls qos rewrite ip dscp** command must remain enabled in order for the PFC to assign the correct EXP value for the labels that it imposes.

Configuring MPLS QoS on the PFC

These sections describe how to configure MPLS QoS on the PFC:

Enabling QoS Globally

Before you can configure QoS on the PFC, you must enable the QoS functionality globally using the **mls qos** command. This command enables default QoS conditioning of traffic.

When the **mls qos** command is enabled, the PFC assigns a priority value to each frame. This value is the internal DSCP. The internal DSCP is assigned based on the contents of the received frame and the QoS configuration. This value is rewritten to the egress frame's CoS and ToS fields.

To enable QoS globally, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos`	Enables PFC QoS globally on the router.
	`Router(config)# no mls qos`	Disables PFC QoS globally on the router.
Step 2	`Router(config)# end`	Exits configuration mode.
Step 3	`Router# show mls qos`	Verifies the configuration.

This example shows how to enable QoS globally:

```
Router(config)# mls qos
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos
QoS is enabled globally
  Microflow policing is enabled globally
  QoS ip packet dscp rewrite enabled globally

  Qos trust state is DSCP on the following interfaces:
    Gi4/1 Gi4/1.12

  Qos trust state is IP Precedence on the following interfaces:
    Gi4/2 Gi4/2.42
  Vlan or Portchannel(Multi-Earl) policies supported: Yes
  Egress policies supported: Yes


 ----- Module [5] -----
  QoS global counters:
    Total packets: 5957870
    IP shortcut packets: 0
    Packets dropped by policing: 0
    IP packets with TOS changed by policing: 6
    IP packets with COS changed by policing: 0
    Non-IP packets with COS changed by policing: 3
    MPLS packets with EXP changed by policing: 0
```

Enabling Queueing-Only Mode

To enable queueing-only mode, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls qos queueing-only`	Enables queueing-only mode.
	`Router(config)# no mls qos queueing-only`	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 router applies the port CoS value to untagged ingress traffic and to traffic that is received through ports that cannot be configured to trust CoS.

This example shows how to enable queueing-only mode:

```
Router# configure terminal
Router(config)# mls qos queueing-only
Router(config)# end
Router#
```

Restrictions and Usage Guidelines

If QoS is disabled (**no mls qos**) for the PFC, the EXP value is determined as follows:

- For IP-to-MPLS or EoMPLS imposition when the received packet is an IP packet:
  - When QoS is disabled (**no mls qos**), the EXP value is based on the received IP ToS.
  - When QoS is queuing only (**mls qos queueing-only**), the EXP value is based on the received IP ToS.
- For EoMPLS imposition when the received packet is a non-IP packet:
  - When QoS is disabled, the EXP value is based on the ingress CoS.
  - When QoS is queuing only, the EXP value is based on the received IP ToS.
- For MPLS-to-MPLS operations:
  - Swapping when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Swapping when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Imposing an additional label when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation).

  - Imposing additional label when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation).
  - Popping one label when QoS is disabled, the EXP value is based on the underlying EXP value.
  - Popping one label when QoS is queuing only, the EXP value is based on the underlying EXP value.
- EXP value is irrelevant to MPLS-to-IP disposition.

Configuring a Class Map to Classify MPLS Packets

You can use the **match mpls experimental topmost** command to define traffic classes inside the MPLS domain by packet EXP values. This allows you to define service policies to police the EXP traffic on a per-interface basis by using the **police** command.

To configure a class map, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# class-map` *class_name*	Specifies the class map to which packets will be matched.
Step 2	`Router(config-cmap)# match mpls experimental topmost` *value*	Specifies the packet characteristics that will be matched to the class.
Step 3	`Router(config-cmap)# exit`	Exits class-map configuration mode.

This example shows that all packets that contain MPLS experimental value 3 are matched by the traffic class named exp3:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# class-map exp3
Router(config-cmap)# match mpls experimental topmost 3
Router(config-cmap)# exit
Router(config)# policy-map exp3
Router(config-pmap)# class exp3
Router(config-pmap-c)# police 1000000 8000000 conform-action transmit exceed-action drop
Router(config-pmap-c)# exit
Router(config-pmap)# end
Router# show class exp3
 Class Map match-all exp3 (id 61)
   Match mpls experimental topmost 3
Router# show policy-map exp3
  Policy Map exp3
    Class exp3
     police cir 1000000 bc 8000000 be 8000000 conform-action transmit exceed-action drop
Router# show running-config interface fastethernet 3/27
Building configuration...

Current configuration : 173 bytes
!
interface FastEthernet3/27
  ip address 47.0.0.1 255.0.0.0
  tag-switching ip
 end

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

```
Router(config)# interface fastethernet 3/27
Router(config-if)# service-policy input exp3
Router(config-if)#
Router#
Enter configuration commands, one per line.  End with CNTL/Z.
Router# show running-config interface fastethernet 3/27
Building configuration...

Current configuration : 173 bytes
!
interface FastEthernet3/27
  ip address 47.0.0.1 255.0.0.0
  tag-switching ip
  service-policy input exp3
 end

Router#
1w4d: %SYS-5-CONFIG_I: Configured from console by console
Router# show mls qos mpls
 QoS Summary [MPLS]:      (* - shared aggregates, Mod - switch module)

      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
------------------------------------------------------------------------------
    Fa3/27  5  In       exp3    0    2   dscp  0              0              0

       All  5   -    Default    0    0*    No  0     3466140423              0
Router# show policy-map interface fastethernet 3/27
 FastEthernet3/27

  Service-policy input: exp3

    class-map: exp3 (match-all)
      Match: mpls experimental topmost 3
      police :
        1000000 bps 8000000 limit 8000000 extended limit
      Earl in slot 5 :
        0 bytes
        5 minute offered rate 0 bps
        aggregate-forwarded 0 bytes action: transmit
        exceeded 0 bytes action: drop
        aggregate-forward 0 bps exceed 0 bps

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/27
Router(config-if)# service-policy output ip2tag
Router(config-if)# end
Router# show mls qos ip
 QoS Summary [IPv4]:      (* - shared aggregates, Mod - switch module)

      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
------------------------------------------------------------------------------
     Vl300  5  In          x   44    1     No  0              0              0
    Fa3/27  5 Out      iptcp   24    2     --  0              0              0

       All  5   -    Default    0    0*    No  0     3466610741              0
```

Restrictions and Usage Guidelines

The following restrictions and guidelines apply when classifying MPLS packets:

- The **match mpls experimental** command specifies the name of an EXP field value to be used as the match criterion against which packets are checked to determine if they belong to the class specified by the class map.
- To use the **match mpls experimental** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. After you identify the class, you can use the **match mpls experimental** command to configure its match criteria.
- If you specify more than one command in a class map, only the last command entered applies. The last command overrides the previously entered commands.

Configuring the MPLS Packet Trust State on Ingress Ports

You can use the **no mls qos mpls trust exp** command to apply port or policy trust to MPLS packets in the same way that you apply them to Layer 2 packets.

To configure the MPLS packet trust state of an ingress port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {{type slot/port} | {port-channel number}}`	Selects the interface to configure.
Step 2	`Router(config-if)# no mls qos mpls trust exp`	Sets the trust state of an MPLS packet so that all trusted cases (trust cos, trust dscp, trust ip-precedence) are treated as trust-cos.
	`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 on the PFC does not attempt to apply commands from more than one policy map class to matched traffic.

Configuring a Policy Map to Set the EXP Value on All Imposed Labels

To set the value of the MPLS EXP field on all imposed label entries, use the **set mpls experimental imposition** command in QoS policy-map class configuration mode. To disable the setting, use the no form of this command.

Note The **set mpls experimental imposition** command replaces the **set mpls experimental** command.

	Command	Purpose
Step 1	`Router(config)# policy-map` *policy_name*	Creates a policy map.
Step 2	`Router(config-pmap)# class-map` *name* [**match-all** | **match-any**]	Accesses the QoS class-map configuration mode to configure QoS class maps.
Step 3	`Router(config-pmap-c)# set mpls experimental imposition` {*mpls-exp-value* | *from-field* [**table** *table-map-name*]}	Sets the value of the MPLS experimental (EXP) field on all imposed label entries.
Step 4	`Router(config-pmap-c)# exit`	Exits class-map configuration mode.

The following example sets the MPLS EXP imposition value according to the DSCP value defined in the MPLS EXP value 3.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# access-l 101 p tcp any any
Router(config)# class-map iptcp
Router(config-cmap)# match acc 101
Router(config-cmap)# exit
Router(config)#
Router(config-cmap)# policy-map ip2tag
Router(config-pmap)# class iptcp
Router(config-pmap-c)# set mpls exp imposition 3
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#
Router#
1w4d: %SYS-5-CONFIG_I: Configured from console by console
Router#
Router# show policy-map ip2tag
  Policy Map ip2tag
    Class iptcp
      set mpls experimental imposition 3
Router# show class iptcp
 Class Map match-all iptcp (id 62)
   Match access-group101

Router# configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastethernet 3/27
Router(config-if)# ser in ip2tag
Router(config-if)#
Routers
1w4d: %SYS-5-CONFIG_I: Configured from console by console
Router# show pol ip2tag
  Policy Map ip2tag
    Class iptcp
      set mpls experimental imposition 3
Router# show class-map iptcp
 Class Map match-all iptcp (id 62)
   Match access-group  101

Router# show access-l 101
Extended IP access list 101
    10 permit tcp any any
Router# show mls qos ip
 QoS Summary [IPv4]:      (* - shared aggregates, Mod - switch module)

      Int Mod Dir  Class-map DSCP  Agg  Trust Fl   AgForward-By   AgPoliced-By
                                   Id         Id
-------------------------------------------------------------------------------
    Fa3/27  5  In      iptcp   24    2     No  0              0              0
     Vl300  5  In          x   44    1     No  0              0              0

       All  5   -    Default    0    0*    No  0     3466448105              0
Router#
Router# show policy-map interface fastethernet 3/27
 FastEthernet3/27

  Service-policy input: ip2tag

    class-map: iptcp (match-all)
      Match: access-group 101
      set mpls experimental 3:
      Earl in slot 5 :
        0 bytes
        5 minute offered rate 0 bps
        aggregate-forwarded 0 bytes

    class-map: class-default (match-any)
      Match: any

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

This example shows how to verify the configuration:

```
Router# show policy map ip2tag
  Policy Map ip2tag
    Class iptcp
      set mpls experimental imposition 3
```

EXP Value Imposition Guidelines and Restrictions

When setting the EXP value on all imposed labels, follow these guidelines and restrictions:

- Use the **set mpls experimental imposition** command during label imposition. This command sets the MPLS EXP field on all imposed label entries.

- The **set mpls experimental imposition** command is supported only on input interfaces (imposition).
- The **set mpls experimental imposition** command does not mark the EXP value directly; instead, it marks the internal DSCP that is mapped to EXP through the internal DSCP-to-EXP global map.
- It is important to note that classification (based on the original received IP header) and marking (done to the internal DSCP) do not distinguish between IP-to-IP traffic and IP-to-MPLS traffic. The commands that you use to mark IP ToS and mark EXP have the same result as when you mark the internal DSCP.
- To set the pushed label entry value to a value different from the default value during label imposition, use the **set mpls experimental imposition** command.
- You optionally can use the **set mpls experimental imposition** command with the IP precedence, DSCP field, or QoS IP ACL to set the value of the MPLS EXP field on all imposed label entries.
- When imposing labels onto the received IP traffic with the PFC, you can mark the EXP field using the **set mpls experimental imposition** command.

For more information on this command, see the *Cisco IOS Switching Services Command Reference, Release 12.3* located at this URL:

http://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swi_n1.html#wp1092877

Configuring a Policy Map Using the Police Command

Policing is a function in the PFC hardware that provides the ability to rate limit a particular traffic class to a specific rate. The PFC supports aggregate policing and microflow policing.

Aggregate policing meters all traffic that ingresses into a port, regardless of different source, destination, protocol, source port, or destination port. Microflow policing meters all traffic that ingresses into a port, on a per flow (per source, destination, protocol, source port, and destination port). For additional information on aggregate and microflow policing, see the "Policers" section on page 44-18.

To configure traffic policing, use the **police** command. For information on this command, see the *Cisco 7600 Series Router Cisco IOS Command Reference*.

	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 44-61.
- 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 PFC, you can mark down out-of-profile traffic using the **police** command **exceed-action policed-dscp-transmit** and **violate-action policed-dscp-transmit** keywords. The PFC does not mark in-profile traffic; when marking down out-of-profile traffic, the PFC marks the outgoing topmost label. The PFC does not propagate the marking down through the label stack.
- With MPLS, the flow key is based on the label and EXP value; there is no flowmask option. Otherwise, flow key operation is similar to IP-to-IP. See the "Configuring a Policy Map" section on page 44-61.
- 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 PFC, you can mark the EXP field using the **conform-action set-mpls-exp-imposition-transmit** keywords.
- During IP-to-MPLS imposition, IP ToS marking is not supported. If you configure a policy to mark IP ToS, the PFC marks the EXP value.

Displaying a Policy Map

You can display a policy map with an interface summary for MPLS QoS classes or with the configuration of all classes configured for all service policies on the specified interface.

Displaying a PFC-Mode MPLS QoS Policy Map Class Summary

To display a PFC-mode 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 PFC-mode MPLS QoS policy map class summary.

This example shows how to display a PFC-mode 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 PFC-Mode MPLS QoS Egress EXP Mutation

These sections describe how to configure MPLS QoS egress EXP mutation on the PFC:

- Configuring Named EXP Mutation Maps, page 43-29
- Attaching an Egress EXP Mutation Map to an Interface, page 43-29

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

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 PFC, there are two ways to forward packets through a network:

- Short Pipe mode—In Short Pipe mode, the egress PE router uses the original packet marking instead of the marking used by the intermediate provider (P) routers. EXP marking does not propagate to the packet ToS byte.

  For a description of this mode, see the "Short Pipe Mode" section on page 43-31.

  For the configuration information, see the "Configuring Short Pipe Mode" section on page 43-34.

- 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 43-32.

  For the configuration procedure, see the "Configuring Uniform Mode" section on page 43-39.

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/12_2t/12_2t13/feature/guide/ftdtmode.html

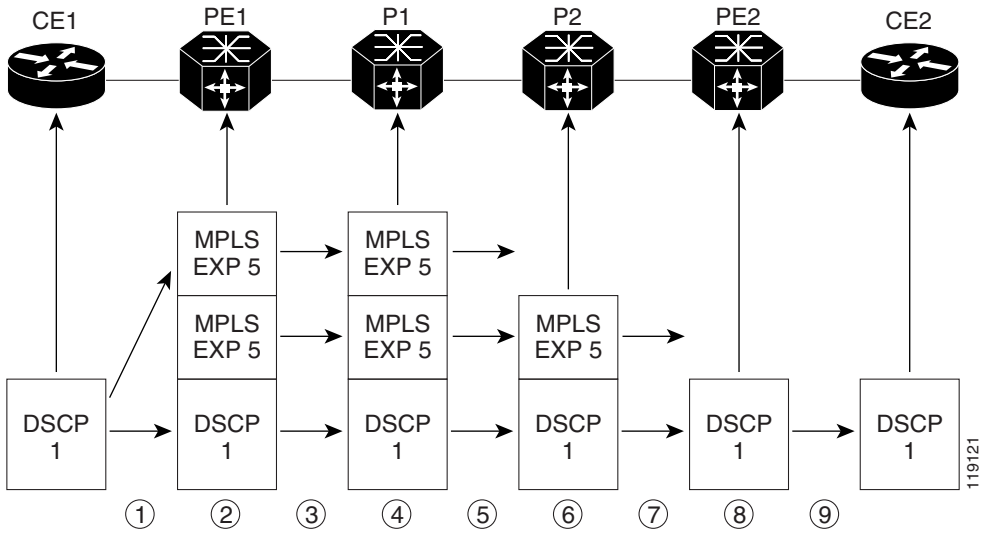
Short Pipe Mode

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 43-2 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/12_2t/12_2t13/feature/guide/ftdtmode.html

Short Pipe Mode Restrictions and Guidelines

The following restriction applies to Short Pipe mode:

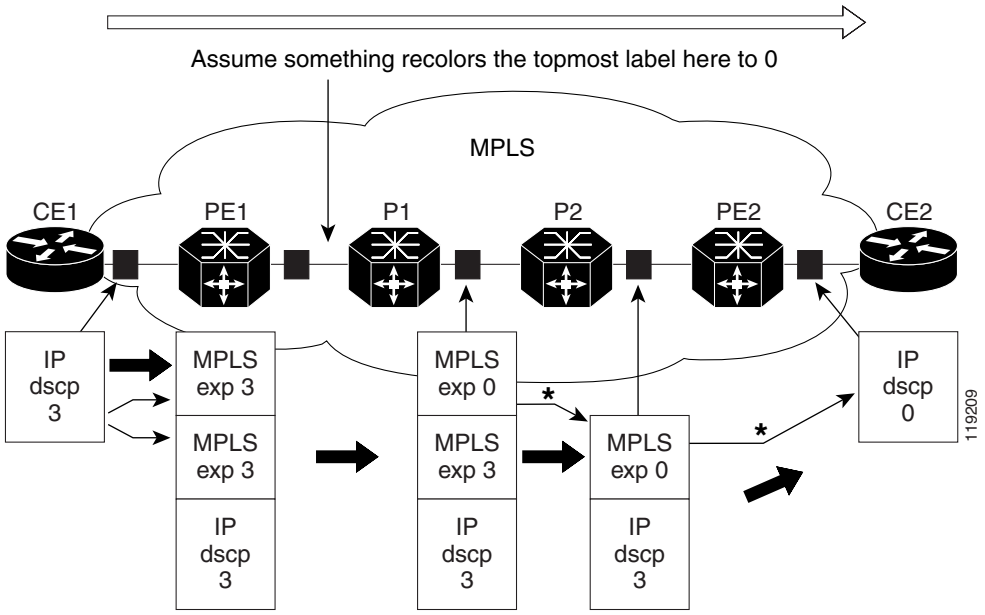
- Short Pipe mode is not supported if the MPLS-to-IP egress interface is EoMPLS (the adjacency has the end of marker (EOM) bit set).

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 43-3 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 43-34
- Configuring Ingress PE Router—P Facing Interface, page 43-35
- Configuring the P Router—Output Interface, page 43-37
- Configuring the Egress PE Router—Customer Facing Interface, page 43-38

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 PFC IP policies. For information about the classification for PFC IP policies based on IP ACL/DSCP/precedence, see http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/software/122sx/swcg/qos.htm.

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 QoS functionality.
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 QoS functionality.
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 QoS functionality.
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 QoS functionality.
Step 2	`Router(config)# class-map` *class_name*	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	`Router(config-c-map)# match ip dscp` *dscp_values*	Uses the DSCP values as the match criteria.
Step 4	`Router(config)# policy-map` *name*	Configures the QoS policy for packets that match the class or classes.
Step 5	`Router(config-p-map)# class` *class_name*	Associates the traffic class with the service policy.
Step 6	`Router(config-p-map-c)# bandwidth` {*bandwidth_kbps* | **percent** *percent*}	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	`Router(config-p-map)# class class-default`	Specifies the default class so that you can configure or modify its policy.
Step 8	`Router(config-p-map-c)# random-detect dscp-based`	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 9	`Router(config)# interface` *type slot/port*	Selects an interface to configure.
Step 10	`Router(config-if)# service-policy output` *name*	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Note The **bandwidth** command and **random-detect** command are not supported on LAN ports.

Configuration Example

This example shows how to classify a packet based on its IP DSCP value and provide appropriate discard and scheduling treatments:

```
Router(config)# mls qos
Router(config)# class-map IP-PREC-4
Router(config-c-map)# match ip precedence 4
Router(config)# policy-map output-qos
Router(config-p-map)# class IP-PREC-4
Router(config-p-map-c)# bandwidth percent 40
Router(config-p-map)# class class-default
Router(config-p-map-c)# random-detect
Router(config)# interface GE-WAN 3/2.32
Router(config-if)# service-policy output output-qos
```

Configuring Uniform Mode

This section describes how to configure the following:

Note The steps that follow show one way, but not the only way, to configure the Uniform mode.

Configuring the Ingress PE Router—Customer Facing Interface

For Uniform mode, setting the trust state to IP precedence or IP DSCP allows the PFC' to copy the IP PHB into the MPLS PHB.

Note This description applies to PFC QoS for LAN 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 QoS functionality.
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.
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 QoS functionality.
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 QoS functionality.
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 44

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 Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding IEEE 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 router port and assigns the port to a VLAN before making available any services offered by the router or the LAN.

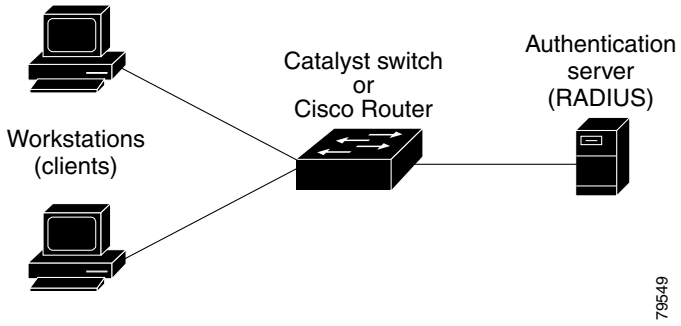
Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

These sections describe 802.1X port-based authentication:

Device Roles

With IEEE 802.1X port-based authentication, the devices in the network have specific roles as shown in Figure 44-1.

Figure 44-1 802.1X Device Roles



The specific roles shown in Figure 44-1 are as follows:

- *Client*—The device (workstation) that requests access to the LAN and router services and responds to requests from the router. 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 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 router whether or not the client is authorized to access the LAN and router services. Because the router 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.

- *Router* (also called the *authenticator* and *back-end authenticator*)—Controls the physical access to the network based on the authentication status of the client. The router 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 router includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

  When the router receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is re-encapsulated 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 router 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 router or the client can initiate authentication. If you enable authentication on a port by using the **dot1x port-control auto** interface configuration command, the router must initiate authentication when it determines that the port link state transitions from down to up. The router then sends an EAP-request/identity frame to the client to request its identity (typically, the router 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.

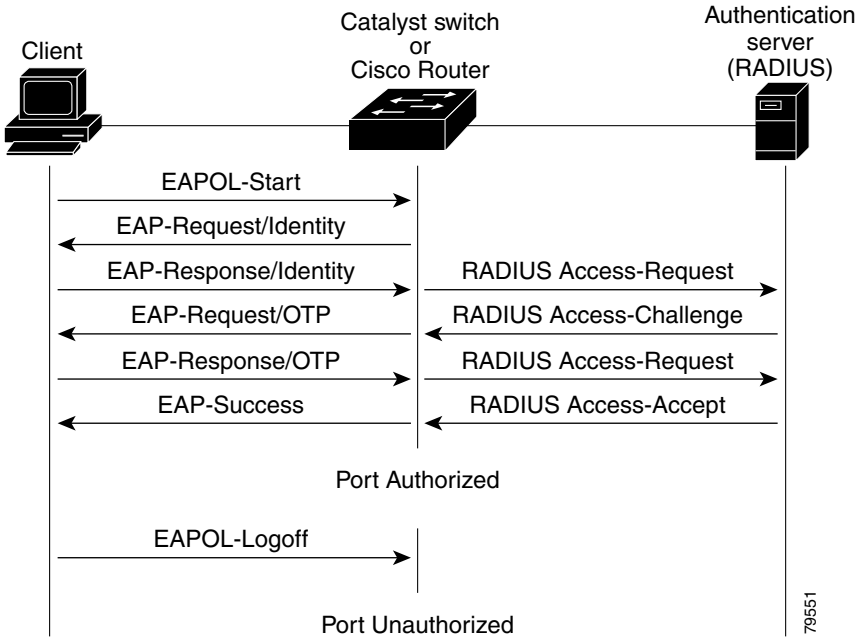
If the client does not receive an EAP-request/identity frame from the router during bootup, the client can initiate authentication by sending an EAPOL-start frame, which prompts the router to request the client's identity.

Note If IEEE 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 44-4.

When the client supplies its identity, the router 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 router port becomes authorized. For more information, see the "Ports in Authorized and Unauthorized States" section on page 44-4.

The specific exchange of EAP frames depends on the authentication method being used. Figure 44-2 shows a message exchange initiated by the client using the One-Time-Password (OTP) authentication method with a RADIUS server.

Figure 44-2 Message Exchange



Ports in Authorized and Unauthorized States

The router 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 IEEE 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 router 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 router 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 router 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 router 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 router 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 router 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.

Using IEEE 802.1X Authentication with DHCP Snooping

When the Dynamic Host Configuration Protocol (DHCP) snooping option-82 with data insertion feature is enabled, the router can insert a client's IEEE 802.1X-authenticated user identity information into the DHCP discovery process, allowing the DHCP server to assign IP addresses from different IP address pools to different classes of end users. This feature allows you to secure the IP addresses given to the end users for accounting purposes and to grant services based on Layer 3 criteria.

After a successful 802.1X authentication, the port is put into the forwarding state and stores the attributes that it receives from the RADIUS server. While performing DHCP snooping, the router acts as a DHCP relay agent, receiving DHCP messages and regenerating those messages for transmission on another interface.

After 802.1X authentication, when a client sends a DHCP discovery message, the router receives the packet and adds a RADIUS attributes suboption section to the packet containing the stored RADIUS attributes of the client. The router then submits the discovery broadcast again. The DHCP server receives the modified DHCP discovery packet and can, if configured to do so, use the authenticated user identity information when creating the IP address assignment.

The mapping of user to IP address can be on a one-to-one, one-to-many, or many-to-many basis. The one-to-many mapping allows the same user to authenticate through 802.1X hosts on multiple ports.

When 801.X authentication and DHCP snooping option-82 with data insertion features are enabled, the router will automatically insert the authenticated user identity information. To configure DHCP snooping option-82 with data insertion see the "DHCP Snooping Option-82 Data Insertion" section on page 37-3.

For information about the data inserted in the RADIUS attributes suboption, see RFC 4014, "Remote Authentication Dial-In User Service (RADIUS) Attributes Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Information Option."

Supported Topologies

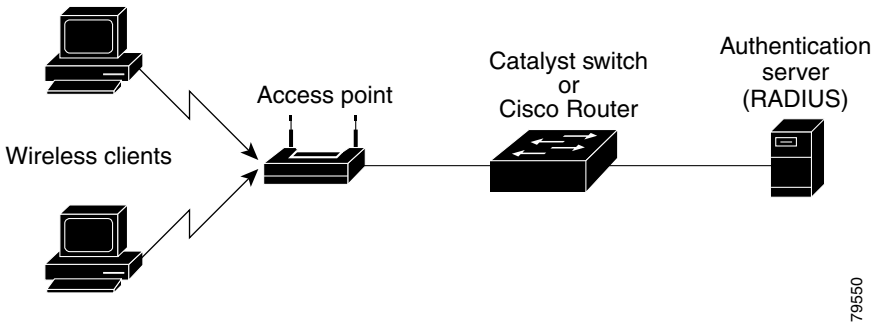
The IEEE 802.1X port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration (see Figure 44-1 on page 44-2), only one client can be connected to the 802.1X-enabled router port. The router detects the client when the port link state changes to the up state. If a client leaves or is replaced with another client, the router changes the port link state to down, and the port returns to the unauthorized state.

Figure 44-3 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 router 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 router.

Figure 44-3 Wireless LAN Example



Default IEEE 802.1X Port-Based Authentication Configuration

Table 44-1 shows the default IEEE 802.1X configuration.

Table 44-1 Default 802.1X Configuration

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled
RADIUS server IP address	None specified
RADIUS server User Datagram Protocol (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 router remains in the quiet state following a failed authentication exchange with the client)
Retransmission time	30 seconds (number of seconds that the router 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 router 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 router 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 router waits for a reply before retransmitting the response to the server)

IEEE 802.1X Port-Based Authentication Guidelines and Restrictions

When configuring IEEE 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 IEEE 802.1X Port-Based Authentication

These sections describe how to configure IEEE 802.1X port-based authentication:

Enabling IEEE 802.1X Port-Based Authentication

To enable IEEE 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 in global configuration mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# aaa new-model`	Enables AAA.
Step 3	`Router(config)# aaa authentication dot1x {default} method1 [method2...]`	Creates an 802.1X port-based authentication method list.
Step 4	`Router(config)# dot1x system-auth-control`	Globally enables 802.1X port-based authentication.
Step 5	`Router(config)# interface type slot/port`	Enters interface configuration mode and specifies the interface to be enabled for 802.1X port-based authentication. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 6	`Router(config-if)# dot1x port-control auto`	Enables 802.1X port-based authentication on the interface.
Step 7	`Router(config)# end`	Returns to privileged EXEC mode.
Step 8	`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**.

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 router 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 Router-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 in global configuration mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# ip radius source-interface` *interface_name*	Specifies that the RADIUS packets have the IP address of the indicated interface.
Step 3	`Router(config)# radius-server host` {*hostname* | *ip_address*}	Configures the RADIUS server host name or IP address on the router. If you want to use multiple RADIUS servers, reenter this command.
Step 4	`Router(config)# radius-server key` *string*	Configures the authorization and encryption key used between the router and the RADIUS daemon running on the RADIUS server.
Step 5	`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 router 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 and the *Cisco IOS Security Command Reference*, Release 12.2 at these URLs:

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

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

Note You also need to configure some settings on the RADIUS server. These settings include the IP address of the router and the key string to be shared by both the server and the router. For more information, refer to the RADIUS server documentation.

This example shows how to configure the RADIUS server parameters on the router:

```
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 IEEE 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 44-11.

To enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Selects an interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 2	`Router(config-if)# dot1x reauthentication`	Enables periodic reauthentication of the client, which is disabled by default.
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 router only if periodic reauthentication is enabled.
Step 4	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Router# show dot1x all`	Verifies your entries.

This example shows how to enable periodic reauthentication, set the number of seconds between reauthentication attempts to 4000, then verify the entries:

```
Router(config)# interface gigabitethernet 4/1
Router(config-if)# dot1x reauthentication
Router(config-if)# dot1x timeout reauth-period 4000
Router(config-if)# end
Router# show dot1x all
```

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 slot/port`	Manually reauthenticates the client connected to a port. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 2	`Router# show dot1x all`	Verifies your entries.

This example shows how to manually reauthenticate the client connected to Fast Ethernet port 5/1 then verify the entries:

```
Router# dot1x re-authenticate interface fastethernet 5/1
Starting reauthentication on FastEthernet 5/1
Router# show dot1x all
```

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 in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# dot1x initialize interface type slot/port`	Initializes the authentication for the client connected to a port. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 2	`Router# show dot1x all`	Verifies your entries.

This example shows how to initialize the authentication for the client connected to Fast Ethernet port 5/1 then verify the entries:

```
Router# dot1x initialize interface fastethernet 5/1
Starting reauthentication on FastEthernet 5/1
Router# show dot1x all
```

Changing the Quiet Period

When the router cannot authenticate the client, the router 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 in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port`	Selects an interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 2	`Router(config-if)# dot1x timeout quiet-period seconds`	Sets the number of seconds that the router remains in the quiet state following a failed authentication exchange with the client. The range is 0 to 65535 seconds; the default is 60.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

This example shows how to set the quiet time on the router to 30 seconds then verify the entries:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x timeout quiet-period 30
Router(config-if)# end
Router# show dot1x all
```

Changing the Router-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the router with an EAP-response/identity frame. If the router 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 router waits for client notification, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type slot/port*	Selects an interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 2	`Router(config-if)# dot1x timeout tx-period` *seconds*	Sets the number of seconds that the router 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.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

This example shows how to set 60 as the number of seconds that the router waits for a response to an EAP-request/identity frame from the client before retransmitting the request, then verify the entries:

```
Router(config)# interface fastethernet 4/1
Router(config-if)# dot1x timeout tx-period 60
Router(config-if)# end
Router# show don1x all
```

Setting the Router-to-Client Retransmission Time for EAP-Request Frames

The client notifies the router that it received the EAP-request frame. If the router does not receive this notification, the router waits a set period of time, and then retransmits the frame. You may set the amount of time that the router waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the router-to-client retransmission time for the EAP-request frames, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type slot/port*	Selects an interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 2	`Router(config-if)# dot1x timeout supp-timeout` *seconds*	Sets the router-to-client retransmission time for the EAP-request frame.

	Command	Purpose
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

This example shows how to set the router-to-client retransmission time for the EAP-request frame to 25 seconds, then verify the entries:

```
Router(config)# interface fastethernet 4/1
Router(config-if)# dot1x timeout supp-timeout 25
Router(config-if)# end
Router# show dot1x all
```

Setting the Router-to-Authentication-Server Retransmission Time for Layer 4 Packets

The authentication server notifies the router each time it receives a Layer 4 packet. If the router does not receive a notification after sending a packet, the router waits a set period of time and then retransmits the packet. You may set the amount of time that the router 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 router to the authentication server, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type slot/port*	Selects an interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 2	`Router(config-if)# dot1x timeout server-timeout` *seconds*	Sets the router-to-authentication-server retransmission time for Layer 4 packets.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

This example shows how to set the router-to-authentication-server retransmission time for Layer 4 packets to 25 seconds, then verify the entries:

```
Router(config)# interface gigabitethernet 5/2
Router(config-if)# dot1x timeout server-timeout 25
Router(config-if)# end
Router# show dot1x all
```

Setting the Router-to-Client Frame Retransmission Number

In addition to changing the router-to-client retransmission time, you can change the number of times that the router 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 router-to-client frame retransmission number, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type slot/port*	Selects an interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
Step 2	`Router(config-if)# dot1x max-req` *count*	Sets the number of times that the router sends an EAP-request/identity frame to the client before restarting the authentication process. The range is 1 to 10; the default is 2.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

This example shows how to set 5 as the number of times that the router sends an EAP-request/identity request before restarting the authentication process, then verify the entries:

```
Router(config)# interface fastethernet 4/1
Router(config-if)# dot1x max-req 5
Router(config-if)# end
Router# show dot1x all
```

Enabling Multiple Hosts

You can attach multiple hosts to a single IEEE 802.1X-enabled port as shown in Figure 44-3 on page 44-5. 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 in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type slot/port*	Selects an interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
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.
Step 3	`Router(config-if)# end`	Returns to privileged EXEC mode.
Step 4	`Router# show dot1x all`	Verifies your entries.

This example shows how to enable 802.1X on Fast Ethernet interface 5/1, allow multiple hosts, then verify the entries:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x host-mode multi-host
Router(config-if)# end
Router# show dot1x all
```

Resetting the IEEE 802.1X Configuration to the Default Values

To reset the IEEE 802.1X configuration to the default values, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type slot/port*	Selects an interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
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.

Displaying IEEE 802.1X Status

To display global IEEE 802.1X administrative and operational status for the router, 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 keywords and arguments in these commands, refer to the *Cisco IOS Security Command Reference, Release 12.2 SR*.

CHAPTER 45

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 Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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 45-6 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 45-4.

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 45-1 shows the default port security configuration for an interface.

Table 45-1 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 *Cisco 7600 Series Router Cisco IOS Command Reference*, 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.

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 45-7).

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`	Enables port security on the trunk.
	`Router(config-if)# no switchport port-security`	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`	Enables port security on the port.
	`Router(config-if)# no switchport port-security`	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}`	(Optional) Sets the violation mode and the action to be taken when a security violation is detected.
	`Router(config-if)# no switchport port-security violation`	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*`]`	Configures a static MAC address as secure on the port. **Note** Per-VLAN configuration is supported only on trunks.
	`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 45-8).
- 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.
  - 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 45-10
- Configuring Secure MAC Address Aging Time on a Port, page 45-11

Configuring the Secure MAC Address Aging Type on a Port

With a PFC3, you can configure the secure MAC address aging type on a port.

To configure the secure MAC address aging type on a port, perform this task:

	Command	Purpose
Step 1	`Router(config)# interface` *type*[1] *slot/port*	Selects the LAN port to configure.
Step 2	`Router(config-if)# switchport port-security aging type {absolute | inactivity}`	Configures the secure MAC address aging type on the port (default is absolute).
	`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 router 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 46

Configuring UDLD

This chapter describes how to configure the UniDirectional Link Detection (UDLD) protocol on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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.

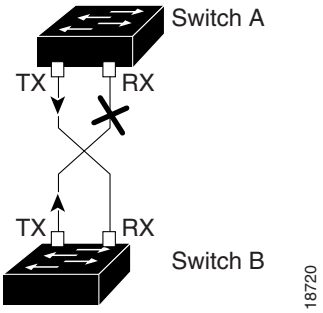
A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does 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 Cisco 7600 series router 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 46-1 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 46-1 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.

Default UDLD Configuration

Table 46-1 shows the default UDLD configuration.

Table 46-1 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

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 47

Configuring NetFlow and NDE

This chapter describes how to configure NetFlow statistics collection and NetFlow Data Export (NDE) on the Cisco 7600 series routers.

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Cisco IOS NetFlow Command Reference* at this URL:

  http://www.cisco.com/en/US/docs/ios/netflow/command/reference/nf_book.html
- NetFlow version 9 is supported—See this publication:

  Cisco IOS NetFlow Configuration Guide
- NetFlow multicast support includes the NetFlow v9 export format feature. See this publication:

  Cisco IOS NetFlow Configuration Guide

  You do not need to configure multicast fast switching or multicast distributed fast switching (MDFS); multicast CEF switching is supported.

This chapter consists of these sections:

Understanding How NetFlow and NDE Work

These sections describe how NetFlow and NDE work:

NetFlow and NDE Overview

NetFlow collects statistics about traffic that flows through the router. NetFlow Data Export (NDE) enables you to export those statistics to an external data collector for analysis.

NetFlow and NDE are either enabled globally or enabled on individual interfaces, depending on which software release you are using:

- In Cisco IOS Release 12.2SRA and earlier releases, NetFlow is enabled globally, which means that statistics are gathered for all interfaces on the router.
- In Cisco IOS Release 12.2SRB, you can enable NetFlow on individual interfaces for IPv4 traffic on Layer 3 interfaces. NetFlow for IPv6 traffic continues to operate in global mode. For more information about this feature, see the "Per-Interface NetFlow and NDE" section on page 47-10.

Note Beginning in Release 12.2SRB, global-mode NetFlow for IPv4 traffic is no longer the default. To achieve the same global-mode functionality as before, you must now manually enable NetFlow on each Layer 3 interface where you want to capture statistics for IPv4 traffic flows.

You can collect statistics for both routed and bridged traffic. Note, however, that the PFC3A collects statistics only for routed traffic.

You can configure two external data collector addresses, which improves the probability of receiving complete NetFlow data by providing redundant data streams with a PFC3.

To reduce the volume of statistics collected, use:

- NetFlow Sampling, which reduces the number of statistics collected
- NetFlow aggregation, which merges collected statistics

NetFlow and NDE on the MSFC

The NetFlow cache on the MSFC captures statistics for flows routed in software. The MSFC supports NetFlow aggregation for traffic routed in software. For more information, see the Cisco IOS NetFlow Configuration Guide.

The MSFC supports NetFlow ToS-based router aggregation, For more information, see the Cisco IOS NetFlow Configuration Guide.

NetFlow and NDE on the PFC

The NetFlow cache on the PFC captures statistics for flows routed in hardware. The PFC supports sampled NetFlow and NetFlow aggregation for traffic routed in hardware. The PFC does not support NetFlow ToS-Based Router Aggregation.

These sections describe NetFlow and NDE on the PFC in more detail:

Flow Masks

This section describes the flow masks that are used to create NetFlow entries. Two sets of flow masks are available: for Release 12.2SRA and Release 12.2SRB. NetFlow applies the selected flow mask to all statistics gathered on the router.

Release 12.2SRA

Cisco IOS Release 12.2SRA uses the following types of flow masks to create NetFlow entries:

- source-only—A less-specific flow mask. The PFC maintains one entry for each source IP address. All flows from a given source IP address use this entry.
- destination—A less-specific flow mask. The PFC maintains one entry for each destination IP address. All flows to a given destination IP address use this entry.
- destination-source—A more-specific flow mask. The PFC maintains one entry for each source and destination IP address pair. All flows between same source and destination IP addresses use 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 PFC creates and maintains a separate cache entry for each IP flow. A full entry includes the source IP address, destination IP address, protocol, and protocol interfaces.
- full-interface—The most-specific flow mask. Adds the source VLAN SNMP ifIndex to the information in the full-flow mask.

Release 12.2SRB

Cisco IOS Release 12.2SRB use the following flow masks:

- destination-source-interface—A more-specific flow mask. Adds the source VLAN SNMP ifIndex to the information in the destination-source flow mask.
- full-interface—The most-specific flow mask. Adds the source VLAN SNMP ifIndex to the information in the full-flow mask.

Other flow masks are handled as follows in order to accommodate per-interface mode for IPv4 traffic:

- Source-only, destination, and destination-source flow masks are treated as destination-source-interface.
- Full flow masks are treated as full-interface.

NDE Versions

NDE on the PFC supports NDE versions 5, 7, and 9 for the statistics captured on the PFC.
For information about NetFlow version 9, see the publication at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123_1/nfv9expf.htm

The following tables describe the supported NDE fields:

- Table 47-1—Version 5 header format
- Table 47-2—Version 7 header format
- Table 47-3—Version 5 flow record format
- Table 47-4—Version 7 flow record format

Table 47-1 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 47-2 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

Note Some fields in the flow records might not have values, depending on the current flow mask. Unsupported fields contain a zero (0).

Table 47-3 NDE Version 5 Flow Record Format

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field					
			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	—	—	—	—	—	—	—	—
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. With PFC3CXL, PFC3C, PFC3BXL, or PFC3B, for ICMP traffic, contains the ICMP code and type values.
5. Always zero for hardware-switched flows.

Table 47-4 NDE Version 7 Flow Record Format

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field					
			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	—	—	—	—	—	—	—	—
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. With PFC3CXL, PFC3C, PFC3BXL, or PFC3B, for ICMP traffic, contains the ICMP code and type values.
5. Always zero for hardware-switched flows.

MLS Cache Entries

NetFlow captures traffic statistics in the NetFlow cache on the PFC.

NetFlow maintains traffic statistics for each active flow in the NetFlow cache 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 cache that have expired since the last export. Flow entries in the NetFlow cache expire and are flushed from the NetFlow cache 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 1920 seconds [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 47-26 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 7600 series router 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 7600 series routers:

- NetFlow Traffic Sampling, page 47-7
- NetFlow Flow Sampling, page 47-8

NetFlow Traffic Sampling

NetFlow traffic sampling provides NetFlow data for a subset of traffic forwarded by a Cisco router 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. NetFlow traffic sampling is used on platforms that perform software-based NetFlow accounting, such as Cisco 7200 series routers and Cisco 7600 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 7600 series routers, 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 7600 series routers 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 47-5 lists the sampling rates and export intervals for time-based NetFlow flow sampling.

Table 47-5 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

NetFlow Aggregation

For information about NetFlow aggregation support on the PFC and DFCs, see the “NetFlow Aggregation” section of the document at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt2/xcfnfov.htm

Note

- In Release 12.2SRB, you must enable NetFlow on individual interfaces in order to enable the hardware flow cache to be populated. When enabled, the cache is populated with flows only from those interfaces where NetFlow is enabled.
- In Release 12.2SRA, configuring an aggregation scheme allows the hardware flow cache to be populated. The cache is globally populated with information for all Layer 3 interfaces.
- Configuring NetFlow aggregation for the MSFC also configures it for the PFC and DFCs. (See Configuring NetFlow Aggregation for Flows on the MSFC, page 47-22, for a pointer to configuration instructions).
- NetFlow aggregation uses NDE version 8.

Per-Interface NetFlow and NDE

In Cisco IOS Release 12.2SRB and later releases, the per-interface NetFlow and NDE feature allows you to enable NetFlow on individual interfaces in order to gather and export statistics for IPv4 traffic flows on those interfaces. Previously, when you enabled NetFlow, statistics were gathered for all of the interfaces on the router (global mode).

If you upgrade to Release 12.2SRB (per-interface mode) from an earlier release (global mode), you must issue the **ip flow ingress** command on individual interfaces to activate NetFlow. The upgrade process automatically converts existing global-mode flowmasks into the corresponding per-interface type (source, destination, and destination-source become destination-source-interface, and full becomes full-interface).

If you downgrade from Release 12.2SRB to an earlier release, NetFlow resumes global-mode operation (gathering statistics for all router interfaces) and the 12.2SRB flowmasks remain in effect.

The per-interface NetFlow feature improves NetFlow table utilization and performance as follows:

- Provides more room in the NetFlow table for flows that are of interest. With per-interface NetFlow, table entries are created only for those interfaces where NetFlow is enabled. This reduces the number of unwanted entries in the table, leaving more room for those flows that you are interested in. Previously, entries were created for all router interfaces.

  Creating table entries only for interfaces where NetFlow is enabled improves performance because:

  - The NetFlow table is shared by all flow-based features (NetFlow, QoS, multicast, and so on).
  - If the NetFlow table gets too full, NetFlow shortcuts might not being installed, which can result in flow statistics (and accounting information) being lost.
- Helps to ensure that the export of NDE records to the Netflow Data Collector (NFC) at a high rate of speed does not overwhelm the NFC and cause important accounting data to be lost. Since statistics are gathered and exported for specific interfaces only, the number of NDE records sent to the NFC is more manageable.
- Helps to ensure that there is less unintentional conflict between NDE and other features.

The following sections provide information about per-interface NetFlow and NDE and some additional NetFlow and NDE related features that are being introduced in Release 12.2SRB:

Per-Interface NetFlow and NDE Usage Guidelines and Limitations

Consider the following usage guidelines and limitations when you configure per-interface NetFlow and NDE on the Cisco 7600 router:

- Supported in Cisco IOS Release 12.2SRB and later releases.
- Supported on RSP720, Sup720, and Sup32.
- Supported for IPv4 unicast and multicast traffic on Layer 3 interfaces.
  For IPv6 flows, NetFlow and NDE operate in global mode, not per-interface mode.
- When you enable NetFlow and NDE for Layer 2 (bridged) flows, the features are also automatically enabled for Layer 3 (routed) flows on the interface. To disable NetFlow and NDE for the interface, you must disable the feature for both the Layer 2 and Layer 3 flows. Use the **no ip flow ingress layer2-switched** command to disable L2 flows and **no ip flow ingress** to disable L3 flows.
- Do not configure per-interface NetFlow on any interface where QoS micro-policing is used.
- Beginning in Release 12.2SRB, the router supports both NDE flow mask and QoS flow mask; however, you cannot configure both types of flow masks on the same interface.
- When NDE and multicast non-RPF are both enabled, NDE has the potential to lose statistics. This potential loss occurs because NetFlow and NDE are enabled globally for multicast flows, which means that the NetFlow table could overflow.
- The following limitations apply to flow masks in per-interface mode:
  - You cannot configure different flow mask types for individual interfaces. Only a single flow mask type is supported for all interfaces configured for per-interface NetFlow or NDE.
  - The same flow mask is used for both routed (L3) and bridged (L2) NetFlow entries for NDE.
  - All source and destination flow masks are treated as destination-source-interface and both of the full masks are treated as full-interface. See the "Flow Masks" section on page 47-3 for a description of flow mask types.
- All of guidelines and limitations in the "NetFlow and NDE Configuration Guidelines and Restrictions" section on page 47-14 apply.

Configuring Per-Interface NetFlow and NDE

Following is a summary of the steps you must perform to configure per-interface NetFlow and NDE on Cisco 7600 routers. Detailed procedures for each step are provided in the sections later in this chapter.

1. If you plan to export NetFlow statistics, globally enable NDE on the router by issuing the following commands:

```
configure terminal
```

```
ip flow-export destination
ip flow-export version
mls nde sender version
```

2. Enable NetFlow on individual interfaces by issuing the following commands:

```
configure terminal
interface
  ip flow ingress
```

3. (Optional) To configure NetFlow sampling, do the following:
   a. Enable sampled NetFlow globally on the router (**mls sampling**).
   b. Enable sampled NetFlow on individual interfaces (**mls netflow sampling**).
4. Verify the NDE configuration to ensure that it does not conflict with other features such as QoS or multicast. Use the **show ip interface** command to verify the configuration (see the "Verifying Per-Interface NetFlow and NDE" section on page 47-12).

Verifying Per-Interface NetFlow and NDE

To verify whether per-interface NetFlow and NDE are properly configured, use the **show ip interface** command (as shown here). In the command output, fields showing NetFlow and NDE configuration information are shown in boldface.

```
Router# show ip interface gig2/9
GigabitEthernet2/9 is up, line protocol is up
  Internet address is 10.0.0.1/8
  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: 224.0.0.5 224.0.0.2 224.0.0.6
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  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 Flow switching is disabled
  IP CEF switching is enabled
  IP CEF switching turbo vector
  IP Null turbo vector
  Associated unicast routing topologies:
    Topology "base", operation state is UP
  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
```

```
BGP Policy Mapping is disabled
Input features: Ingress-NetFlow
Output features: Post-Ingress-NetFlow, HW Shortcut Installation
Post encapsulation features: HW Shortcut Installation
Sampled Netflow is disabled
IP Routed Flow creation is enabled in netflow table
IP Bridged Flow creation is disabled in netflow table
WCCP Redirect outbound is disabled
WCCP Redirect inbound is disabled
WCCP Redirect exclude is disabled
IP multicast multilayer switching is disabled
```

NetFlow v9 for IPv6

Cisco IOS Release 12.2SRB introduces support for NetFlow version 9 for IPv6. For information about how to configure this feature on the Cisco 7600 router, see its feature module description in the new feature documentation for Release 12.2SRB at the following URL:

http://www.cisco.com/en/US/products/ps6922/products_feature_guides_list.html

NDE on VRF Interfaces

Cisco IOS Release 12.2SRB introduces support for NDE on VRF interfaces. This new feature enables the Cisco 7600 router to capture and export NetFlow statistics for IPv4 packets in an MPLS Virtual Private Network (VPN). In this scenario, the router is functioning as provider edge (PE) router at the edge of an MPLS network.

For additional information about NDE on VRF interfaces, see its feature module description in the new feature documentation for Release 12.2SRB at the following URL:

http://www.cisco.com/en/US/products/ps6922/products_feature_guides_list.html

Default NetFlow and NDE Configuration

Table 47-6 shows the default NetFlow and NDE configuration.

Table 47-6 Default NetFlow and NDE Configuration

Feature	Default Value
NetFlow	Disabled. 12.2SRB—Per-interface mode for IPv4 unicast (global mode for all else). 12.2SRA—Global mode.
NDE	Disabled.
NDE on VRF interfaces	12.2SRB—Disabled. 12.2SRA—Not available.

Table 47-6 Default NetFlow and NDE Configuration (continued)

Feature	Default Value
NetFlow and NDE of ingress bridged IP traffic	Disabled.
NDE source addresses	None.
NDE data collector address and UDP port	None.
NDE filters	None.
NetFlow Mask	None.
NetFlow Sampling	Disabled.
NetFlow Aggregation	Disabled.
Populating additional NDE fields	Enabled.

NetFlow and NDE Configuration Guidelines and Restrictions

When configuring NetFlow and NDE, follow these guidelines and restrictions:

NetFlow and NDE support IP multicast traffic only with NetFlow version 9. With other NetFlow versions, you can display NetFlow statistics for IP multicast traffic with the **show mls ip multicast** command.

- All PFCs (except the PFC3A) support NetFlow and NDE for bridged IP traffic.
- NDE does not support Internetwork Packet Exchange (IPX) traffic.
- The Policy Feature Card 3 (PFC3) does not use the NetFlow table for Layer 3 switching in hardware.
- If the NetFlow table utilization exceeds these recommended utilization levels, there is an increased probability that there will be insufficient room to store statistics:

PFC	Recommended NetFlow Table Utilization	Total NetFlow Table Capacity
PFC3CXL PFC3BXL	235,520 (230K) entries	262,144 entries
PFC3C PFC3B	117,760 (115K) entries	131,072 entries
PFC3A	65,536 (64K) entries	131,072 entries

- No statistics are available for flows that are switched when the NetFlow table is full.
- The Cisco 7600 series router uses the Netflow table to maintain information about flow-based features. Normally, the Feature Manager creates a Netflow table entry for a flow-based feature only on the line card where the flow ingresses. However, because TCP intercept is a global feature, the router creates an entry for each TCP intercept flow on each of the installed PFCs and DFCs, not just the ingress PFC or DFC. This means that the PFC or DFC where the TCP intercept flow ingresses will have a non-zero packet count, but the other PFC and the DFCs will have a count of zero packets for the flow. [CSCek47971]
- The following IPv4 Netflow and NDE options are not available for IPv6 flows: [CSCek55571]
  - Aggregation support (**ip flow-aggregation cache** command)
  - Export of Layer 2 switched IPv6 flows

- Netflow and NDE sampling
- NDE filter support

Multicast NDE Configuration Guidelines

Observe the following guidelines when you configure multicast NDE on the Cisco 7600:

- In Release 12.2SRB and later releases, multicast NDE and QoS microflow policing cannot both be configured on the same interface. However, the features can be configured on different interfaces.
- To configure multicast NDE, issue both the **ip flow ingress** and **ip multicast netflow ingress** commands. Note that the **ip multicast netflow ingress** command is enabled by default.

Release 12.2SRB and Later Releases

Beginning in Release 12.2SRB, for IPv4 flows, the router supports per-interface mode NetFlow and NDE only. For IPv6 flows, NetFlow and NDE continue to operate in global mode.

See the "Per-Interface NetFlow and NDE" section on page 47-10 for information about per-interface NetFlow and NDE and its usage guidelines and restrictions.

Configuring NetFlow and NDE

These sections describe how to configure NetFlow and NDE:

Note

- You must enable NetFlow on the MSFC Layer 3 interfaces to support NDE on the PFC and NDE on the MSFC.
- You must enable NDE on the MSFC to support NDE on the PFC.
- When you configure NAT and NDE on an interface, the PFC sends all traffic in fragmented packets to the MSFC to be processed in software. (CSCdz51590)

Configuring NetFlow and NDE for Flows on the PFC

These sections describe how to configure NetFlow and NDE for flows on the PFC:

Configuring NetFlow for Flows on the PFC

These sections describe how to configure NetFlow statistics collection for flows on the PFC:

Enabling NetFlow on the PFC (Release 12.2SRA)

To enable NetFlow statistics collection for flows on the PFC in Release 12.2SRA, perform this task. For information about enabling NetFlow in Release 12.2SRB and later releases, see the following section.

Command	Purpose
`Router(config)# mls netflow`	Enables NetFlow on the PFC.
`Router(config)# no mls netflow`	Disables NetFlow on the PFC.

This example shows how to enable NetFlow statistics collection:

```
Router(config)# mls netflow
```

Enabling Per-Interface NetFlow (Release 12.2SRB and Later)

To enable NetFlow statistics collection for flows on the PFC in Release 12.2SRB and later releases, perform this task. See the "Per-Interface NetFlow and NDE" section on page 47-10 for information about how the router operates in NetFlow and NDE per-interface mode. For detailed information about command syntax, see the command reference documents listed at the beginning of this chapter.

Command	Purpose
`Router(config)# mls flow ip`	Configures the flow mask to use for NetFlow entries.
`Router(config)# interface` *interface*	Selects the interface to enable NetFlow on.
`Router(config-if)# [no] ip flow ingress`	Enables NetFlow on a Layer 3 interface. Issue the command on each interface where you want to enable the feature. Use the **no** form of the command to disable NetFlow and NDE on the interface.
`Router(config-if)# exit`	Exits interface configuration mode.
`Router(config)# mls nde sender` `Router(config)# ip flow-export destination {`*hostname* `|` *ip-address*`} udp-port`	(Optional) Enables NDE. Issue these commands if you plan to export NetFlow statistics. Specifies an external host (name or IP address) to send NetFlow statistics to and the port
`Router(config)# mls nde sender`	(Optional) Enables NDE. Use this command if you plan to export NetFlow statistics.

Command	Purpose
`Router(config)# ip flow-export destination {hostname | ip-address} udp-port`	(Optional) Specifies the host name or IP address of the external host to export NetFlow statistics to and specifies the port to send the statistics to.
`Router(config)# show ip interface interface`	Displays the configuration of the specified interface. Examine the configuration to ensure that the NDE configuration does not conflict with other features such as QoS or multicast (see "Verifying Per-Interface NetFlow and NDE").

Configuring NetFlow Flow Sampling

These sections describe how to configure sampled NetFlow on the PFC:

- Configuring NetFlow Flow Sampling Globally (Release 12.2SRB and Release 12.2SRA), page 47-17
- Configuring Per-Interface Mode NetFlow Flow Sampling (Release 12.2SRB), page 47-17
- Configuring NetFlow Flow Sampling on a Layer 3 Interface (Release 12.2SRA), page 47-18

Note NDE on the MSFC does not support NetFlow Flow Sampling.

Configuring NetFlow Flow Sampling Globally (Release 12.2SRB and Release 12.2SRA)

To configure sampled NetFlow globally in Release 12.2SRB and Release 12.2SRA, perform this task:

	Command	Purpose
Step 1	`Router(config)# mls sampling {time-based rate | packet-based rate [interval]}`	Enables sampled NetFlow and configures the rate. For packet-based sampling, optionally configures the export interval.
	`Router(config)# no mls sampling`	Clears the sampled NetFlow configuration.
Step 2	`Router(config)# end`	Exits configuration mode.

When you configure sampled NetFlow globally, note the following information:

- The valid values for *rate* are 64, 128, 256, 512, 1024, 2048, 4096, and 8192.
- The valid values for the packet-based export *interval* are from 8,000 through 16,000.
- To export any data in Release 12.2SRA, you must also configure sampled NetFlow on a Layer 3 interface.

See the "NetFlow Sampling" section on page 47-7 for more information.

Configuring Per-Interface Mode NetFlow Flow Sampling (Release 12.2SRB)

In Release 12.2SRB and later releases, you must enable sampled NetFlow globally and on individual interfaces (as shown in the following example).

In the example, the **mls sampling** command enables sampled NetFlow globally and the **mls netflow sampling** command enables sampled NetFlow on the interface (in this example, Fast Ethernet port 5/12).

```
Router# configure terminal
Router(config)# mls sampling packet-based 64
Router(config)# interface fastethernet 5/12
```

```
Router(config-if)# mls netflow sampling
Router(config)# end
Router#
```

Configuring NetFlow Flow Sampling on a Layer 3 Interface (Release 12.2SRA)

In Release 12.2SRA, with the full-interface or destination-source-interface flow masks, you can enable or disable sampled NetFlow on individual Layer 3 interfaces. With all other flow masks, sampled NetFlow is enabled or disabled globally.

To configure sampled NetFlow on a Layer 3 interface in Release 12.2SRA, make sure that sampled NetFlow is enabled globally and perform this task:

	Command	Purpose
Step 1	`Router(config)# interface {vlan vlan_ID | type slot/port}`	Specifies the Layer 3 interface to configure. **Note** The Layer 3 interface must be configured with an IP address.
Step 2	`Router(config-if)# mls netflow sampling`	Enables sampled NetFlow on the Layer 3 interface.
	`Router(config-if)# no mls netflow sampling`	Disables sampled NetFlow on the Layer 3 interface.
Step 3	`Router(config)# end`	Exits configuration mode.

This example shows how to enable sampled NetFlow 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 NetFlow Aggregation for Flows on the PFC

NetFlow aggregation is configured automatically for flows on the PFC and DFCs when you configure NetFlow aggregation for the MSFC (see the "Configuring NetFlow Aggregation for Flows on the MSFC" section on page 47-22 for a pointer to configuration instructions).

To display NetFlow aggregation cache information for the PFC or DFCs, perform this task:

Command	Purpose
`Router # show ip cache flow aggregation {as | destination-prefix | prefix | protocol-port | source-prefix) module slot_num`	Displays the NetFlow aggregation cache information.
`Router # show mls netflow aggregation flowmask`	Displays the NetFlow aggregation flow mask information. This command is applicable only to Release 12.2SRA; the command is not applicable in Release 12.2SRB.

Note The PFC and DFCs do not support NetFlow ToS-based router aggregation.

This example shows how to display the NetFlow aggregation cache information:

```
Router# show ip cache flow aggregation destination-prefix module 1
IPFLOW_DST_PREFIX_AGGREGATION records and statistics for module :1
IP Flow Switching Cache, 278544 bytes
```

```
2 active, 4094 inactive, 6 added
236 ager polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
Dst If Dst Prefix Msk AS Flows Pkts B/Pk Active
Gi7/9 9.1.0.0 /16 0 3003 12M 64 1699.8
Gi7/10 11.1.0.0 /16 0 3000 9873K 64 1699.8
Router#
```

This example displays the NetFlow aggregation flow mask information (Release 12.2SRA only):

```
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#
```

Setting the Minimum IP MLS Flow Mask (Release 12.2SRA Only)

You can set the minimum specificity of the flow mask for the NetFlow cache on the PFC (see the "Flow Masks" section on page 47-3). The actual flow mask that is used will have at least the specificity configured by the **mls flow ip** command.

Note The task does not apply to Release 12.2SRB, which supports only the interface-destination-source and interface-full flow masks.

To set the minimum IP 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 flow mask for the protocol.
`Router(config)# no mls flow ip`	Reverts to the default IP flow mask (null).

This example shows how to set the minimum IP flow mask:

```
Router(config)# mls flow ip destination
```

To display the IP 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 cache 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 and NDE Configuration Guidelines and Restrictions" section on page 47-14), only adjacency statistics might be available for some flows.

To keep the NetFlow cache size below the recommended utilization, enable the following parameters when using the **mls aging** command:

- **normal**—Configures the wait before aging out and deleting entries that are not covered by fast or long aging.
- **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 the aging time for deleting entries that are always in use. Long aging is used to prevent counter wraparound, which can cause inaccurate statistics.

A typical cache entry that is removed is the entry for flows to and from a Domain Name Server (DNS) or TFTP server. This entry might not be used again after it is created. The PFC saves space in the NetFlow cache for other data when it detects and ages out these entries.

If you need to enable MLS fast aging time, initially set the value to 128 seconds. If the size of the NetFlow cache continues to grow over the recommended utilization, decrease the setting until the cache size stays below the recommended utilization. If the cache continues to grow over the recommended utilization, decrease the normal MLS aging time.

To configure an 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 an MLS aging time for a NetFlow cache 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 an 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
```

Enabling NDE

For both Release 12.2SRA and Release 12.2 SRB, perform this task to globally enable NDE:

Command	Purpose
`Router(config)# mls nde sender [version {5 | 7}]`	Enables NDE for flows on the PFC and (optionally) configures the NDE version. Specify an NDE version that matches the NetFlow collector that the data is being exported to.
`Router(config)# ip flow-export destination {hostname | ip-address} udp-port`	Identifies the
`Router(config)# no mls nde sender`	Disables NDE for flows on the PFC.
`Router(config)# no mls nde sender version`	Reverts to the default (version 7).

Note

- NDE for the PFC uses the source interface configured for the MSFC (see the "Configuring the MSFC NDE Source Layer 3 Interface" section on page 47-22).
- NetFlow version 9 is supported—See this publication:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123_1/nfv9expf.htm

This example shows how to globally enable NDE for flows on the PFC:

```
Router(config)# mls nde sender
```

This example shows how to globally enable NDE for the PFC and configure NDE version 5:

```
Router(config)# mls nde sender version 5
```

Configuring NetFlow and NDE for Flows on the MSFC

This section supplements the NetFlow procedures at this URL:

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

These sections describe how to configure NDE on the MSFC:

Enabling NetFlow for Flows on the MSFC

In Release 12.2SRB and later releases, NDE is automatically enabled on an interface when you enable NetFlow on the interface (**ip flow ingress**). However, for NDE to work, you must globally enable it and specify a destination to export the statistics to (**mls nde sender** and **ip flow-export destination**).

In Release 12.2SRA, enable NetFlow on the MSFC by performing this task for each Layer 3 interface where you want to enable NDE.

	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` `Router(config-if)# ip route-cache flow`	Enables NetFlow.

Configuring NetFlow Aggregation for Flows on the MSFC

To configure NetFlow aggregation for flows on the MSFC, use the procedures in the section "Configuring an Aggregation Cache" at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt2/xcfnfc.htm#wp1001058

Note

- Configuring NetFlow aggregation for the MSFC automatically configures it for the PFC and DFCs.
- In Release 12.2SRB, you must enable NetFlow on individual interfaces in order to enable the hardware flow cache to be populated. When enabled, the cache is populated with flows only from those interfaces where NetFlow is enabled.
- In Release 12.2SRA, configuring an aggregation scheme allows the hardware flow cache to be populated. The cache is globally populated with information for all L3 interfaces.

To configure NetFlow ToS-based router aggregation for the MSFC, use the procedures at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s15/dtnfltos.htm

Note The PFC and DFCs do not support NetFlow ToS-based router aggregation.

Configuring the MSFC NDE Source Layer 3 Interface

To configure the Layer 3 interface used as the source of the NDE packets containing statistics from the MSFC, 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 MSFC.
`Router(config)# no ip flow-export source`	Clears the NDE source interface configuration.

When configuring the MSFC 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. This hardware supports the Netflow Multiple Export Destinations feature:

- PFC3

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 router 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.

Enabling NetFlow and NDE for Ingress Bridged IP Traffic

All PFCs (except the PFC3A) support NetFlow and NDE for ingress bridged IP traffic. The following sections describe how to enable NetFlow and NDE for ingress bridged IP traffic:

- Enabling NetFlow for Ingress Bridged IP Traffic in VLANs, page 47-24
- Enabling NDE for Ingress Bridged IP Traffic in VLANs, page 47-24

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 47-7).

- For each VLAN where you want to enable NetFlow and NDE for bridged IP traffic, you must create a corresponding VLAN interface, assign an IP address to it, and issue the **no shutdown** command to bring the interface up.
- When you enable NetFlow for bridged IP traffic on a VLAN, export of the bridged traffic is enabled by default as long as NDE is globally enabled.

Enabling NetFlow for Ingress Bridged IP Traffic in VLANs

To enable NetFlow for ingress bridged IP traffic in VLANs, perform this task:

Command	Purpose
`Router(config)# ip flow ingress layer2-switched vlan` *vlan_ID*`[-`*vlan_ID*`] [, `*vlan_ID*`[-`*vlan_ID*`]]`	Enables NetFlow for ingress bridged IP traffic in the specified VLANs. **Note** NetFlow for ingress bridged IP traffic in a VLAN requires that NetFlow on the PFC be enabled with the **mls netflow** command.
`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 NDE for Ingress Bridged IP Traffic in VLANs

To enable 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 47-28.

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 to host 172.20.52.37:

```
Router(config)# mls nde flow include destination 172.20.52.37 255.255.255.255
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 47-28.

Usage Guidelines to Configure Protocol Flow Filter

Follow these restrictions and usage guidelines to configure NetFlow Data Export Filter:

- Only one filter is supported to include or exclude flow export. The flow export configuration is based on source IP, destination IP, source Port, destination port and protocol.
- If you separately configure each filter parameter, the final filter consists of all the configured filter values as shown in the next example:

```
Router(config)#mls nde flow include src-port 100
Router#sh run | I mls nde flow
mls nde flow include protocol tcp src-port 100
Router(config)#mls nde flow include dest-port 200
Router#sh run | I mls nde flow
mls nde flow include protocol tcp src-port 100 dest-port 200
Router#
```

- If you reconfigure a filter with a new value, the old value is overwritten as shown in the next example:

```
Router(config)#mls nde flow include dest-port 200
Router#sh run | I mls nde flow
mls nde flow include dest-port 200
Router(config)#mls nde flow include dest-port 500
Router#sh run | I mls nde flow
mls nde flow include dest-port 500
```

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 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 Cisco 7600 series routers. Policy Feature Card 3 (PFC3) supports ERSPAN (see the "ERSPAN Guidelines and Restrictions" section on page 48-10).

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- Shared port adapter (SPA) 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

SPAN copies traffic from one or more ports, one or more EtherChannels, or one or more VLANs, and sends the monitored traffic to one or more destinations such as a SwitchProbe device or other remote monitoring (RMON) probe.

SPAN does not affect the switching of traffic on sources. You must dedicate the destination for SPAN use. The SPAN-generated copies of traffic compete with user traffic for router resources.

These sections provide an overview of local SPAN, RSPAN, and ERSPAN:

Local SPAN Overview

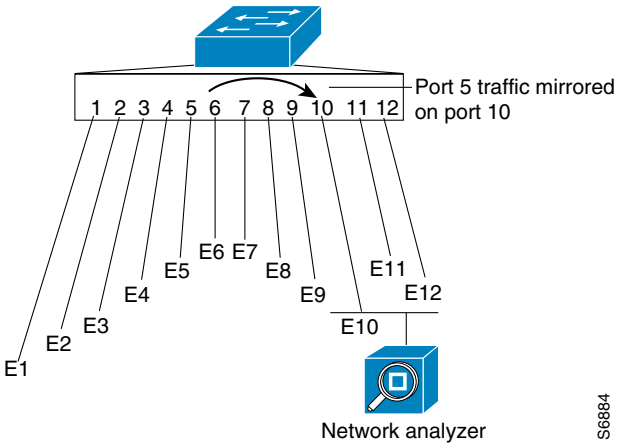
A local SPAN session is an association of source ports and source VLANs with one or more destinations. You configure a local SPAN session on a single router. Local SPAN does not have separate source and destination sessions.

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 generic routing encapsulation (GRE)-encapsulated traffic from source ports.

Each local SPAN session can have either ports or VLANs as sources, but not both.

Local SPAN copies traffic from one or more source ports in any VLAN or from one or more VLANs to a destination for analysis (see Figure 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 destinations on different routers. This provides remote monitoring of multiple routers across your network (see Figure 48-2). RSPAN uses a Layer 2 VLAN to carry SPAN traffic between routers.

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 routers. To configure an RSPAN source session on one router, you associate a set of source ports or VLANs with an RSPAN VLAN. To configure an RSPAN destination session on another router, you associate the destinations with the RSPAN VLAN.

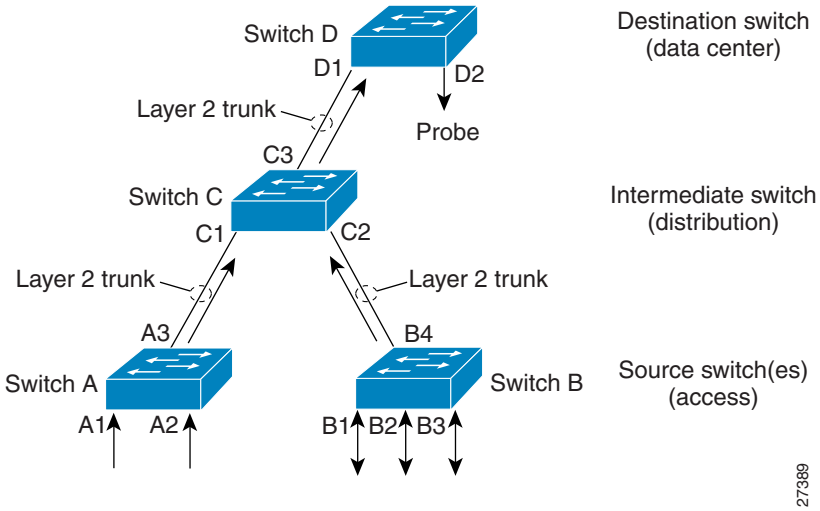
The traffic for each RSPAN session is carried as Layer 2 nonroutable traffic over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating routers. All participating routers must be trunk-connected at Layer 2.

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 destinations on different routers. This provides remote monitoring of multiple routers across your network (see Figure 48-3). ERSPAN uses a GRE tunnel to carry traffic between routers.

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 routers.

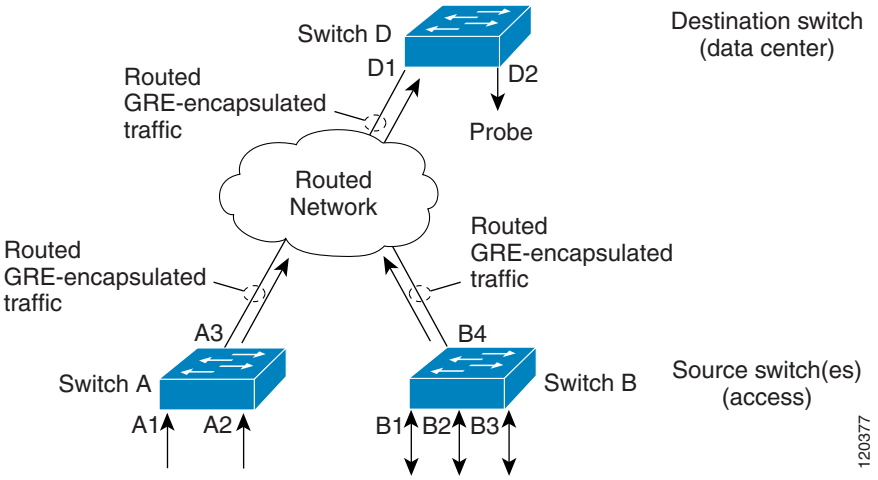
To configure an ERSPAN source session on one router, you associate a set of source ports or VLANs with a destination IP address, ERSPAN ID number, and optionally with a VPN routing and forwarding (VRF) name. To configure an ERSPAN destination session on another router, you associate the destination ports 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 destinations.

Figure 48-3 ERSPAN Configuration



Understanding the Traffic Monitored at SPAN Sources

These sections describe the traffic that local SPAN, RSPAN, and ERSPAN sources can monitor:

- Monitored Traffic Direction, page 48-4
- Monitored Traffic Type, 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 Type

By default, local SPAN and ERSPAN monitor all traffic, including multicast and bridge protocol data unit (BPDU) frames. RSPAN does not support BPDU monitoring.

Duplicate Traffic

In some configurations, SPAN sends multiple copies of the same source traffic to the destination. For example, in a configuration with a bidirectional SPAN session (both ingress and egress) for two SPAN sources, called s1 and s2, to a SPAN destination, called d1, if a packet enters the router 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 and EtherChannels

A source port or EtherChannel is a port or EtherChannel monitored for traffic analysis. You can configure both Layer 2 and Layer 3 ports as SPAN sources. SPAN can monitor one or more source ports or EtherChannels in a single SPAN session. You can configure ports or EtherChannels in any VLAN as SPAN sources. Trunk ports or EtherChannels can be configured as sources and mixed with nontrunk sources. 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 and EtherChannels in the source VLANs become sources of SPAN traffic.

Note Layer 3 VLAN interfaces on source VLANs are not sources of SPAN traffic. Traffic that enters a VLAN through a Layer 3 VLAN interface is monitored when it is transmitted from the router through an egress port of EtherChannel that is in the source VLAN.

Local SPAN, RSPAN, and ERSPAN Destinations

A SPAN destination is a Layer 2 or Layer 3 LAN port or, with Release 12.2(33)SRC and later, an Etherchannel, to which local SPAN, RSPAN, or ERSPAN sends traffic for analysis. When you configure a port or EtherChannel as a SPAN destination, it is dedicated for use only by the SPAN feature.

Destination EtherChannels do not support the Port Aggregation Protocol (PAgP) or Link Aggregation Control Protocol (LACP) EtherChannel protocols; only the on mode is supported, with all EtherChannel protocol support disabled.

There is no requirement that the member links of a destination EtherChannel be connected to a device that supports EtherChannels. For example, you can connect the member links to separate network analyzers. See Chapter 12, "Configuring EtherChannels" for more information about EtherChannels.

Destinations, by default, cannot receive any traffic. With Release 12.2(33)SRC and later, you can configure Layer 2 destinations to receive traffic from any attached devices.

Destinations, by default, do not transmit anything except SPAN traffic. Layer 2 destinations that you have configured to receive traffic can be configured to learn the Layer 2 address of any devices attached to the destination and transmit traffic that is addressed to the devices.

You can configure trunk ports as destinations, which allows trunk destinations to transmit encapsulated traffic. You can use allowed VLAN lists to configure destination trunk VLAN filtering.

Local SPAN, RSPAN, and ERSPAN Configuration Guidelines and Restrictions

These sections describe local SPAN, RSPAN, and ERSPAN configuration guidelines and restrictions:

Feature Incompatibilities

These feature incompatibilities exist with local SPAN, RSPAN, and ERSPAN:

- ES line cards do not support SPAN sessions on EVC.
- Unknown Unicast Flood Blocking (UUFB) ports cannot be RSPAN or Local SPAN egress-only destinations. (CSCsj27695)
- 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.

- 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.
- 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 Trunking Protocol (VTP)
  - Dynamic Trunking Protocol (DTP)
  - IEEE 802.1Q tunneling
- ES modules do not support SPAN session on Ethernet virtual circuits (EVC).

Note SPAN destination ports can participate in IEEE 802.3Z Flow Control.

Local SPAN, RSPAN, and ERSPAN Session Limits

For Release 12.2(33)SRC and later, Table 48-1 shows the PFC3 local SPAN, RSPAN, and ERSPAN session limits. Table 48-2 shows the PFC3 local SPAN, RSPAN, and ERSPAN source and destination limits.

Table 48-1 PFC3 Local SPAN, RSPAN, and ERSPAN Session LImits

	Local and Source Sessions		Destination Sessions	
Total Sessions	Local SPAN, RSPAN Source, ERSPAN Source Ingress or Egress or Both	Local SPAN Egress-Only	RSPAN Destination Sessions	ERSPAN Destination Sessions
80	2	14	64	23

Table 48-2 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 that is copying traffic from a single egress SPAN source port sends only egress traffic to the network analyzer. However, if you configure more than one egress SPAN source port, the traffic that is sent to the network analyzer also includes these types of ingress traffic that were received from the egress SPAN source ports:
  - Any unicast traffic that is flooded on the VLAN
  - Broadcast and multicast traffic

  This situation occurs because an egress SPAN source port receives these types of traffic from the VLAN but then recognizes itself as the source of the traffic and drops it instead of sending it back to the source from which it was received. Before the traffic is dropped, SPAN copies the traffic and sends it to the SPAN destination. (CSCds22021)
- Entering additional **monitor session** commands does not clear previously configured SPAN parameters. You must enter the **no monitor session** command to clear configured SPAN parameters.
- Connect a network analyzer to the SPAN destination.
- Within a SPAN session, all of the SPAN destinations receive all of the traffic from all of the SPAN sources, except when source-VLAN filtering is configured on the SPAN source.
- You can configure destination trunk VLAN filtering to select which traffic is transmitted from the SPAN destination.
- You can configure both Layer 2 LAN ports (LAN ports configured with the **switchport** command) and Layer 3 LAN ports (LAN ports not configured with the **switchport** command) as sources or destinations.
- You cannot mix individual source ports and source VLANs within a single session.
- If you specify multiple ingress source ports, the ports can belong to different VLANs.
- Within a session, you cannot configure both VLANs as SPAN sources and do source VLAN filtering. You can configure VLANs as SPAN sources or you can do source VLAN filtering of traffic from source ports and EtherChannels, but not both in the same session.
- You cannot configure source VLAN filtering for internal VLANs.
- When enabled, local SPAN, RSPAN, and ERSPAN use any previously entered configuration.
- When you specify sources and do not specify a traffic direction (ingress, egress, or both), “both” is used by default.
- SPAN copies Layer 2 Ethernet frames, but SPAN does not copy source trunk port Inter-Switch Link Protocol (ISL) or 802.1Q tags. You can configure destinations as trunks to send locally tagged traffic to the traffic analyzer.

  Note A destination configured as a trunk tags traffic from a Layer 3 LAN source 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 or EtherChannel can be a SPAN destination for only one SPAN session. SPAN sessions cannot share destinations.
- SPAN destinations cannot be SPAN sources.
- Destination ports never participate in any spanning tree instance. Local SPAN includes BPDUs in the monitored traffic, so any BPDUs seen on the destination are from the source. RSPAN does not support BPDU monitoring.
- All packets sent through the router for transmission from a port configured as an egress source are copied to the destination, including packets that do not exit the router through the egress port. This is because STP has put the egress port into the blocking state or, on an egress trunk port because STP has put the VLAN into the blocking state on the trunk port.

VSPAN Guidelines and Restrictions

Note Local SPAN, RSPAN, and ERSPAN all support VSPAN.

These are VSPAN guidelines and restrictions:

- VSPAN sessions do not support VLAN filtering.
- For VSPAN sessions with both ingress and egress configured, two packets are forwarded from the destination to the analyzer if the packets get switched on the same VLAN (one as ingress traffic from the ingress port and one as egress traffic from the egress port).
- VSPAN only monitors traffic that leaves or enters Layer 2 ports in the VLAN.
  - If you configure a VLAN as an ingress source and traffic gets routed into the monitored VLAN, the routed traffic is not monitored because it never appears as ingress traffic entering a Layer 2 port in the VLAN.
  - If you configure a VLAN as an egress source and traffic gets routed out of the monitored VLAN, the routed traffic is not monitored because it never appears as egress traffic leaving a Layer 2 port in the VLAN.

RSPAN Guidelines and Restrictions

These are RSPAN guidelines and restrictions:

- All participating routers must be connected by Layer 2 trunks.
- Any network device that supports RSPAN VLANs can be an RSPAN intermediate device.
- Networks impose no limit on the number of RSPAN VLANs that the networks carry.
- Intermediate network devices might impose limits on the number of RSPAN VLANs that they can support.
- You must configure the RSPAN VLANs in all source, intermediate, and destination network devices. If enabled, the 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 router 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:

- ERSPAN is supported on the PFC3B, PFC3BXL, PFC3C, and PFC3CXL.
- A WS-SUP720 (a Supervisor Engine 720 manufactured with a PFC3A), can only support ERSPAN if it has hardware version 3.2 or later. Enter the **show module version | include WS-SUP720-BASE** command to display the hardware version. For example:

```
Router# show module version | include WS-SUP720-BASE
 7    2  WS-SUP720-BASE     SAD075301SZ Hw :3.2
```

- For ERSPAN packets, the “protocol type” field value in the GRE header is 0x88BE.
- The payload of a Layer 3 ERSPAN packet is a copied Layer 2 Ethernet frame, excluding any ISL or 802.1Q tags.
- ERSPAN adds a 50-byte header to each copied Layer 2 Ethernet frame and replaces the 4-byte cyclic redundancy check (CRC) trailer.
- ERSPAN supports jumbo frames that contain Layer 3 packets of up to 9,202 bytes. If the length of the copied Layer 2 Ethernet frame is greater than 9,170 (9,152-byte Layer 3 packet), ERSPAN truncates the copied Layer 2 Ethernet frame to create a 9,202-byte ERSPAN Layer 3 packet.
- Regardless of any configured MTU size, ERSPAN creates Layer 3 packets that can be as long as 9,202 bytes. ERSPAN traffic might be dropped by any interface in the network that enforces an MTU size smaller than 9,202 bytes.
- With the default MTU size (1,500 bytes), if the length of the copied Layer 2 Ethernet frame is greater than 1,468 bytes (1,450-byte Layer 3 packet), the ERSPAN traffic is dropped by any interface in the network that enforces the 1,500-byte MTU size.

Note The **mtu** interface command and the **system jumbomtu** command (see the “Configuring Jumbo Frame Support” section on page 8-8) set the maximum Layer 3 packet size (default is 1,500 bytes, maximum is 9,216 bytes).

- All participating routers 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 Differentiated Services Code Point (DSCP) value to prioritize ERSPAN traffic for Quality of Service (QoS).
- The only supported destination for ERSPAN traffic is an ERSPAN destination session on a PFC3.
- All ERSPAN source sessions on a router must use the same origin IP address, configured with the **origin ip address** command (see the “Configuring ERSPAN Source Sessions” section on page 48-25).
- 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 48-27).
- The ERSPAN source session’s destination IP address, which must be configured on an interface on the destination router, is the source of traffic that an ERSPAN destination session sends to the destinations. You configure the same address in both the source and destination sessions with the **ip address** command.
- The ERSPAN ID differentiates the ERSPAN traffic arriving at the same destination IP address from various different ERSPAN source sessions.
- ERSPAN egress is not supported on EVC ports.

Configuring Local SPAN, RSPAN, and ERSPAN

These sections describe how to configure local SPAN, RSPAN, and ERSPAN:

Configuring a Destination as an Unconditional Trunk (Optional)

To tag the monitored traffic as it leaves a destination, configure the destination as a trunk before you configure it as a destination.

To configure the destination as a trunk, perform this task in interface configuration mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface {type slot/port | port-channel number}`	Selects the interface to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**
Step 3	`Router(config-if)# switchport`	Configures the interface 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 interface as either an ISL or 802.1Q trunk.
Step 5	`Router(config-if)# switchport mode trunk`	Configures the port to trunk unconditionally.

This example shows how to configure a port as an unconditional IEEE 802.1Q trunk:

```
Router# configure terminal
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
```

Note Releases earlier than Release 12.2(33)SRC required you to enter the **switchport nonegotiate** command when you configured a destination port as an unconditional trunk. This requirement has been removed in Release 12.2(33)SRC and later.

Configuring Destination Trunk VLAN Filtering (Optional)

Note In addition to filtering VLANs on a trunk, you can also apply the allowed VLAN list to access ports.

Destination trunk VLAN filtering is applied at the destination. Destination trunk VLAN filtering does not reduce the amount of traffic being sent from the SPAN sources to the SPAN destinations.

When a destination is a trunk, you can use the list of VLANs allowed on the trunk to filter the traffic transmitted from the destination. (CSCeb01318)

Destination trunk VLAN filtering removes the restriction that, within a SPAN session, all destinations receive all the traffic from all the sources. Destination trunk VLAN filtering allows you to select, on a per-VLAN basis, the traffic that is transmitted from each destination trunk to the network analyzer.

To configure destination trunk VLAN filtering on a destination trunk, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface type slot/port`	Selects the destination trunk port to configure. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**
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.

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
```

Configuring Destination Port Permit Lists (Optional)

To prevent accidental configuration of ports as destinations, you can create a permit list of the ports that are valid for use as destinations. With a destination port permit list configured, you can only configure the ports in the permit list as destinations.

To configure a destination port permit list, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor permit-list`	Enables use of the destination port permit list.
Step 3	`Router(config)# monitor permit-list destination interface` *type slot/port*`[-`*port*`] [, `*type slot/port* `- `*port*`]`	Configures a destination port permit list or adds to an existing destination port permit list. *type*—**ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**
Step 4	`Router(config)# do show monitor permit-list`	Verifies the configuration.

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

These sections describe how to configure local SPAN sessions:

- Configuring Local SPAN (SPAN Configuration Mode), page 48-15
- Configuring Local SPAN (Global Configuration Mode), page 48-17

Configuring Local SPAN (SPAN Configuration Mode)

Note To tag the monitored traffic as it leaves a destination, you must configure the destination to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination as an Unconditional Trunk (Optional)" section on page 48-12).

To configure a local SPAN session in SPAN configuration mode, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session` *local_span_session_number* `type [local | local-tx]`	Configures a local SPAN session number and enters local SPAN session configuration mode. • Enter the **local** keyword to configure ingress or egress or both SPAN sessions. • Enter the **local-tx** keyword to configure egress-only SPAN sessions.
Step 3	`Router(config-mon-local)# description` *session_description*	(Optional) Describes the local SPAN session.
Step 4	`Router(config-mon-local)# source {`*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list* | *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*`} [rx | tx | both]`	Associates the local SPAN session number with source ports or VLANs, and selects the traffic direction to be monitored. **Note** When you enter the **local-tx** keyword in the **monitor session** command, the **rx** and **both** keywords are not available and the **tx** keyword is required. To make best use of the available SPAN sessions, it is always preferable to configure **local-tx** sessions instead of **local** sessions with the **tx** keyword.
Step 5	`Router(config-mon-local)# filter {`*single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*`}`	(Optional) Configures source VLAN filtering when the local SPAN source is a trunk port.
Step 6	`Router(config-mon-local)# destination {`*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list*`} [ingress [learning]]`	Associates the local SPAN session number with the destinations.
Step 7	`Router(config-mon-local)# no shutdown`	Activates the local SPAN session. **Note** The **no shutdown** and **shutdown** commands are not supported for local-tx egress-only SPAN sessions.
Step 8	`Router(config-mon-local)# end`	Exits configuration mode.

When configuring monitor sessions, note the following information:

- *session_description* can be up to 240 characters and cannot contain special characters; with Release 12.2(33)SRC and later, the description can contain spaces.

Note You can enter 240 characters after the **description** command.

- *local_span_session_number* can range from 1 to 80.
- *single_interface* is:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

    Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 12-7.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the switch to transmit traffic that is addresses to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, not the following:
  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 10-6.
  - If the destination is a trunk and the attached device transmits tagged traffic back to the router, you can use either ISL or 802.1Q trunking.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the router, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.
  - Do not configure the destination with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.
  - Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

This example shows how to configure session 1 to monitor ingress traffic from Gigabit Ethernet port 1/1 and configure Gigabit Ethernet port 1/2 as the destination:

```
Router# configure terminal
Router(config)# monitor session 1 type local
Router(config-mon-local)# source interface gigabitethernet 1/1 rx
```

```
Router(config-mon-local)# destination interface gigabitethernet 1/2
Router(config-mon-local)# no shutdown
Router(config-mon-local)# end
```

For additional examples, see the "Configuration Examples" section on page 48-30 .

Configuring Local SPAN (Global Configuration Mode)

Note To tag the monitored traffic as it leaves a destination, you must configure the destination to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination as an Unconditional Trunk (Optional)" section on page 48-12).

You can configure up to two local SPAN sessions in global configuration mode.

You can use SPAN configuration mode for all SPAN configuration tasks.

You must use SPAN configuration mode to configure the supported maximum number of SPAN sessions.

Local SPAN does not use separate source and destination sessions. To configure a local SPAN session, configure local SPAN sources and destinations with the same session number. To configure a local SPAN session, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session` *local_span_session_number* **source** {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list* | *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*} [**rx** | **tx** | **both**]	Associates the local SPAN source session number with the source ports or VLANs and selects the traffic direction to be monitored.
Step 3	`Router(config)# monitor session` *local_span_session_number* **destination** {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list*} [**ingress** [**learning**]]	Associates the local SPAN session number and the destinations.

When configuring local SPAN sessions, note the following information:

- *local_span_session_number* can range from 1 to 66.
- *single_interface* is:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 12-7.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- Enter the **ingress** keyword to configure destinations to receive traffic from attached services.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the router to transmit traffic that is addressed to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:

  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 8-6.
  - If the destination is a trunk and the attached device transmits tagged traffic back to the router, you can use either ISL or 802.1Q trunking.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the router, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.
  - Do not configure the destination with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.
  - Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

This example shows how to configure Fast Ethernet port 5/1 as a bidirectional source for session 1:

```
Router(config)# monitor session 1 source interface fastethernet 5/1
```

This example shows how to configure Fast Ethernet port 5/48 as the destination for SPAN session 1:

```
Router(config)# monitor session 1 destination interface fastethernet 5/48
```

For additional examples, see the "Configuration Examples" section on page 48-30.

Configuring RSPAN

RSPAN uses a source session on one router and a destination session on a different router. 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`	Configures the VLAN as an RSPAN VLAN.
Step 4	`Router(config-vlan)# end`	Updates the VLAN database and returns to privileged EXEC mode.

Configuring RSPAN Sessions (SPAN Configuration Mode)

These sections describe how to configure RSPAN sessions in SPAN configuration mode:

- Configuring RSPAN Source Sessions in SPAN Configuration Mode, page 48-19
- Configuring RSPAN Destination Sessions in SPAN Configuration Mode, page 48-20

Configuring RSPAN Source Sessions in SPAN Configuration Mode

To configure an RSPAN source session in SPAN configuration mode, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session` *RSPAN_source_session_number* `type rspan-source`	Configures an RSPAN source session number and enters RSPAN source session configuration mode for the session.
Step 3	`Router(config-mon-rspan-src)# description` *session_description*	(Optional) Describes the RSPAN source session.
Step 4	`Router(config-mon-rspan-src)# source` {*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 source ports or VLANs, and selects the traffic direction to be monitored.
Step 5	`Router(config-mon-rspan-src)# filter` {*single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*}	(Optional) Configures source VLAN filtering when the RSPAN source is a trunk port.
Step 6	`Router(config-mon-rspan-src)# destination remote vlan` *rspan_vlan_ID*	Associates the RSPAN source session number session number with the RSPAN VLAN.
Step 7	`Router(config-mon-rspan-src)# no shutdown`	Activates the RSPAN source session.
Step 8	`Router(config-mon-rspan-src)# end`	Exits configuration mode.

When configuring RSPAN source sessions, note the following information:

- *session_description* can be up to 240 characters and cannot contain special characters; with Release 12.2(33)SRC and later, the description can contain spaces.

Note You can enter 240 characters after the **description** command.

- *RSPAN_source_span_session_number* can range from 1 to 80.
- *single_interface* is:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port_channel** *number*
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- See the "Configuring RSPAN VLANs" section on page 48-19 for information about the RSPAN VLAN ID.

This example shows how to configure session 1 to monitor bidirectional traffic from Gigabit Ethernet port 1/1:

```
Router# configure terminal
Router(config)# monitor session 1 type rspan-source
Router(config-mon-rspan-src)# source interface gigabitethernet 1/1
Router(config-mon-rspan-src)# destination remote vlan 2
Router(config-mon-rspan-src)# no shutdown
Router(config-mon-rspan-src)# end
```

For additional examples, see the "Configuration Examples" section on page 48-30.

Configuring RSPAN Destination Sessions in SPAN Configuration Mode

Note To tag the monitored traffic, you must configure the port to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination as an Unconditional Trunk (Optional)" section on page 48-12).

You can configure an RSPAN destination session on the RSPAN source session router 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_source_session_number* `type rspan-destination`	Configures an RSPAN destination session number and enters RSPAN destination session configuration mode for the session.
Step 3	`Router(config-mon-rspan-dst)# description` *session_description*	(Optional) Describes the RSPAN destination session.
Step 4	`Router(config-mon-rspan-dst)# source remote vlan` rspan_vlan_ID	Associates the RSPAN destination session number RSPAN VLAN.
Step 5	`Router(config-mon-rspan-dst)# destination` {single_interface | interface_list | interface_range | mixed_interface_list} [**ingress** [**learning**]]	Associates the RSPAN destination session number with the RSPAN VLAN.
Step 6	`Router(config-mon-rspan-dst)# end`	Exits configuration mode.

When configuring RSPAN destination sessions, note the following information:

- *RSPAN_destination_session_number* can range from 1 to 80.
- *single_interface* is:
  - **interface** *type slot/port*; type is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

  Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 12-7.

- *interface_list* is *single_interface* **,** *single_interface* **,** *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* **,** *interface_range* **,** ...
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the switch to transmit traffic that is addressed to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:
  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 10-6.
  - If the destination is a trunk and the attached device transmits tagged traffic back to the switch, you can use either ISL or 802.1Q trunking.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the switch, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.

  - Do not configure the destinations with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.
  - Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.
- The **no shutdown** and **shutdown** commands are not supported for RSPAN destination sessions.

This example shows how to configure RSPAN VLAN 2 as the source for session 1 and Gigabit Ethernet port 1/2 as the destination:

```
Router# configure terminal
Router(config)# monitor session 1 type rspan-destination
Router(config-rspan-dst)# source remote vlan2
Router(config-rspan-dst)# destination interface gigabitethernet 1/2
Router(config-rspan-dst)# end
```

For additional examples, see the "Configuration Examples" section on page 48-30.

Configuring RSPAN Sessions (Global Configuration Mode)

These sections describe how to configure RSPAN sessions in global configuration mode

- Configuring RSPAN Source Sessions in Global Configuration Mode, page 48-22
- Configuring RSPAN Destination Sessions in Global Configuration Mode, page 48-23

Configuring RSPAN Source Sessions in Global Configuration Mode

To configure an RSPAN source session in global configuration mode, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session` *RSPAN_source_session_number* **source** {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list* | *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*} [**rx** | **tx** | **both**]	Associates the RSPAN source 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 wit the RSPAN VLAN.

When configuring RSPAN source sessions, note the following information:

- To configure RSPAN VLANs, see the "Configuring RSPAN VLANs" section on page 48-19.
- *RSPAN_source_session_number* can range from 1 to 66.
- *single_interface* is:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- See the "Configuring RSPAN VLANs" section on page 48-19 for information about the RSPAN VLAN ID.

This example shows how to configure Fast Ethernet port 5/2 as the source for session 2:

```
Router(config)# monitor session 2 source interface fastethernet 5/2
```

This example shows how to configure RSPAN VLAN 200 as the destination for session 2:

```
Router(config)# monitor session 2 destination remote vlan 200
```

For additional examples, see the "Configuration Examples" section on page 48-30.

Configuring RSPAN Destination Sessions in Global Configuration Mode

Note To tag the monitored traffic, you must configure the port to trunk unconditionally before you configure it as a destination (see the "Configuring a Destination as an Unconditional Trunk (Optional)" section on page 48-12).

You can configure an RSPAN destination session on the RSPAN source session router to monitor RSPAN traffic locally.

To configure an RSPAN destination session in global configuration mode, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session` *RSPAN_destination_session_number* `source remote vlan` *rspan_vlan_ID*	Associates the RSPAN destination session number with the RSPAN VLAN.
Step 3	`Router(config)# monitor session` *RSPAN_destination_session_number* `destination` {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list*} [`ingress` {`learning`}]	Associates the RSPAN destination session number with the destinations.

When configuring monitor sessions, not the following information:

- *RSPAN_destination_session_number* can range from 1 to 66.
- See the "Configuring RSPAN VLANs" section on page 48-19 for information about the RSPAN VLAN ID.
- *single_interface* is:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

  Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 12-7.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

  Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the switch to transmit traffic that is addresses to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:
  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 10-6.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the switch, you can use either ISL or 802.1Q trunking.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the switch, use 802.1Q trunking with the native VLAN configured to accept the traffic from the attached device.
  - Do not configure the destinations with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.
  - Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

This example shows how to configure RSPAN VLAN 200 as the source for session 3:

```
Router(config)# monitor session 3 source remote vlan 200
```

This example shows how to configure Fast Ethernet port 5/47 as the destination for session 3:

```
Router(config)# monitor session 3 destination interface fastethernet 5/4
```

For additional examples, see the "Configuration Examples" section on page 48-30.

Configuring ERSPAN

ERSPAN uses separate source and destination sessions. You configure the source and destination sessions on different routers. These sections describe how to configure ERSPAN sessions:

- Configuring ERSPAN Source Sessions, page 48-25
- Configuring ERSPAN Destination Sessions, page 48-27

Note The PFC3 supports ERSPAN (see the "ERSPAN Guidelines and Restrictions" section on page 48-10).

Configuring ERSPAN Source Sessions

To configure an ERSPAN source session, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session` *ERSPAN_source_session_number* `type erspan-source`	Configures an ERSPAN source session number and enters ERSPAN source session configuration mode for the session.
Step 3	`Router(config-mon-erspan-src)# description` *session_description*	(Optional) Describes the ERSPAN source session.
Step 4	`Router(config-mon-erspan-src)# source` {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list* | *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*} [**rx** | **tx** | **both**]	Associates the ERSPAN source session number with the CPU, the source ports, or VLANs, and selects the traffic direction to be monitored.
Step 5	`Router(config-mon-erspan-src)# filter` {*single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*}	(Optional) Configures source VLAN filtering when the ERSPAN source is a trunk port.
Step 6	`Router(config-mon-erspan-src)# destination`	Enters ERSPAN source session destination configuration mode.
Step 7	`Router(config-mon-erspan-src-dst)# ip address` *ip_address*	Configures the ERSPAN flow destination IP address, which must also be configured on an interface on the destination router and be entered in the ERSPAN destination session configuration (see the "Configuring ERSPAN Destination Sessions" section on page 48-27, Step 6).
Step 8	`Router(config-mon-erspan-src-dst)# erspan-id` *ERSPAN_flow_id*	Configures the ID number used by the source and destination sessions to identify the ERSPAN traffic, which must also be entered in the ERSPAN destination session configuration (see the "Configuring ERSPAN Destination Sessions" section on page 48-27, Step 7).
Step 9	`Router(config-mon-erspan-src-dst)# origin ip address` *ip_address* [**force**]	Configures the IP address used as the source of the ERSPAN traffic.
Step 10	`Router(config-mon-erspan-src-dst)# ip ttl` *ttl_value*	(Optional) Configures the IP time-to-live (TTL) value of the packets in the ERSPAN traffic.
Step 11	`Router(config-mon-erspan-src-dst)# ip prec` *ipp_value*	(Optional) Configures the IP precedence value of the packets in the ERSPAN traffic.

	Command	Purpose
Step 12	`Router(config-mon-erspan-src-dst)# ip dscp dscp_value`	(Optional) Configures the IP DSCP value of the packets in the ERSPAN traffic.
Step 13	`Router(config-mon-erspan-src-dst)# vrf vrf_name`	(Optional) Configures the VRF name to use instead of the global routing table.
Step 14	`Router(config-mon-erspan-src)# no shutdown`	Activates the ERSPAN source session.
Step 15	`Router(config-mon-erspan-src-dst)# end`	Exits configuration mode.

When configuring monitor sessions, note the following information:

- *session_description* can be up to 240 characters and cannot contain special characters. With Release 12.2(33)SRC and later, the description can contain spaces.

  Note You can enter 240 characters after the **description** command.

- *ERSPAN_source_session_number* can range from 1 to 66.
- *single_interface* is:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

    Note Port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the"Configuring EtherChannels" section on page 12-7.

- *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 router.
- *ttl_value* can range from 1 to 255.
- *ipp_value* can range from 0 to 7.
- *dscp_value* can range from 0 to 63.

This example shows how to configure session 3 to monitor bidirectional traffic from Gigabit Ethernet port 4/1:

```
Router# configure terminal
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
Router(config-mon-erspan-src-dst)# end
```

For additional examples, see the "Configuration Examples" section on page 48-30.

Configuring ERSPAN Destination Sessions

Note You cannot monitor ERSPAN traffic locally.

To configure an ERSPAN destination session, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session` *ERSPAN_destination_session_number* `type erspan-destination`	Configures an ERSPAN destination session number and enters ERSPAN destination session configuration mode for the session.
Step 3	`Router(config-mon-erspan-dst)# description` *session_description*	(Optional) Describes the ERSPAN destination session.
Step 4	`Router(config-mon-erspan-dst)# destination` {*single_interface* | *interface_list* | *interface_range* | *mixed_interface_list*} [**ingress** [**learning**]	Associates the ERSPAN destination session number with the destination ports.
Step 5	`Router(config-mon-erspan-dst)# source`	Enters ERSPAN destination session source configuration mode.
Step 6	`Router(config-mon-erspan-dst-src)# ip address` *ip_address* [**force**]	Configures the ERSPAN flow destination IP address. This must be an address on a local interface and match the address that you entered in the "Configuring ERSPAN Source Sessions" section on page 48-25, Step 7.
Step 7	`Router(config-mon-erspan-dst-src)# erspan-id` *ERSPAN_flow_id*	Configures the ID number used by the destination and destination sessions to identify the ERSPAN traffic. This must match the ID that you entered in the "Configuring ERSPAN Source Sessions" section on page 48-25, Step 8.
Step 8	`Router(config-mon-erspan-dst-src)# vrf` *vrf_name*	(Optional) Configures the VRF name used instead of the global routing table.
Step 9	`Router(config-mon-erspan-dst)# no shutdown`	Activates the ERSPAN destination session.
Step 10	`Router(config-mon-erspan-dst-src)# end`	Exits configuration mode.

When configuring monitor sessions, note the following information:

- *ERSPAN_destination_session_number* can range from 1 to 66.
- *single_interface* is:
  - **interface** *type slot/port*; *type* is **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
  - **interface port-channel** *number*

  Note Destination port channel interfaces must be configured with the **channel-group** *group_num* **mode on** command and the **no channel-protocol** command. See the "Configuring EtherChannels" section on page 12-7.

- *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 router.

  Note You must also change all ERSPAN source session destination IP addresses (see the "Configuring ERSPAN Source Sessions" section on page 48-25, Step 7).

- *ERSPAN_flow_id* can range from 1 to 1023.
- Enter the **ingress** keyword to configure destinations to receive traffic from attached devices.
- Enter the **learning** keyword to enable MAC address learning from the destinations, which allows the router to transmit traffic that is addressed to devices attached to the destinations.

  When configuring destinations with the **ingress** and **learning** keywords, note the following:
  - Configure the destinations for Layer 2 switching. See the "Configuring LAN Interfaces for Layer 2 Switching" section on page 10-6.
  - If the destination is a trunk and the attached device transmits traffic back to the router, you can use either ISL or 802.1Q trunking.
  - If the destination is a trunk and the attached device transmits untagged traffic back to the router, use 802.1Q trunking with native VLAN configured to accept the traffic from the attached device.
  - Do not configure the destinations with Layer 3 addresses. Use a VLAN interface to route traffic to and from devices attached to destinations.
  - Destinations are held in the down state. To route the traffic to and from attached devices, configure an additional active Layer 2 port in the VLAN to keep the VLAN interface up.

This example shows how to configure an ERSPAN destination session to send ERSPAN ID 101 traffic arriving at IP address 10.1.1.1 to Gigabit Ethernet port 2/1:

```
Router# configure terminal
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 48-30.

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 48-25.

To configure source VLAN filtering when the local SPAN or RSPAN source is a trunk port, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session` *session_number* `filter` *single_vlan* | *vlan_list* | *vlan_range* | *mixed_vlan_list*	Configures source VLAN filtering when the local SPAN or RSPAN source is a trunk port.

When configuring source VLAN filtering, note the following information:

- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID* - *last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...

This example shows how to monitor VLANs 1 through 5 and VLAN 9 when the source is a trunk port:

```
Router(config)# monitor session 2 filter vlan 1 - 5 , 9
```

Verifying the Configuration

To verify the configuration, enter the **show monitor session** command.

This example shows how to verify the configuration of session 2:

```
Router# show monitor session 2
Session 2
------------
Type : Remote Source Session

Source Ports:
    RX Only:       Fa3/1
Dest RSPAN VLAN:   901
Router#
```

This example shows how to display the full details of session 2:

```
Router# show monitor session 2 detail
Session 2
------------
Type : Remote Source Session

Source Ports:
    RX Only:       Fa1/1-3
    TX Only:       None
    Both:          None
Source VLANs:
    RX Only:       None
    TX Only:       None
    Both:          None
Source RSPAN VLAN: None
Destination Ports: None
Filter VLANs:      None
Dest RSPAN VLAN:   901
```

Configuration Examples

This example shows the configuration of RSPAN source session 2:

```
Router(config)# monitor session 2 source interface fastethernet1/1 - 3 rx
Router(config)# monitor session 2 destination remote vlan 901
```

This example shows how to clear the configuration for sessions 1 and 2:

```
Router(config)# no monitor session range 1-2
```

This example shows the configuration of an RSPAN source session with multiple sources:

```
Router(config)# monitor session 2 source interface fastethernet 5/15 , 7/3 rx
Router(config)# monitor session 2 source interface gigabitethernet 1/2 tx
Router(config)# monitor session 2 source interface port-channel 102
Router(config)# monitor session 2 source filter vlan 2 - 3
Router(config)# monitor session 2 destination remote vlan 901
```

This example shows how to remove sources for a session:

```
Router(config)# no monitor session 2 source interface fastethernet 5/15 , 7/3
```

This example shows how to remove options for sources for a session:

```
Router(config)# no monitor session 2 source interface gigabitethernet 1/2
Router(config)# no monitor session 2 source interface port-channel 102 tx
```

This example shows how to remove 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 Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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 router 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 router reboots, but many applications (for example, device inventory, billing, and fault detection) require maintenance of this correspondence.

You can poll the router 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 Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding How Power Management Works

These sections describe power management in the Cisco 7600 series routers:

Note Installed power supplies in a system can be of different wattage ratings. Installed power supplies can also be both AC-input, both DC-input, or one AC-input and one DC-input. Power supplies can be configured in either redundant or non-redundant mode. For detailed information on supported power supply configurations, refer to the *Cisco 7600 Series Router 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. • Both power supplies come on. The total available wattage is the output wattage of the higher wattage power supply. When system power usage exceeds the maximum sharing limit of lower wattage power supply, system will shutdown the lower capacity supply to protect it from overcurrent.
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 12.2SX on the Supervisor Engine 720, Supervisor Engine 32, and Supervisor Engine 2* publication at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/ol_4164.htm

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 Cisco 7600 series router:

```
Router# show platform hardware capacity cpu
CPU Resources
  CPU utilization: Module             5 seconds       1 minute       5 minutes
                   1  RP               0% /  0%             1%             1%
```

```
                    1  SP                5% /  0%             5%              4%
                    7                   69% /  0%            69%             69%
                    8                   78% /  0%            74%             74%

  Processor memory: Module   Bytes:       Total           Used       %Used
                     1  RP              176730048       51774704       29%
                     1  SP              192825092       51978936       27%
                     7                  195111584       35769704       18%
                     8                  195111584       35798632       18%
  I/O memory: Module          Bytes:       Total           Used       %Used
              1  RP                      35651584       12226672       34%
              1  SP                      35651584        9747952       27%
              7                          35651584        9616816       27%
              8                          35651584        9616816       27%
Router#
```

This example shows how to display EOBC-related statistics for the route processor, the switch processor, and the DFCs in the Cisco 7600 series router:

```
Router# show platform hardware capacity eobc EOBC Resources
  Module                      Packets/sec     Total packets     Dropped packets
  1  RP      Rx:                       61            108982                   0
             Tx:                       37             77298                   0
  1  SP      Rx:                       34            101627                   0
             Tx:                       39            115417                   0
  7          Rx:                        5             10358                   0
             Tx:                        8             18543                   0
  8          Rx:                        5             12130                   0
             Tx:                       10             20317                   0
Router#
```

This example shows how to display the current and peak switching utilization:

```
Router# show platform hardware capacity fabric Switch Fabric Resources
  Bus utilization: current is 100%, peak was 100% at 12:34 12mar45
  Fabric utilization:      ingress                      egress
                                                        current peak
      Module channel speed current peak                 current peak
       1      0       20G   100%  100% 12:34 12mar45  100%    100% 12:34 12mar45
       1      1       20G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
       4      0       20G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
      13      0        8G   12%    80% 12:34 12mar45   12%     80% 12:34 12mar45
Router#
```

This example shows how to display information about the total capacity, the bytes used, and the percentage that is used for the flash and NVRAM resources present in the system:

```
Router# show platform hardware capacity flash
Flash/NVRAM Resources
  Usage: Module Device              Bytes:      Total          Used     %Used
         1  RP  bootflash:                   31981568      15688048       49%
         1  SP  disk0:                      128577536     105621504       82%
         1  SP  sup-bootflash:               31981568      29700644       93%
         1  SP  const_nvram:                   129004           856        1%
         1  SP  nvram:                         391160         22065        6%
         7      dfc#7-bootflash:            15204352        616540        4%
         8      dfc#8-bootflash:            15204352             0        0%
Router#
```

This example shows how to display the capacity and utilization of the EARLs present in the system:

```
Router# show platform hardware capacity forwarding
L2 Forwarding Resources
```

```
          MAC Table usage:   Module  Collisions  Total       Used     %Used
                             6               0  65536         11        1%
            VPN CAM usage:                        Total       Used     %Used
                                                    512          0        0%
L3 Forwarding Resources
             FIB TCAM usage:                     Total       Used     %Used
                  72 bits (IPv4, MPLS, EoM)     196608         36        1%
                144 bits (IP mcast, IPv6)        32768          7        1%

                    detail:      Protocol                   Used     %Used
                                 IPv4                         36        1%
                                 MPLS                          0        0%
                                 EoM                           0        0%

                                 IPv6                          4        1%
                                 IPv4 mcast                    3        1%
                                 IPv6 mcast                    0        0%

            Adjacency usage:                     Total       Used     %Used
                                                1048576        175        1%

     Forwarding engine load:
                    Module       pps   peak-pps  peak-time
                    6              8       1972  02:02:17 UTC Thu Apr 21 2005

Netflow Resources
          TCAM utilization:      Module       Created      Failed     %Used
                                 6                  1           0        0%
          ICAM utilization:      Module       Created      Failed     %Used
                                 6                  0           0        0%

                Flowmasks:   Mask#   Type        Features
                       IPv4:     0   reserved    none
                       IPv4:     1   Intf FulNAT_INGRESS NAT_EGRESS FM_GUARDIAN
                       IPv4:     2   unused      none
                       IPv4:     3   reserved    none

                       IPv6:     0   reserved    none
                       IPv6:     1   unused      none
                       IPv6:     2   unused      none
                       IPv6:     3   reserved    none

CPU Rate Limiters Resources
             Rate limiters:       Total         Used      Reserved     %Used
                    Layer 3           9            4             1       44%
                    Layer 2           4            2             2       50%

ACL/QoS TCAM Resources
  Key: ACLent - ACL TCAM entries, ACLmsk - ACL TCAM masks, AND - ANDOR,
       QoSent - QoS TCAM entries, QOSmsk - QoS TCAM masks, OR - ORAND,
       Lbl-in - ingress label, Lbl-eg - egress label, LOUsrc - LOU source,
       LOUdst - LOU destination, ADJ - ACL adjacency

  Module ACLent ACLmsk QoSent QoSmsk Lbl-in Lbl-eg LOUsrc LOUdst  AND OR  ADJ
  6          1%     1%     1%     1%     1%     1%     0%     0%   0% 0%   1%

Router#
```

This example shows how to display the interface resources:

```
Router# show platform hardware capacity interface Interface Resources
  Interface drops:
    Module    Total drops:    Tx            Rx      Highest drop port: Tx  Rx
```

```
    9                           0             2                           0  48

  Interface buffer sizes:
    Module                              Bytes:     Tx buffer           Rx buffer
         1                                             12345               12345
         5                                             12345               12345
Router#
```

This example shows how to display SPAN information:

```
Router# show platform hardware capacity monitor SPAN Resources
  Source sessions: 2 maximum, 0 used
    Type                                    Used
    Local                                      0
    RSPAN source                               0
    ERSPAN source                              0
    Service module                             0
  Destination sessions: 64 maximum, 0 used
    Type                                    Used
    RSPAN destination                          0
    ERSPAN destination (max  24)               0
Router#
```

This example shows how to display the capacity and utilization of resources for Layer 3 multicast functionality:

```
Router# show platform hardware capacity multicast
L3 Multicast Resources
  IPv4 replication mode: ingress
  IPv6 replication mode: ingress
  Bi-directional PIM Designated Forwarder Table usage: 4 total, 0 (0%) used
  Replication capability: Module                                IPv4     IPv6
                          5                                   egress   egress
                          9                                  ingress  ingress
  MET table Entries: Module                                Total    Used    %Used
                     5                                     65526       6       0%
Router#
```

This example shows how to display information about the system power capacities and utilizations:

```
Router# show platform hardware capacity power
Power Resources
  Power supply redundancy mode: administratively combined operationally combined
  System power: 1922W, 0W (0%) inline, 1289W (67%) total allocated
  Powered devices: 0 total
Router#
```

This example shows how to display the capacity and utilization of QoS policer resources for each EARL in the Cisco 7600 series router.

```
Router# show platform hardware capacity qos
QoS Policer Resources
  Aggregate policers: Module                      Total         Used     %Used
                      1                            1024          102       10%
                      5                            1024            1        1%
  Microflow policer configurations: Module        Total         Used     %Used
                                    1                64           32       50%
                                    5                64            1        1%
Router#
```

This example shows how to display information about the key system resources:

```
Router# show platform hardware capacity systems System Resources
  PFC operating mode: PFC3BXL
  Supervisor redundancy mode: administratively rpr-plus, operationally rpr-plus
  Switching Resources: Module   Part number              Series       CEF mode
                       5        WS-SUP720-BASE        supervisor            CEF
                       9        WS-X6548-RJ-45            CEF256            CEF
Router#
```

This example shows how to display VLAN information:

```
Router# show platform hardware capacity vlan VLAN Resources
  VLANs: 4094 total, 10 VTP, 0 extended, 0 internal, 4084 free Router#
```

Determining Sensor Temperature Threshold

The system sensors set off alarms based on different temperature threshold settings. You can determine the allowed temperatures for the sensors by using the **show environment alarm threshold** command.

This example shows how to determine sensor temperature thresholds:

```
Router> show environment alarm threshold
environmental alarm thresholds:

power-supply 1 fan-fail: OK
  threshold #1 for power-supply 1 fan-fail:
    (sensor value != 0) is system minor alarm power-supply 1 power-output-fail: OK
  threshold #1 for power-supply 1 power-output-fail:
    (sensor value != 0) is system minor alarm fantray fan operation sensor: OK
  threshold #1 for fantray fan operation sensor:
    (sensor value != 0) is system minor alarm operating clock count: 2
  threshold #1 for operating clock count:
    (sensor value < 2) is system minor alarm
  threshold #2 for operating clock count:
    (sensor value < 1) is system major alarm operating VTT count: 3
  threshold #1 for operating VTT count:
    (sensor value < 3) is system minor alarm
  threshold #2 for operating VTT count:
    (sensor value < 2) is system major alarm VTT 1 OK: OK
  threshold #1 for VTT 1 OK:
    (sensor value != 0) is system minor alarm VTT 2 OK: OK
  threshold #1 for VTT 2 OK:
    (sensor value != 0) is system minor alarm VTT 3 OK: OK
  threshold #1 for VTT 3 OK:
    (sensor value != 0) is system minor alarm clock 1 OK: OK
  threshold #1 for clock 1 OK:
    (sensor value != 0) is system minor alarm clock 2 OK: OK
  threshold #1 for clock 2 OK:
    (sensor value != 0) is system minor alarm module 1 power-output-fail: OK
  threshold #1 for module 1 power-output-fail:
    (sensor value != 0) is system major alarm module 1 outlet temperature: 21C
  threshold #1 for module 1 outlet temperature:
    (sensor value > 60) is system minor alarm
  threshold #2 for module 1 outlet temperature:
    (sensor value > 70) is system major alarm module 1 inlet temperature: 25C
  threshold #1 for module 1 inlet temperature:
    (sensor value > 60) is system minor alarm
  threshold #2 for module 1 inlet temperature:
    (sensor value > 70) is system major alarm module 1 device-1 temperature: 30C
  threshold #1 for module 1 device-1 temperature:
    (sensor value > 60) is system minor alarm
```

```
threshold #2 for module 1 device-1 temperature:
  (sensor value > 70) is system major alarm module 1 device-2 temperature: 29C
threshold #1 for module 1 device-2 temperature:
  (sensor value > 60) is system minor alarm
threshold #2 for module 1 device-2 temperature:
  (sensor value > 70) is system major alarm module 5 power-output-fail: OK
threshold #1 for module 5 power-output-fail:
  (sensor value != 0) is system major alarm module 5 outlet temperature: 26C
threshold #1 for module 5 outlet temperature:
  (sensor value > 60) is system minor alarm
threshold #2 for module 5 outlet temperature:
  (sensor value > 75) is system major alarm module 5 inlet temperature: 23C
threshold #1 for module 5 inlet temperature:
  (sensor value > 50) is system minor alarm
threshold #2 for module 5 inlet temperature:
  (sensor value > 65) is system major alarm EARL 1 outlet temperature: N/O
threshold #1 for EARL 1 outlet temperature:
  (sensor value > 60) is system minor alarm
threshold #2 for EARL 1 outlet temperature:
  (sensor value > 75) is system major alarm EARL 1 inlet temperature: N/O
threshold #1 for EARL 1 inlet temperature:
  (sensor value > 50) is system minor alarm
threshold #2 for EARL 1 inlet temperature:
  (sensor value > 65) is system major alarm
```

Understanding 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 *Cisco 7600 Series Router 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.

Table 50-2 Environmental Monitoring for Supervisor Engine and Switching Modules (continued)

Component	Alarm Type	LED Indication	Action
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.
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 Cisco 7600 series routers:

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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 router while the router 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 router 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 router 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

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 router. 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 router 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)# 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 non-disruptive 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 non disruptive tests.

To display the available tests and their attributes, and determine which commands are in the non disruptive 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 router 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 router 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 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 supervisors, 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 (PFC) of the supervisor engine.
  - Reset the system or the module you are testing before returning the system to normal operating mode.

- Recommended tasks:
  - If you have a distributed forwarding card (DFC) installed, remove all switching modules and then reboot the system before starting the memory test on the central PFC of the supervisor engine or route switch processor.
  - Turn off all background health monitoring tests on the supervisor engine and switching modules using the **no diagnostic monitor module** *num* **test all** command.

Diagnostic Sanity Check

You can run the diagnostic sanity check in order to see potential problem areas in your network. The sanity check runs a set of predetermined checks on the configuration with a possible combination of certain system states to compile a list of warning conditions. The checks are designed to look for anything that seems out of place and are intended to serve as an aid for maintaining the system sanity.

To run the diagnostic sanity check, perform this task:

Command	Purpose
show diagnostic sanity	Runs a set of tests on all of the Gigabit Ethernet WAN interfaces in the Cisco 7600 series router.

This example displays samples of the messages that could be displayed with the **show diagnostic sanity** command:

```
Router# show diagnostic sanity
Pinging default gateway 10.6.141.1 ....
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.6.141.1, timeout is 2 seconds:
..!!.
Success rate is 0 percent (0/5)

IGMP snooping disabled please enable it for optimum config.

IGMP snooping disabled but RGMP enabled on the following interfaces,
please enable IGMP for proper config :
Vlan1, Vlan2, GigabitEthernet1/1

Multicast routing is enabled globally but not enabled on the following
interfaces:
GigabitEthernet1/1, GigabitEthernet1/2

A programming algorithm mismatch was found on the device bootflash:
Formatting the device is recommended.

The bootflash: does not have enough free space to accomodate the crashinfo file.

Please check your confreg value : 0x0.

Please check your confreg value on standby: 0x0.

The boot string is empty. Please enter a valid boot string .
Could not verify boot image "disk0:" specified in the boot string on the
slave.

Invalid boot image "bootflash:asdasd" specified in the boot string on the
slave.
```

```
Please check your boot string on the slave.

UDLD has been disabled globally - port-level UDLD sanity checks are
being bypassed.
OR
[
The following ports have UDLD disabled. Please enable UDLD for optimum
config:
Fa9/45

The following ports have an unknown UDLD link state. Please enable UDLD
on both sides of the link:
Fa9/45
]

The following ports have portfast enabled:
Fa9/35, Fa9/45

The following ports have trunk mode set to on:
Fa4/1, Fa4/13

The following trunks have mode set to auto:
Fa4/2, Fa4/3

The following ports with mode set to desirable are not trunking:
Fa4/3, Fa4/4

The following trunk ports have negotiated to half-duplex:
Fa4/3, Fa4/4

The following ports are configured for channel mode on:
Fa4/1, Fa4/2, Fa4/3, Fa4/4

The following ports, not channeling are configured for channel mode
desirable:
Fa4/14

The following vlan(s) have a spanning tree root of 32768:
1

The following vlan(s) have max age on the spanning tree root different from
the default:
1-2

The following vlan(s) have forward delay on the spanning tree root different
from the default:
1-2

The following vlan(s) have hello time on the spanning tree root different
from the default:
1-2

The following vlan(s) have max age on the bridge different from the
default:
1-2

The following vlan(s) have fwd delay on the bridge different from the
default:
1-2

The following vlan(s) have hello time on the bridge different from the
default:
1-2
```

```
The following vlan(s) have a different port priority than the default
on the port FastEthernet4/1
1-2

The following ports have recieve flow control disabled:
Fa9/35, Fa9/45

The following inline power ports have power-deny/faulty status:
Gi7/1, Gi7/2

The following ports have negotiated to half-duplex:
Fa9/45

The following vlans have a duplex mismatch:
Fas 9/45

The following interafaces have a native vlan mismatch:
interface (native vlan - neighbor vlan)
 Fas 9/45 (1 - 64)

The value for Community-Access on read-only operations for SNMP is the same
as default. Please verify that this is the best value from a security point
of view.

The value for Community-Access on write-only operations for SNMP is the same
as default. Please verify that this is the best value from a security point
of view.

The value for Community-Access on read-write operations for SNMP is the same
as default. Please verify that this is the best value from a security point
of view.

Please check the status of the following modules:
8,9

Module 2 had a MINOR_ERROR.

The Module 2 failed the following tests:
TestIngressSpan

The following ports from Module2 failed test1:
1,2,4,48
```

CHAPTER 52

Configuring Web Cache Services Using WCCP

This chapter describes how to configure the Cisco 7600 series routers to redirect traffic to cache engines (web caches) using the Web Cache Communication Protocol (WCCP), and describes how to manage cache engine clusters (cache farms).

Note

- For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

  http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html
- The PFC supports WCCP.
- To use the WCCP Layer 2 PFC redirection feature, configure WCCP on the Cisco 7600 series router as described in this chapter and configure accelerated WCCP on the cache engine as described in the following publication:

  http://www.cisco.com/univercd/cc/td/doc/product/webscale/uce/acns42/cnfg42/transprt.htm#xtocid34
- A cache engine configured for mask assignment that tries to join a farm where the selected assignment method is hash remains out of the farm as long as the cache engine assignment method does not match that of the existing farm.
- With WCCP Layer 2 PFC redirection as the forwarding method for a service group, the packet counters in the **show ip wccp** *service_name* command output displays flow counts instead of packet counts.

This chapter consists of these sections:

Note The tasks in this chapter assume that you have already configured cache engines on your network. For specific information on hardware and network planning associated with Cisco Cache Engines and WCCP, see the Product Literature and Documentation links available on the Cisco.com Web Scaling site.

Understanding WCCP

These sections describe WCCP:

WCCP Overview

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that allows you to integrate cache engines (such as the Cisco Cache Engine 550) into your network infrastructure.

Note Cisco Systems replaced the Cache Engine 500 Series platforms with Content Engine Platforms in July 2001. Cache Engine Products were the Cache Engine 505, 550, 570, and 550-DS3. Content Engine Products are the Content Engine 507, 560, 590, and 7320.

The Cisco IOS WCCP feature allows use of Cisco Cache Engines (or other caches running WCCP) to localize web traffic patterns in the network, enabling content requests to be fulfilled locally. Traffic localization reduces transmission costs and download time.

WCCP enables Cisco IOS routing platforms to transparently redirect content requests. The main benefit of transparent redirection is that users need not configure their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to a cache engine. The word “transparent” is this case means that the end user does not know that a requested file (such as a web page) came from the cache engine instead of from the originally specified server.

When a cache engine receives a request, it attempts to service it from its own local cache. If the requested information is not present, the cache engine issues its own request to the originally targeted server to get the required information. When the cache engine retrieves the requested information, it forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

WCCP enables a series of cache engines, called a *cache engine cluster*, to provide content to a router or multiple routers. Network administrators can easily scale their cache engines to handle heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each cache member to work in parallel, resulting in linear scalability. Clustering cache engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 cache engines to scale to your desired capacity.

Hardware Acceleration

Cisco 7600 series routers provide WCCP Layer 2 PFC redirection hardware acceleration for directly connected Cisco Cache Engines, which is more efficient than Layer 3 redirection in software on the MSFC with generic route encapsulation (GRE).

WCCP Layer 2 PFC redirection allows Cisco Cache Engines to use hardware-supported Layer 2 redirection. A directly connected Cache Engine can be configured to negotiate use of the WCCP Layer 2 PFC Redirection feature. The WCCP Layer 2 PFC redirection feature requires no configuration on the MSFC. The **show ip wccp web-cache detail** command displays which redirection method is in use for each cache.

The following guidelines apply to WCCP Layer 2 PFC redirection:

- The WCCP Layer 2 PFC redirection feature sets the IP flow mask to full-flow mode.
- You can configure the Cisco Cache Engine software release 2.2 or later releases to use the WCCP Layer 2 PFC redirection feature.
- Layer 2 redirection takes place on the PFC and is not visible to the MSFC. The **show ip wccp web-cache detail** command on the MSFC displays statistics for only the first packet of a Layer 2 redirected flow, which provides an indication of how many flows, rather than packets, are using Layer 2 redirection. Entering the **show mls entries** command displays the other packets in the Layer 2 redirected flows.

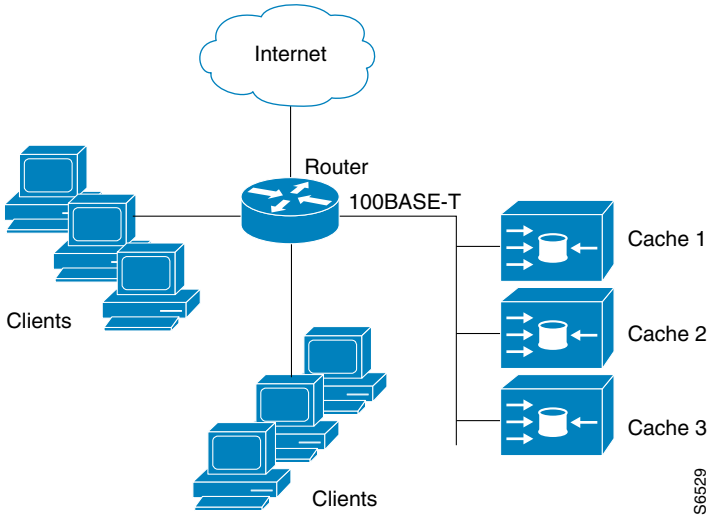
Note

- The PFC provides hardware acceleration for generic route encapsulation (GRE). If you use WCCP Layer 3 redirection with generic route encapsulation (GRE), there is hardware support for encapsulation, but the PFC3 does not provide hardware support for decapsulation of WCCP GRE traffic.
- Releases of Cisco Application and Content Networking System (ACNS) software later than Release 4.2.1 support the **accelerated** keyword.

Understanding WCCPv1 Configuration

With WCCP-Version 1, only a single router services a cluster. In this scenario, this router is the device that performs all the IP packet redirection. Figure 52-1 illustrates how this configuration appears.

Figure 52-1 Cisco Cache Engine Network Configuration Using WCCP-Version 1



Content is not duplicated on the cache engines. The benefit of using multiple caches is that you can scale a caching solution by clustering multiple physical caches to appear as one logical cache.

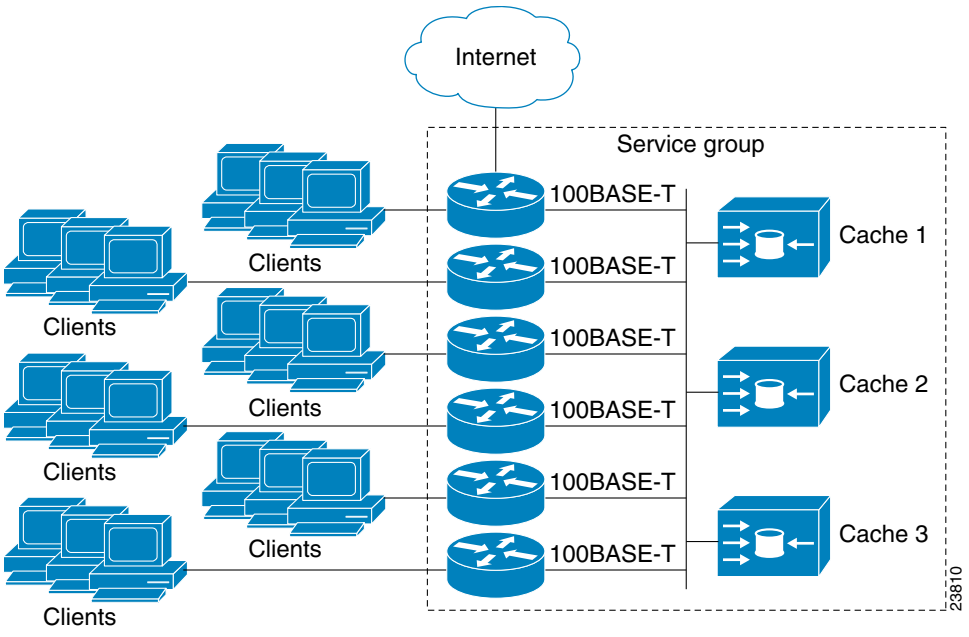
The following sequence of events details how WCCPv1 configuration works:

1. Each cache engine is configured by the system administrator with the IP address of the control router. Up to 32 cache engines can connect to a single control router.
2. The cache engines send their IP addresses to the control router using WCCP, indicating their presence. Routers and cache engines communicate to each other via a control channel; this channel is based on UDP port 2048.
3. This information is used by the control router to create a cluster view (a list of caches in the cluster). This view is sent to each cache in the cluster, essentially making all the cache engines aware of each other. A stable view is established after the membership of the cluster remains the same for a certain amount of time.
4. Once a stable view has been established, one cache engine is elected as the lead cache engine. (The lead is defined as the cache engine seen by all the cache engines in the cluster with the lowest IP address). This lead cache engine uses WCCP to indicate to the control router how IP packet redirection should be performed. Specifically, the lead cache engine designates how redirected traffic should be distributed across the cache engines in the cluster.

Understanding WCCPv2 Configuration

Multiple routers can use WCCPv2 to service a cache cluster. This is in contrast to WCCPv1 in which only one router could redirect content requests to a cluster. Figure 52-2 illustrates a sample configuration using multiple routers.

Figure 52-2 Cisco Cache Engine Network Configuration Using WCCP v2



The subset of cache engines within a cluster and routers connected to the cluster that are running the same service is known as a *service group*. Available services include TCP and User Datagram Protocol (UDP) redirection.

Using WCCPv1, the cache engines were configured with the address of the single router. WCCPv2 requires that each cache engine be aware of all the routers in the service group. To specify the addresses of all the routers in a service group, you must choose one of the following methods:

- Unicast—A list of router addresses for each of the routers in the group is configured on each cache engine. In this case the address of each router in the group must be explicitly specified for each cache engine during configuration.
- Multicast—A single multicast address is configured on each cache engine. In the multicast address method, the cache engine sends a single-address notification that provides coverage for all routers in the service group. For example, a cache engine could indicate that packets should be sent to a multicast address of 224.0.0.100, which would send a multicast packet to all routers in the service group configured for group listening using WCCP (see the **ip wccp group-listen** interface configuration command for details).

The multicast option is easier to configure because you need only specify a single address on each cache engine. This option also allows you to add and remove routers from a service group dynamically, without needing to reconfigure the cache engines with a different list of addresses each time.

The following sequence of events details how WCCPv2 configuration works:

1. Each cache engine is configured with a list of routers.
2. Each cache engine announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of cache engines in the group.
3. Once the view is consistent across all cache engines in the cluster, one cache engine is designated as the lead and sets the policy that the routers need to deploy in redirecting packets.

The following sections describe how to configure WCCPv2 on routers so they may participate in a service group.

WCCPv2 Features

These sections describe WCCPv2 features:

- Support for Non-HTTP Services
- Support for Multiple Routers
- MD5 Security
- Web Cache Packet Return
- Load Distribution

Support for Non-HTTP Services

WCCPv2 allows redirection of traffic other than HTTP (TCP port 80 traffic), including a variety of UDP and TCP traffic. WCCPv1 supported the redirection of HTTP (TCP port 80) traffic only. WCCPv2 supports the redirection of packets intended for other ports, including those used for proxy-web cache handling, File Transfer Protocol (FTP) caching, FTP proxy handling, web caching for ports other than 80, and real audio, video, and telephony applications.

To accommodate the various types of services available, WCCPv2 introduces the concept of multiple *service groups*. Service information is specified in the WCCP configuration commands using dynamic services identification numbers (such as "98") or a predefined service keywords (such as "web-cache"). This information is used to validate that service group members are all using or providing the same service.

The cache engines in service group specify traffic to be redirected by protocol (TCP or UDP) and port (source or destination). Each service group has a priority status assigned to it. Packets are matched against service groups in priority order.

Support for Multiple Routers

WCCPv2 allows multiple routers to be attached to a cluster of cache engines. The use of multiple routers in a service group allows for redundancy, interface aggregation, and distribution of the redirection load.

MD5 Security

WCCPv2 provides optional authentication that enables you to control which routers and cache engines become part of the service group using passwords and the HMAC MD5 standard. Shared-secret MD5 one-time authentication (set using the **ip wccp** [**password** [**0-7**] *password*] global configuration command) enables messages to be protected against interception, inspection, and replay.

Web Cache Packet Return

If a cache engine is unable to provide a requested object it has cached due to error or overload, the cache engine will return the request to the router for onward transmission to the originally specified destination server. WCCPv2 provides a check on packets that determines which requests have been returned from the cache engine unserviced. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the cache cluster). This provides error handling transparency to clients.

Typical reasons why a cache engine would reject packets and initiate the packet return feature include the following:

- Instances when the cache engine is overloaded and has no room to service the packets
- Instances when the cache engine is filtering for certain conditions that make caching packets counterproductive (for example, when IP authentication has been turned on)

Load Distribution

WCCPv2 can be used to adjust the load being offered to individual cache engines to provide an effective use of the available resources while helping to ensure high quality of service (QoS) to the clients. WCCPv2 allows the designated cache to adjust the load on a particular cache and balance the load across the caches in a cluster. WCCPv2 uses three techniques to perform load distribution:

- Hot Spot Handling—Allows an individual hash bucket to be distributed across all the cache engines. Prior to WCCPv2, information from one hash bucket could only go to one cache engine.
- Load Balancing—Allows the set of hash buckets assigned to a cache engine to be adjusted so that the load can be shifted from an overwhelmed cache engine to other members that have available capacity.
- Load Shedding—Enables the router to selectively redirect the load to avoid exceeding the capacity of a cache engine.

By using these hashing parameters, you can prevent one cache from being overloaded and reduce the potential for congestion.

Restrictions for WCCPv2

The following limitations apply to WCCP v2:

- WCCP works only with IP networks.
- For routers servicing a multicast cluster, the time to live (TTL) value must be set at 15 or fewer.
- Because the messages may now be IP multicast, members may receive messages that will not be relevant or are duplicates. Appropriate filtering needs to be performed.
- Service groups can comprise up to 32 cache engines and 32 routers.
- All cache engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Multicast addresses must be from 224.0.0.0 to 239.255.255.255.

Configuring WCCP

The following configuration tasks assume that you have already installed and configured the cache engines you want to include in your network. You must configure the cache engines in the cluster before configuring WCCP functionality on your routers. Refer to the*Cisco Cache Engine Configuration Guide* for cache engine configuration and setup tasks.

IP must be configured on the router interface connected to the cache engines and on the router interface connected to the Internet. Cisco Cache Engines require use of a Fast Ethernet interface for a direct connection. Examples of router configuration tasks follow this section. For complete descriptions of the command syntax, refer to the Release 12.2 *Cisco IOS Configuration Fundamentals Command Reference*.

These sections describe how to configure WCCP:

- Specifying a Version of WCCP, page 52-7 (Optional)
- Configuring a Service Group Using WCCPv2, page 52-8 (Required)
- Excluding Traffic on a Specific Interface from Redirection, page 52-9 (Optional)
- Registering a Router to a Multicast Address, page 52-10 (Optional)
- Using Access Lists for a WCCP Service Group, page 52-10 (Optional)
- Setting a Password for a Router and Cache Engines, page 52-11 (Optional)

Specifying a Version of WCCP

Until you configure a WCCP service using the **ip wccp** {**web-cache** | *service-number*} global configuration command, WCCP is disabled on the router. The first use of a form of the **ip wccp** command enables WCCP. By default WCCPv2 is used for services, but you can use WCCPv1 functionality instead. To change the running version of WCCP from Version 2 to Version 1, or to return to WCCPv2 after an initial change, perform this task in EXEC mode:

Command	Purpose
`Router# ip wccp version {1 | 2}`	Specifies which version of WCCP to configure on a router. WCCPv2 is the default version.

WCCPv1 does not use the WCCP commands from earlier Cisco IOS versions. Instead, use the WCCP commands documented in this chapter. If a function is not allowed in WCCPv1, an error prompt will be printed to the screen. For example, if WCCPv1 is running on the router and you try to configure a dynamic service, the following message will be displayed: "WCCP V1 only supports the web-cache service." The **show ip wccp** EXEC command will display the WCCP protocol version number that is currently running on your router.

Configuring a Service Group Using WCCPv2

WCCPv2 uses service groups based on logical redirection services, deployed for intercepting and redirecting traffic. The standard service is web cache, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the cache engines. This service is referred to as a *well-known service*, because the characteristics of the web cache service are known by both the router and cache engines. A description of a well-known service is not required beyond a service identification (in this case, the command line interface (CLI) provides a **web-cache** keyword in the command syntax).

In addition to the web cache service, there can be up to seven dynamic services running concurrently in a service group.

Note More than one service can run on a router at the same time, and routers and cache devices can be part of multiple service groups at the same time.

The dynamic services are defined by the cache engines; the cache instructs the router which protocol or ports to intercept, and how to distribute the traffic. The router itself does not have information on the characteristics of the dynamic service group's traffic, because this information is provided by the first web cache to join the group. In a dynamic service, up to eight ports can be specified within a single protocol.

Cisco Cache Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other cache devices may use this service number for some other service. The following configuration information deals with enabling general services on Cisco routers. Refer to the cache server documentation for information on configuring services on cache devices.

To enable a service on a Cisco 7600 series router, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip wccp {web-cache | service-number} [accelerated] [group-address groupaddress] [redirect-list access-list] [group-list access-list] [password password]`	Specifies a web cache or dynamic service to enable on the router, specifies the IP multicast address used by the service group, specifies any access lists to use, specifies whether to use MD5 authentication, and enables the WCCP service.

	Command	Purpose
Step 2	`Router(config)# interface type number`	Specifies an interface to configure and enters interface configuration mode.
Step 3	`Router(config-if)# ip wccp {web-cache | service-number} redirect {out | in}`	Enables WCCP redirection on the specified interface.

Note A future release of Cisco Application and Content Networking System (ACNS) software (Release 4.2.2 or later) supports the **ip wccp** *service* **accelerated** command.

As indicated by the **out** and **in** keyword options in the **ip wccp** *service* **redirect** command, redirection can be specified for outbound interfaces or inbound interfaces.

Inbound traffic can be configured to use Cisco Express Forwarding (CEF), distributed Cisco Express Forwarding (dCEF), Fast Forwarding, or Process Forwarding.

Configuring WCCP for redirection for inbound traffic on interfaces allows you to avoid the overhead associated with CEF forwarding for outbound traffic. Setting an output feature on any interface results in the slower switching path of the feature being taken by all packets arriving at all interfaces. Setting an input feature on an interface results in only those packets arriving at that interface taking the configured feature path; packets arriving at other interfaces will use the faster default path. Configuring WCCP for inbound traffic also allows packets to be classified before the routing table lookup, which provides faster redirection of packets.

Specifying a Web Cache Service

To configure a web-cache service, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip wccp web-cache`	Enables the web cache service on the router.
Step 2	`Router(config)# interface type number`	Targets an interface number for which the web cache service will run, and enters interface configuration mode.
Step 3	`Router(config-if)# ip wccp web-cache redirect {out | in}`	Enables the check on packets to determine if they qualify to be redirected to a web cache, using the interface specified in Step 2.

Excluding Traffic on a Specific Interface from Redirection

To exclude any interface from redirecting inbound traffic, perform this task in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies an interface to configure, and enters interface configuration mode.
Step 2	`Router(config-if)# ip wccp redirect exclude in`	Allows inbound packets on this interface to be excluded from redirection.

Registering a Router to a Multicast Address

If you decide to use the multicast address option for your service group, you must configure the router to listen for the multicast broadcasts on an interface. To configure the router, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip wccp {web-cache | service-number} group-address groupaddress`	Specifies the multicast address for the service group.
Step 2	`Router(config)# interface type number`	Specifies the interface to be configured for multicast reception.
Step 3	`Router(config-if)# ip wccp {web-cache | service-number} group-listen`	Enables the reception of IP multicast packets (content originating from the cache engines) on the interface specified in Step 2.

For network configurations where redirected traffic needs to traverse an intervening router, the router being traversed must be configured to perform IP multicast routing. You must configure the following two components to enable traversal over an intervening router:

- Enable IP multicast routing using the **ip multicast routing** interface configuration command.
- Enable the interfaces to which the cache engines will connect to receive multicast transmissions using the **ip wccp group-listen** interface configuration command (note that earlier Cisco IOS versions required the use of the **ip pim** interface configuration command).

Using Access Lists for a WCCP Service Group

To configure the router to use an access list to determine which traffic should be directed to which cache engines, perform this task in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list permit ip host host-address [destination-address | destination-host | any]`	Creates an access list that enables or disables traffic redirection to the cache engine.
Step 2	`Router(config)# ip wccp web-cache group-list access-list`	Indicates to the router from which IP addresses of cache engines to accept packets.

To disable caching for certain clients, perform this task in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list permit ip host host-address [destination-address | destination-host | any]`	Creates an access list that enables or disables traffic redirection to the cache engine.
Step 2	`Router(config)# ip wccp web-cache redirect-list access-list`	Sets the access list used to enable redirection.

Setting a Password for a Router and Cache Engines

MD5 password security requires that each router and cache engine that wants to join a service group be configured with the service group password. The password can consist of up to seven characters. Each cache engine or router in the service group will authenticate the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication will be discarded.

To configure an MD5 password for use by the router in WCCP communications, perform this task in global configuration mode:

Command	Purpose
`Router(config)# ip wccp web-cache password password`	Sets an MD5 password on the router.

Verifying and Monitoring WCCP Configuration Settings

To verify and monitor the configuration settings for WCCP, use the following commands in EXEC mode:

Command	Purpose
`Router# show ip wccp [web-cache | service-number]`	Displays global information related to WCCP, including the protocol version currently running, the number of cache engines in the routers service group, which cache engine group is allowed to connect to the router, and which access list is being used.
`Router# show ip wccp {web-cache | service-number} detail`	Queries the router for information on which cache engines of a specific service group the router has detected. The information can be displayed for either the web cache service or the specified dynamic service.
`Router# show ip interface`	Displays status about whether any **ip wccp** redirection commands are configured on an interface. For example, "Web Cache Redirect is enabled / disabled."
`Router# show ip wccp {web-cache | service-number} view`	Displays which devices in a particular service group have been detected and which cache engines are having trouble becoming visible to all other routers to which the current router is connected. The **view** keyword indicates a list of addresses of the service group. The information can be displayed for either the web cache service or the specified dynamic service. For further troubleshooting information, use the **show ip wccp** {**web-cache** | *service number*} **service** command.

WCCP Configuration Examples

This section provides the following configuration examples:

- Changing the Version of WCCP on a Router Example, page 52-12

Changing the Version of WCCP on a Router Example

The following example shows the process of changing the WCCP version from the default of WCCPv2 to WCCPv1, and enabling the web-cache service in WCCPv1:

```
Router# show ip wccp
% WCCP version 2 is not enabled
Router# configure terminal
Router(config)# ip wccp version 1
Router(config)# end
Router# show ip wccp
% WCCP version 1 is not enabled

Router# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip wccp web-cache
Router(config)# end
Router# show ip wccp
Global WCCP information:
    Router information:
        Router Identifier:                  10.4.9.8
        Protocol Version:                   1.0
. . .
```

Performing a General WCCPv2 Configuration Example

The following example shows a general WCCPv2 configuration session:

```
Router# configure terminal
Router(config)# ip wccp web-cache group-address 224.1.1.100 password alaska1
Router(config)# interface vlan 20
Router(config-if)# ip wccp web-cache redirect out
```

Running a Web Cache Service Example

The following example shows a web cache service configuration session:

```
router# configure terminal
router(config)# ip wccp web-cache
router(config)# interface vlan 20
router(config-if)# ip wccp web-cache redirect out
Router(config-if)# ^Z
Router# copy running-config startup-config
```

The following example shows a configuration session in which redirection of HTTP traffic arriving on VLAN interface 30 is enabled:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface vlan 30
Router(config-if)# ip wccp web-cache redirect in
Router(config-if)# ^Z
Router# show ip interface vlan 30 | include WCCP Redirect
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
```

Running a Reverse Proxy Service Example

The following example assumes you a configuring a service group using Cisco Cache Engines, which use dynamic service 99 to run a reverse proxy service:

```
router# configure terminal
router(config)# ip wccp 99
router(config)# interface vlan 40
router(config-if)# ip wccp 99 redirect out
```

Registering a Router to a Multicast Address Example

The following example shows how to register a router to a multicast address of 224.1.1.100:

```
Router(config)# ip wccp web-cache group-address 224.1.1.100
Router(config)# interface vlan 50
Router(config-if)# ip wccp web cache group-listen
```

The following example shows a router configured to run a reverse proxy service, using the multicast address of 224.1.1.1. Redirection applies to packets outgoing through VLAN interface 60:

```
Router(config)# ip wccp 99 group-address 224.1.1.1
Router(config)# interface vlan 60
Router(config-if)# ip wccp 99 redirect out
```

Using Access Lists Example

To achieve better security, you can use a standard access list to notify the router which IP addresses are valid addresses for a cache engine attempting to register with the current router. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```
router(config)# access-list 10 permit host 11.1.1.1
router(config)# access-list 10 permit host 11.1.1.2
router(config)# access-list 10 permit host 11.1.1.3
router(config)# ip wccp web-cache group-list 10
```

To disable caching for certain clients, servers, or client/server pairs, you can use WCCP access lists. The following example shows that any requests coming from 10.1.1.1 to 12.1.1.1 will bypass the cache and that all other requests will be serviced normally:

```
Router(config)# ip wccp web-cache redirect-list 120
Router(config)# access-list 120 deny tcp host 10.1.1.1 any
Router(config)# access-list 120 deny tcp any host 12.1.1.1
Router(config)# access-list 120 permit ip any any
```

The following example configures a router to redirect web-related packets received through VLAN interface 70, destined to any host except 209.165.196.51:

```
Router(config)# access-list 100 deny ip any host 209.165.196.51
Router(config)# access-list 100 permit ip any any
Router(config)# ip wccp web-cache redirect-list 100
Router(config)# interface vlan 70
Router(config-if)# ip wccp web-cache redirect in
```

Setting a Password for a Router and Cache Engines Example

The following example shows a WCCPv2 password configuration session where the password is alaska1:

```
router# configure terminal
router(config)# ip wccp web-cache password alaska1
```

Verifying WCCP Settings Example

To verify your configuration changes, use the **more system:running-config** EXEC command. The following example shows that the both the web cache service and dynamic service 99 are enabled on the router:

```
router# more system:running-config

    Building configuration...
    Current configuration:
    !
    version 12.0
    service timestamps debug uptime
    service timestamps log uptime
    no service password-encryption
    service udp-small-servers
    service tcp-small-servers
    !
    hostname router4
    !
    enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNTh1
    enable password alabama1
    !
    ip subnet-zero
    ip wccp web-cache
    ip wccp 99
    ip domain-name cisco.com
    ip name-server 10.1.1.1
    ip name-server 10.1.1.2
    ip name-server 10.1.1.3
    !
    !
    !
    interface Vlan200
    ip address 10.3.1.2 255.255.255.0
    no ip directed-broadcast
    ip wccp web-cache redirect out
    ip wccp 99 redirect out
    no ip route-cache
    no ip mroute-cache
    !
```

```
interface Vlan300
ip address 10.4.1.1 255.255.255.0
no ip directed-broadcast
ip wccp 99 redirect out
no ip route-cache
no ip mroute-cache
!
interface Serial0
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
!
line con 0
transport input none
line aux 0
transport input all
line vty 0 4
password alaska1
login
!
end
```

CHAPTER 53

Using the Top N Utility

This chapter describes how to use the Top N utility on the Cisco 7600 series routers.

Note For complete syntax and usage information for the commands used in this chapter, refer to the Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.html

This chapter consists of these sections:

Understanding the Top N Utility

These sections describe the Top N utility:

Top N Utility Overview

The Top N utility allows you to collect and analyze data for each physical port on a router. When the Top N utility starts, it obtains statistics from the appropriate hardware counters and then goes into sleep mode for a user-specified interval. When the interval ends, the utility obtains the current statistics from the same hardware counters, compares the current statistics from the earlier statistics, and stores the difference. The statistics for each port are sorted by one of the statistic types that are listed in Table 53-1.

Table 53-1 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, the Top N utility 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 Utility 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 Utility notifies 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. The Top N Utility displays only those reports that are completed. For reports that are not completed, the Top N Utility displays a short description of the Top N process information.

To terminate a Top N process, enter the **clear top counters interface report** command. Pressing **Ctrl-C** does not terminate Top N 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 the Top N Utility

These sections describe how to use the Top N Utility:

Enabling Top N Utility Report Creation

To enable Top N Utility report 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 Utility report creation.

1. *interface_type* = **ethernet**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**, **port-channel**
2. *statistic_type* = **broadcast**, **bytes**, **errors**, **multicast**, **overflow**, **packets**, **utilization**

When enabling Top N Utility report 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 Utility report 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 the Top N Utility Reports

To display the Top N Utility reports, perform this task:

Command	Purpose
`Router# show top counters interface report [report_num]`	Displays the Top N Utility reports. **Note** To display information about all the reports, do not enter a *report_num* value.

Top N Utility 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 Utility 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 Utility 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 Utility Reports

To clear Top N Utility reports, perform one of these tasks:

Command	Purpose
`Router# clear top counters interface report`	Clears all the Top N Utility reports that have a status of done.
`Router# clear top counters interface report` [*report_num*]	Clears Top N Utility 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 54

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 Cisco 7600 Series Routers Command References at this URL:

http://www.cisco.com/en/US/products/hw/routers/ps368/prod_command_reference_list.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 routers in the path. When the Layer 2 traceroute utility detects a device in the path that does not support Layer 2 traceroute, it continues to send Layer 2 trace queries and allows them to time out.

The Layer 2 traceroute utility can only identify the path from the source device to the destination device. The utility cannot identify the path that a packet takes from the source host to the source device or from the destination device to the destination host.

Usage Guidelines

When using the Layer 2 traceroute utility, follow these guidelines:

- Cisco Discovery Protocol (CDP) must be enabled on all the devices in the network. For the Layer 2 traceroute utility to function properly, do not disable CDP. If any devices in the Layer 2 path are transparent to CDP, the Layer 2 traceroute utility cannot identify these devices on the path.
- A router is defined as reachable from another router 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 router 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 router.
- 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#
```

CHAPTER 55

Configuring Call Home

This chapter describes how to configure the Call Home feature in Cisco IOS Software Release 12.2SX.

Note For complete syntax and usage information for the commands used in this chapter, see the *Cisco 7600 Series Router Cisco IOS Command Reference* at this URL: http://www.cisco.com/en/US/products/ps6922/prod_command_reference_list.html

This chapter includes the following sections:

Understanding Call Home

Call Home provides e-mail-based and web-based notification of critical system events. A versatile range of message formats are available for optimal compatibility with pager services, standard e-mail, or XML-based automated parsing applications. Common uses of this feature may include direct paging of a network support engineer, e-mail notification to a Network Operations Center, XML delivery to a support website, and utilization of Cisco Smart Call Home services for direct case generation with the Cisco Systems Technical Assistance Center (TAC).

The Call Home feature can deliver alert messages containing information on configuration, diagnostics, environmental conditions, inventory, and syslog events.

The Call Home feature can deliver alerts to multiple recipients, referred to as *Call Home destination profiles*, each with configurable message formats and content categories. A predefined destination profile is provided for sending alerts to the Cisco TAC, and you also can define your own destination profiles.

Flexible message delivery and format options make it easy to integrate specific support requirements.

The Call Home feature offers the following advantages:

- Multiple message-format options:

  - Short Text—Suitable for pagers or printed reports.
  - Plain Text—Full formatted message information suitable for human reading.
  - XML—Matching readable format using Extensible Markup Language (XML) and Adaptive Markup Language (AML) document type definitions (DTDs). The XML format enables communication with the Cisco TAC.
- Multiple concurrent message destinations.
- Multiple message categories including configuration, diagnostics, environmental conditions, inventory, and syslog events.
- Filtering of messages by severity and pattern matching.
- Scheduling of periodic message sending.

Obtaining Smart Call Home

If you have a service contract directly with Cisco Systems, you can register your devices for the Smart Call Home service. Smart Call Home provides fast resolution of system problems by analyzing Call Home messages sent from your devices and providing background information and recommendations. For issues that can be identified as known, particularly GOLD diagnostics failures, Automatic Service Requests will be generated with the Cisco TAC.

Smart Call Home offers the following features:

- Continuous device health monitoring and real-time diagnostics alerts.
- Analysis of call home messages from your device and, where appropriate, Automatic Service Request generation, routed to the appropriate TAC team, including detailed diagnostic information to speed problem resolution.
- Secure message transport directly from your device or through a downloadable Transport Gateway (TG) aggregation point. You can use a TG aggregation point in cases requiring support for multiple devices or in cases where security requirements mandate that your devices may not be connected directly to the Internet.
- Web-based access to Call Home messages and recommendations, inventory and configuration information for all Call Home devices. Provides access to associated Field Notices, Security Advisories and End-of-Life Information.

You need the following items to register:

- The SMARTnet contract number for your router.
- Your e-mail address
- Your Cisco.com ID

For detailed information on Smart Call Home, see the Smart Call Home page at this location:

http://www.cisco.com/go/smartcall/

Configuring Call Home

How you configure Call Home depends on how you intend to use the feature. Some information to consider before you configure Call Home includes:

- At least one destination profile (predefined or user-defined) must be configured. The destination profile(s) used depends on whether the receiving entity is a pager, e-mail, or automated service such as Cisco Smart Call Home.
  - If the destination profile uses e-mail message delivery, you must specify a Simple Mail Transfer Protocol (SMTP) server.
  - If the destination profile uses secure HTTP (HTTPS) message transport, you must configure a trustpoint certificate authority (CA).
- The contact e-mail, phone, and street address information should be configured so that the receiver can determine the origin of messages received.
- The router must have IP connectivity to an e-mail server or the destination HTTP server.
- If Cisco Smart Call Home is used, an active service contract must cover the device being configured.

To configure Call Home, follow these steps:

Step 1 Configure your site's contact information.

Step 2 Configure destination profiles for each of your intended recipients.

Step 3 Subscribe each destination profile to one or more alert groups, and set alert options.

Step 4 Configure e-mail settings or HTTPS settings (including CA certificate), depending on the transport method.

Step 5 Enable the Call Home feature.

Step 6 Test Call Home messages.

Tip From the Smart Call Home web application, you can download a basic configuration script to assist you in the configuration of the Call Home feature for use with Smart Call Home and the Cisco TAC. The script will also assist in configuring the trustpoint CA for secure communications with the Smart Call Home service. The script, provided on an as-is basis, can be downloaded from this URL: http://www.cisco.com/go/smartcall/

Configuring Contact Information

Each router must include a contact e-mail address. You can optionally include a phone number, street address, contract ID, customer ID, and site ID.

To assign the contact information, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# call-home`	Enters the Call Home configuration submode.
Step 3	`Router(cfg-call-home)# contact-email-addr` *email-address*	Assigns the customer's e-mail address. Enter up to 200 characters in e-mail address format with no spaces.

	Command	Purpose
Step 4	`Router(cfg-call-home)# phone-number +phone-number`	(Optional) Assigns the customer's phone number. **Note** The number must begin with a plus (**+**) prefix, and may contain only dashes (-) and numbers. Enter up to 16 characters. If you include spaces, you must enclose your entry in quotes (“”).
Step 5	`Router(cfg-call-home)# street-address street-address`	(Optional) Assigns the customer's street address where RMA equipment can be shipped. Enter up to 200 characters. If you include spaces, you must enclose your entry in quotes (“”).
Step 6	`Router(cfg-call-home)# customer-id text`	(Optional) Identifies the customer ID. Enter up to 64 characters. If you include spaces, you must enclose your entry in quotes (“”).
Step 7	`Router(cfg-call-home)# site-id text`	(Optional) Identifies the customer site ID. Enter up to 200 characters. If you include spaces, you must enclose your entry in quotes (“”).
Step 8	`Router(cfg-call-home)# contract-id text`	(Optional) Identifies the customer's contract ID for the router. Enter up to 64 characters. If you include spaces, you must enclose your entry in quotes (“”).

This example shows the configuration of contact information:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# call-home
Router(cfg-call-home)# contact-email-addr username@example.com
Router(cfg-call-home)# phone-number +1-800-555-4567
Router(cfg-call-home)# street-address “1234 Picaboo Street, Any city, Any state, 12345”
Router(cfg-call-home)# customer-id Customer1234
Router(cfg-call-home)# site-id Site1ManhattanNY
Router(cfg-call-home)# contract-id Company1234
Router(cfg-call-home)# exit
Router(config)#
```

Configuring Destination Profiles

A destination profile contains the required delivery information for an alert notification. At least one destination profile is required. You can configure multiple destination profiles of one or more types.

You can use the predefined destination profile or define a desired profile. If you define a new destination profile, you must assign a profile name.

Note If you use the Cisco Smart Call Home service, the destination profile must use the XML message format.

You can configure the following attributes for a destination profile:

- Profile name—A string that uniquely identifies each user-defined destination profile. The profile name is limited to 31 characters and is not case-sensitive. You cannot use **all** as a profile name.

- Transport method—The transport mechanism, either e-mail or HTTP (including HTTPS), for delivery of alerts.
  - For user-defined destination profiles, e-mail is the default, and you can enable either or both transport mechanisms. If you disable both methods, e-mail will be enabled.
  - For the predefined Cisco TAC profile, you can enable either transport method but not both.
- Destination address—The actual address related to the transport method to which the alert should be sent.
- Message formatting—The message format used for sending the alert.
  - The format options for a user-defined destination profile are long-text, short-text, or XML. The default is XML.
  - For the predefined Cisco TAC profile, only XML is allowed.
- Message size—The maximum destination message size. The valid range is 50 to 3,145,728 bytes and the default is 3,145,728 bytes.

To create and configure a destination profile, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# call-home`	Enters the Call Home configuration submode.
Step 3	`Router(cfg-call-home)# profile name`	Enters the Call Home destination profile configuration submode for the specified destination profile. If the specified destination profile does not exist, it is created.
	`Router(cfg-call-home)# no profile name`	Deletes the named user-defined destination profile.
	`Router(cfg-call-home)# no profile all`	Deletes all user-defined destination profiles.
Step 4	`Router(cfg-call-home-profile)# [no] destination transport-method {email | http}`	(Optional) Enables the message transport method. The **no** option disables the method.
Step 5	`Router(cfg-call-home-profile)# destination address {email email-address | http url}`	Configures the destination e-mail address or URL to which Call Home messages will be sent. **Note** When entering a destination URL, include either **http://** or **https://**, depending on whether the server is a secure server. If the destination is a secure server, you must also configure a trustpoint CA.
Step 6	`Router(cfg-call-home-profile)# destination preferred-msg-format {long-text | short-text | xml}`	(Optional) Configures a preferred message format. The default is XML.
Step 7	`Router(cfg-call-home-profile)# destination message-size bytes`	(Optional) Configures a maximum destination message size for the destination profile.
Step 8	`Router(cfg-call-home-profile)# active`	Enables the destination profile. By default, the profile is enabled when it is created.
	`Router(cfg-call-home-profile)# no active`	Disables the destination profile.
Step 9	`Router(cfg-call-home-profile)# exit`	Exits the Call Home destination profile configuration submode and returns to the Call Home configuration submode.

	Command	Purpose
Step 10	`Router(cfg-call-home)# end`	Returns to privileged EXEC mode.
Step 11	`Router# show call-home profile {name | all}`	Displays destination profile configuration for specified profile or all configured profiles.

Copying a Destination Profile

To create a new destination profile by copying an existing profile, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# call-home`	Enters the Call Home configuration submode.
Step 3	`Router(cfg-call-home)# copy profile source-profile target-profile`	Creates a new destination profile with the same configuration settings as the existing destination profile.

Subscribing to Alert Groups

An alert group is a predefined subset of Call Home alerts supported in all routers. Different types of Call Home alerts are grouped into different alert groups depending on their type. These alert groups are available:

- Configuration
- Diagnostic
- Environment
- Inventory
- Syslog

The triggering events for each alert group are listed in the "Alert Group Trigger Events and Commands" section on page 55-15, and the contents of the alert group messages are listed in the "Message Contents" section on page 55-21.

You can select one or more alert groups to be received by a destination profile.

Note A Call Home alert is only sent to destination profiles that have subscribed to the alert group containing that Call Home alert. In addition, the alert group must be enabled.

To subscribe a destination profile to an alert group, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# call-home`	Enters Call Home configuration submode.

	Command	Purpose
Step 3	`Router(cfg-call-home)# alert-group {all|configuration|diagnostic|environment|inventory|syslog}`	Enables the specified alert group. Use the keyword **all** to enable all alert groups. By default, all alert groups are enabled.
	`Router(cfg-call-home)# no alert-group {all|configuration|diagnostic|environment|inventory|syslog}`	Disables the specified alert group. Use the keyword **all** to disable all alert groups.
Step 4	`Router(cfg-call-home)# profile name`	Enters the Call Home destination profile configuration submode for the specified destination profile.
Step 5	`Router(cfg-call-home-profile)# subscribe-to-alert-group configuration [periodic {daily hh:mm | monthly date hh:mm | weekly day hh:mm}]`	Subscribes this destination profile to the Configuration alert group. The Configuration alert group can be configured for periodic notification, as described in the "Configuring Periodic Notification" section on page 55-8.
	`Router(cfg-call-home-profile)# subscribe-to-alert-group all`	Subscribes to all available alert groups.
	`Router(cfg-call-home-profile)# no subscribe-to-alert-group {all|configuration|diagnostic|environment|inventory|syslog}`	Unsubscribes to the specified alert group. Use the keyword **all** to unsubscribe to all alert groups.
Step 6	`Router(cfg-call-home-profile)# subscribe-to-alert-group diagnostic [severity catastrophic|disaster|fatal|critical|major|minor|warning|notification|normal|debugging]`	Subscribes this destination profile to the Diagnostic alert group. The Diagnostic alert group can be configured to filter messages based on severity, as described in the "Configuring Message Severity Threshold" section on page 55-8.
Step 7	`Router(cfg-call-home-profile)# subscribe-to-alert-group environment [severity catastrophic|disaster|fatal|critical|major|minor|warning|notification|normal|debugging]`	Subscribes this destination profile to the Environment alert group. The Environment alert group can be configured to filter messages based on severity, as described in the "Configuring Message Severity Threshold" section on page 55-8.
Step 8	`Router(cfg-call-home-profile)# subscribe-to-alert-group inventory [periodic {daily hh:mm | monthly date hh:mm | weekly day hh:mm}]`	Subscribes this destination profile to the Inventory alert group. The Inventory alert group can be configured for periodic notification, as described in the "Configuring Periodic Notification" section on page 55-8.
Step 9	`Router(cfg-call-home-profile)# subscribe-to-alert-group syslog [severity catastrophic|disaster|fatal|critical|major|minor|warning|notification|normal|debugging] [pattern string]`	Subscribes this destination profile to the Syslog alert group. The Syslog alert group can be configured to filter messages based on severity, as described in the "Configuring Message Severity Threshold" section on page 55-8. You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").
Step 10	`Router(cfg-call-home-profile)# exit`	Exits the Call Home destination profile configuration submode.

Configuring Periodic Notification

When you subscribe a destination profile to either the Configuration or the Inventory alert group, you can choose to receive the alert group messages asynchronously or periodically at a specified time. The sending period can be one of the following:

- Daily—Specify the time of day to send, using an hour:minute format *hh:mm*, with a 24-hour clock (for example, 14:30).
- Weekly—Specify the day of the week and time of day in the format *day hh:mm*, where the day of the week is spelled out (for example, monday).
- Monthly—Specify the numeric date, from 1 to 31, and the time of day, in the format *date hh:mm*.

Configuring Message Severity Threshold

When you subscribe a destination profile to the Diagnostic, Environment, or Syslog alert group, you can set a threshold for the sending of alert group messages based on the message's level of severity. Any message with a value lower than the destination profile's specified threshold is not sent to the destination.

1The severity threshold is configured using the keywords in Table 55-1, and ranges from catastrophic (level 9, highest level of urgency) to debugging (level 0, lowest level of urgency). If no severity threshold is configured, the default is normal (level 1).

Note Call Home severity levels are not the same as system message logging severity levels.

Table 55-1 Severity and Syslog Level Mapping

Level	Keyword	Syslog Level	Description
9	**catastrophic**	N/A	Network-wide catastrophic failure.
8	**disaster**	N/A	Significant network impact.
7	**fatal**	Emergency (0)	System is unusable.
6	**critical**	Alert (1)	Critical conditions, immediate attention needed.
5	**major**	Critical (2)	Major conditions.
4	**minor**	Error (3)	Minor conditions.
3	**warning**	Warning (4)	Warning conditions.
2	**notification**	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	**normal**	Information (6)	Normal event signifying return to normal state.
0	**debugging**	Debug (7)	Debugging messages.

Configuring Syslog Pattern Matching

When you subscribe a destination profile to the Syslog alert group, you can optionally specify a text pattern to be matched within each syslog message. If you configure a pattern, a Syslog alert group message will be sent only if it contains the specified pattern and meets the severity threshold. If the pattern contains spaces, you must enclose it in quotes ("") when configuring it. You can specify up to five patterns for each destination profile.

Configuring General E-Mail Options

To use the e-mail message transport, you must configure at least one Simple Mail Transfer Protocol (SMTP) e-mail server address. You can configure the from and reply-to e-mail addresses, and you can specify up to four backup e-mail servers. You can also set a rate limit on e-mail or HTTP messages.

To configure general e-mail options, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# call-home`	Enters Call Home configuration submode.
Step 3	`Router(cfg-call-home)# mail-server {ipv4-address | name} priority number`	Assigns an e-mail server address and its relative priority among configured e-mail servers. Provide either: • the e-mail server's IP address or • the e-mail server's fully qualified domain *name* (FQDN) of 64 characters or less. Assign a priority *number* between 1 (highest priority) and 100 (lowest priority).
	`Router(cfg-call-home)# no mail-server {ipv4-address | name | all}`	Removes one e-mail server or all e-mail servers from the configuration.
Step 4	`Router(cfg-call-home)# sender from email-address`	(Optional) Assigns the e-mail address that will appear in the from field in Call Home e-mail messages. If no address is specified, the contact e-mail address is used.
Step 5	`Router(cfg-call-home)# sender reply-to email-address`	(Optional) Assigns the e-mail address that will appear in the reply-to field in Call Home e-mail messages.
Step 6	`Router(cfg-call-home)# rate-limit number`	(Optional) Specifies a limit on the number of messages sent per minute, from 1 to 60. The default is 20.

The following notes apply when configuring general e-mail options:

- Backup e-mail servers can be defined by repeating the **mail-server** command using different priority numbers.
- The **mail-server priority** *number* parameter can be configured from 1 to 100. The server with the highest priority (lowest priority number) will be tried first.

This example shows the configuration of general e-mail parameters, including a primary and secondary e-mail server:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# call-home
Router(cfg-call-home)# mail-server smtp.example.com priority 1
Router(cfg-call-home)# mail-server 192.168.0.1 priority 2
Router(cfg-call-home)# sender from username@example.com
Router(cfg-call-home)# sender reply-to username@example.com
Router(cfg-call-home)# exit
Router(config)#
```

Enabling Call Home

To enable or disable the Call Home feature, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode.
Step 2	`Router(config)# service call-home`	Enables the Call Home feature.
	`Router(config)# no service call-home`	Disables the Call Home feature.

Testing Call Home Communications

You can test Call Home communications by sending messages manually using two command types. To send a user-defined Call Home test message, use the **call-home test** command. To send a specific alert group message, use the **call-home send** command.

Sending a Call Home Test Message Manually

To manually send a Call Home test message, perform this task:

	Command	Purpose
Step 1	`Router# call-home test ["test-message"] profile name`	Sends a test message to the specified destination profile. The user-defined test message text is optional, but must be enclosed in quotes ("") if it contains spaces. If no user-defined message is configured, a default message will be sent.

Sending a Call Home Alert Group Message Manually

To manually trigger a Call Home alert group message, perform this task:

	Command	Purpose
Step 1	`Router# call-home send alert-group configuration [profile name]`	Sends a configuration alert group message to one destination profile if specified, or to all subscribed destination profiles.
	`Router# call-home send alert-group diagnostic {module number | slot/subslot | slot/bay_number} [profile name]`	Sends a diagnostic alert group message to the configured destination profile if specified, or to all subscribed destination profiles. You must specify the module or port whose diagnostic information should be sent.
	`Router# call-home send alert-group inventory [profile name]`	Sends an inventory alert group message to one destination profile if specified, or to all subscribed destination profiles.

When manually sending Call Home alert group messages, note the following guidelines:

- Only the configuration, diagnostic, and inventory alert groups can be sent manually.
- When you manually trigger a configuration, diagnostic, or inventory alert group message and you specify a destination profile name, a message is sent to the destination profile regardless of the profile's active status, subscription status, or severity setting.
- When you manually trigger a configuration or inventory alert group message and do not specify a destination profile name, a message is sent to all active profiles that have either a normal or periodic subscription to the specified alert group.
- When you manually trigger a diagnostic alert group message and do not specify a destination profile name, the command will cause the following actions:
  - For any active profile that subscribes to diagnostic events with a severity level of less than minor, a message is sent regardless of whether the module or interface has observed a diagnostic event.
  - For any active profile that subscribes to diagnostic events with a severity level of minor or higher, a message is sent only if the specified module or interface has observed a diagnostic event of at least the subscribed severity level; otherwise, no diagnostic message is sent to the destination profile.

Configuring and Enabling Smart Call Home

For application and configuration information of the Cisco Smart Call Home service, see the "FastStart" section of the *Smart Call Home User Guide* at this location:

http://www.cisco.com/go/smartcall/

The user guide includes configuration examples for sending Smart Call Home messages directly from your device or through a transport gateway (TG) aggregation point. You can use a TG aggregation point in cases requiring support for multiple devices or in cases where security requirements mandate that your devices may not be connected directly to the Internet.

Because the Smart Call Home service uses HTTPS as the transport method, you must also configure its CA as a trustpoint, as described in the *Smart Call Home User Guide*.

Displaying Call Home Configuration Information

To display the configured Call Home information, perform this task:

	Command	Purpose
Step 1	`Router# show call-home`	Displays the Call Home configuration in summary.
	`Router# show call-home detail`	Displays the Call Home configuration in detail.
	`Router# show call-home alert-group`	Displays the available alert groups and their status.
	`Router# show call-home mail-server status`	Checks and displays the availability of the configured e-mail server(s).
	`Router# show call-home profile {all | name}`	Displays the configuration of the specified destination profile. Use the keyword **all** to display the configuration of all destination profiles.
	`Router# show call-home statistics`	Displays the statistics of Call Home events.

Examples 55-2 to 55-8 show the results when using different options of the **show call-home** command.

Example 55-1 Configured Call Home Information

```
Router# show call-home
Current call home settings:
    call home feature : disable
    call home message's from address: switch@example.com
    call home message's reply-to address: support@example.com

    contact person's email address: technical@example.com

    contact person's phone number: +1-408-555-1234
    street address: 1234 Picaboo Street, Any city, Any state, 12345
    customer ID: ExampleCorp
    contract ID: X123456789
    site ID: SantaClara
    Mail-server[1]: Address: smtp.example.com Priority: 1
    Mail-server[2]: Address: 192.168.0.1 Priority: 2
    Rate-limit: 20 message(s) per minute

Available alert groups:
    Keyword                  State   Description
    ------------------------ ------- --------------------------------
    configuration            Disable configuration info
    diagnostic               Disable diagnostic info
    environment              Disable environmental info
    inventory                Enable  inventory info
    syslog                   Disable syslog info

Profiles:
    Profile Name: campus-noc
    Profile Name: CiscoTAC-1

Router#
```

Example 55-2 Configured Call Home Information in Detail

```
Router# show call-home detail
Current call home settings:
    call home feature : disable
    call home message's from address: switch@example.com
    call home message's reply-to address: support@example.com

    contact person's email address: technical@example.com

    contact person's phone number: +1-408-555-1234
    street address: 1234 Picaboo Street, Any city, Any state, 12345
    customer ID: ExampleCorp
    contract ID: X123456789
    site ID: SantaClara
    Mail-server[1]: Address: smtp.example.com Priority: 1
    Mail-server[2]: Address: 192.168.0.1 Priority: 2
    Rate-limit: 20 message(s) per minute

Available alert groups:
    Keyword                  State   Description
    ------------------------ ------- --------------------------------
    configuration            Disable configuration info
    diagnostic               Disable diagnostic info
    environment              Disable environmental info
    inventory                Enable  inventory info
    syslog                   Disable syslog info
```

```
Profiles:

Profile Name: campus-noc
    Profile status: ACTIVE
    Preferred Message Format: long-text
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): noc@example.com
    HTTP  address(es): Not yet set up

    Alert-group               Severity
    ------------------------  ------------
    inventory                 normal

    Syslog-Pattern            Severity
    ------------------------  ------------
    N/A                       N/A

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): callhome@cisco.com
    HTTP  address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

    Periodic configuration info message is scheduled every 1 day of the month at
 09:27

    Periodic inventory info message is scheduled every 1 day of the month at 09:
12

    Alert-group               Severity
    ------------------------  ------------
    diagnostic                minor
    environment               minor

    Syslog-Pattern            Severity
    ------------------------  ------------
    .*                        major

Router#
```

Example 55-3 Available Call Home Alert Groups

```
Router# show call-home alert-group
Available alert groups:
    Keyword                  State   Description
    ------------------------ ------- --------------------------------
    configuration            Disable configuration info
    diagnostic               Disable diagnostic info
    environment              Disable environmental info
    inventory                Enable  inventory info
    syslog                   Disable syslog info

Router#
```

Example 55-4 E-Mail Server Status Information

```
Router# show call-home mail-server status
Please wait. Checking for mail server status ...

Translating "smtp.example.com"
    Mail-server[1]: Address: smtp.example.com Priority: 1 [Not Available]
```

```
    Mail-server[2]: Address: 192.168.0.1 Priority: 2 [Not Available]

Router#
```

Example 55-5 Information for All Destination Profiles (Predefined and User-Defined)

```
Router# show call-home profile all

Profile Name: campus-noc
    Profile status: ACTIVE
    Preferred Message Format: long-text
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): noc@example.com
    HTTP  address(es): Not yet set up

    Alert-group               Severity
    ------------------------  ------------
    inventory                 normal

    Syslog-Pattern            Severity
    ------------------------  ------------
    N/A                       N/A

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): callhome@cisco.com
    HTTP  address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

    Periodic configuration info message is scheduled every 1 day of the month at 09:27

    Periodic inventory info message is scheduled every 1 day of the month at 09:12

    Alert-group               Severity
    ------------------------  ------------
    diagnostic                minor
    environment               minor

    Syslog-Pattern            Severity
    ------------------------  ------------
    .*                        major

Router#
```

Example 55-6 Information for a User-Defined Destination Profile

```
Router# show call-home profile campus-noc

Profile Name: campus-noc
    Profile status: ACTIVE
    Preferred Message Format: long-text
    Message Size Limit: 3145728 Bytes
    Transport Method: email
    Email address(es): noc@example.com
    HTTP  address(es): Not yet set up

    Alert-group               Severity
    ------------------------  ------------
    inventory                 normal
```

```
    Syslog-Pattern            Severity
    ------------------------  ------------
    N/A                       N/A

Router#
```

Example 55-7 Call Home Statistics

```
Router# show call-home statistics
Successful Call-Home Events: 1
Dropped Call-Home Events due to Rate Limiting: 0
Last call-home message sent time: 2007-04-25 11:07:04 GMT+00:00
```

Default Settings

Table 55-2 lists the default Call Home settings.

Table 55-2 Default Call Home Settings

Parameters	Default
Call Home feature status	Disabled
User-defined profile status	Active
Predefined Cisco TAC profile status	Inactive
Transport method	E-mail
Message format type	XML
Destination message size for a message sent in long text, short text, or XML format	3,145,728
Alert group status	Enabled
Call Home message severity threshold	1 (normal)
Message rate limit for messages per minute	20

Alert Group Trigger Events and Commands

Call Home trigger events are grouped into alert groups, with each alert group assigned CLI commands to execute when an event occurs. The CLI command output is included in the transmitted message. Table 55-3 lists the trigger events included in each alert group, including the severity level of each event and the executed CLI commands for the alert group.

Table 55-3 Call Home Alert Groups, Events, and Actions

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
Syslog				Event logged to syslog. (Only sent to TAC if syslog level 0, 1, or 2) CLI commands executed: **show logging**
	SYSLOG	LOG_EMERG	0	System is unusable.
	SYSLOG	LOG_ALERT	1	Action must be taken immediately.
	SYSLOG	LOG_CRIT	2	Critical conditions.
	SYSLOG	LOG_ERR	3	Error conditions.
	SYSLOG	LOG_WARNING	4	Warning conditions.
	SYSLOG	LOG_NOTICE	5	Normal but signification condition.
	SYSLOG	LOG_INFO	6	Informational.
	SYSLOG	LOG_DEBUG	7	Debug-level messages.
	SYSLOG	C2PLUSWITHNODB	2	The module in slot %d has no forwarding daughter board. Power denied.
	SYSLOG	DFCMISMATCH	2	Module %d DFC incompatible with supervisor engine DFC. Power denied.
	SYSLOG	BADFLOWCTRL	2	Module %d not at an appropriate hardware revision level to support DFC. Power denied.
	SYSLOG	BADFLOWCTRL_WARN	2	WARNING: Module %d not at an appropriate hardware revision level to support DFC3.
	SYSLOG	BADPINN1	2	Module %d not at an appropriate hardware revision level to coexist with PFC3 system. Power denied.
	SYSLOG	FANUPGREQ	2	Module %d not supported without fan upgrade.
	SYSLOG	INSUFFCOO	4	Module %d cannot be adequately cooled.
	SYSLOG	PROVISION	6	Module %d does not meet the provisioning requirements, power denied.
	SYSLOG	PWRFAILURE	6	Module %d is being disabled due to power converter failure.
	SYSLOG	LC_FAILURE	3	Module %d has major online diagnostic failure, %s.
	SYSLOG	HARD_RESET	3	Module %d is being hard reset as a part of switchover error recovery.
	SYSLOG	SOFT_RESET	3	Module %d is being soft reset as a part of switchover error recovery.
		DOWNGRADE	6	Fabric-capable module %d not at an appropriate hardware revision level, and can only run in flow-through mode.

Table 55-3 Call Home Alert Groups, Events, and Actions (continued)

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
Environmental				Events related to power, fan, and environment sensing elements, such as temperature alarms. (Sent to TAC.) CLI commands executed: **show environment** **show logging** **show module** **show power**
	FAN_ FAILURE	FANPSINCOMPAT	4	Fan tray and power supply %d are incompatible.
		ALARMCLR	4	The specified alarm condition has been cleared, and shutdown has been cancelled.
	FAN_ FAILURE	FANHIOUTPUT	4	Version %d high-output fan tray is in effect.
	FAN_ FAILURE	FANLOOUTPUT	4	Version %d low-output fan tray is in effect.
	FAN_ FAILURE	FANVERCHK	4	Power supply %d inserted is only compatible with Version %d fan tray.
	FAN_ FAILURE	FANTRAYFAILED	4	Fan tray failed.
	FAN_ FAILURE	FANTRAYOK	4	Fan tray OK.
	FAN_ FAILURE	FANCOUNTFAILED	4	Required number of fan trays is not present.
	FAN_ FAILURE	FANCOUNTOK	4	Required number of fan trays is present.
	FAN_ FAILURE	PSFANFAIL	4	The fan in power supply has failed.
	FAN_ FAILURE	PSFANOK	4	The fan in power supply is OK.
	TEMPERATURE_ALARM	MAJORTEMPALARM	2	Exceeded allowed operating temperature range.
		MAJORTEM-PALARMRECOVER	4	Returned to allowed operating temperature range.
	TEMPERATURE_ALARM	MINORTEMPALARM	4	Exceeded normal operating temperature range.
		MINORTEMPALARM-RECOVER	4	Returned to normal operating temperature range.
	VTT_FAILED	VTTFAILED	4	VTT %d failed.
		VTTOK	4	VTT %d operational.
	VTT_FAILED	VTTMAJFAILED	0	Too many VTT failures to continue system operation.

Table 55-3 Call Home Alert Groups, Events, and Actions (continued)

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
		VTTMAJ RECOVERED	2	Enough VTTs operational to continue system operation.
	CLOCK_FAILED	CLOCKFAILED	4	Clock failed.
		CLOCKOK	4	Clock operational.
	CLOCK_FAILED	CLOCKMAJFAILED	0	Too many clocks failed to continue system operation.
		CLOCKMAJRECOVERED	2	Enough clocks operational to continue system operation.
		SHUTDOWN-SCHEDULED	2	Shutdown for %s scheduled in %d seconds.
		SHUTDOWN_NOT_SCHEDULED	2	Major sensor alarm for %s is ignored, %s will not be shut down.
		SHUTDOWN-CANCELLED	2	Shutdown cancelled.
		SHUTDOWN	2	Shutdown %s now because of %s.
		SHUTDOWN-DISABLED	1	Need to shut down %s now but shutdown action is disabled.
		RESET_SCHEDULED	2	System reset scheduled in seconds.
		CLOCK_SWITCHOVER	2	Changing system switching clock.
		CLOCK_A_MISSING	4	Cannot detect clock A in the system.
		CLOCK_B_MISSING	4	Cannot detect clock B in the system.
		USE_RED_CLOCK	4	System is using the redundant clock (clock B).
		ENABLED	4	Power to module in slot %d set on.
		DISABLED	4	Power to module in slot %d set %s.
		PSOK	4	Power supply %d turned on.
	POWER_SUPPLY_FAILURE	PSFAIL	4	Power supply %d output failed.
		PSREDUNDANTMODE	4	Power supplies set to redundant mode.
		PSCOMBINEDMODE	4	Power supplies set to combined mode.
		PSREDUNDANTMISMATCH	4	Power supplies rated outputs do not match.
		PSMISMATCH	4	Power supplies rated outputs do not match.
		PSNOREDUNDANCY	4	Power supplies are not in full redundancy, power usage exceed lower capacity supply.

Table 55-3 Call Home Alert Groups, Events, and Actions (continued)

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
	POWER_ SUPPLY_ FAILURE	PSOCPSHUTDOWN	2	Power usage exceeds power supply %d allowable capacity.
		PSREDUNDANTONE-SUPPLY	4	In power-redundancy mode, system is operating on one power supply.
		PSREDUNDANT-BOTHSUPPLY	4	In power-redundancy mode, system is operating on both power supplies.
	POWER_ SUPPLY_ FAILURE	UNDERPOWERED	4	Insufficient power to operate all FRUs in system.
	POWER_ SUPPLY_ FAILURE	COULDNOTREPOW-ER	4	Wanted to repower FRU (slot %d) but could not.
	POWER_ SUPPLY_ FAILURE	POWERDENIED	4	Insufficient power, module in slot %d power denied.
		UNSUPPORTED	4	Unsupported module in slot %d, power not allowed: %s.
	POWER_ SUPPLY_ FAILURE	INSUFFICIENT POWER	2	Powering down all line cards as there is not enough power to operate all critical cards.
		INPUTCHANGE	4	Power supply %d input has changed. Power capacity adjusted to %sW.
		PSINPUTDROP	4	Power supply %d input has dropped.
Inventory				Inventory status should be provided whenever a unit is cold-booted, or when FRUs are inserted or removed. This is considered a noncritical event, and the information is used for status and entitlement. (Sent to TAC.) CLI commands executed: **remote command switch show version** **show diagbus** **show idprom all** **show install running (ION only)** **show inventory** **show module** **show version**
	HARDWARE_ INSERTION	INSPS	6	Power supply inserted in slot %d.
	HARDWARE_ REMOVAL	REMPS	6	Power supply removed from slot %d.

Table 55-3 Call Home Alert Groups, Events, and Actions (continued)

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
	HARDWARE_REMOVAL	REMCARD	6	Card removed from slot %d, interfaces disabled.
		STDBY_REMCARD	6	The OIR facility on the standby supervisor engine was notified by the active supervisor engine that a processor from slot[n] has been removed.
	HARDWARE_INSERTION	INSCAR	6	Card inserted in slot %d, interfaces are now online.
		STDBY_INSCARD	6	The standby supervisor engine was notified, card online in slot %d.
		SEQ_MISMATCH	6	SCP sequence mismatch for card in slot %d : %s.
	HARDWARE_REMOVAL	UNKNOWN	3	Unknown card in slot %d, card is being disabled.
		STDBY_UNKNOWN	3	The standby supervisor engine was notified, Unknown card in slot %d.
	HARDWARE_REMOVAL	UNSUPPORTED	3	Card in slot %d is unsupported. %s.
		PWRCYCLE	3	Card in module %d, is being power-cycled %s.
		STDBY_PWRCYCLE	3	The standby supervisor engine was notified, Card in module %d is being power-cycled %s.
		CONSOLE	6	Changing console ownership to %s processor.
		RUNNING_CONFIG	6	During switchover, the OIR facility is unable to clean up running-config processor.
		DISALLOW	6	Supervisor engine attempting to come up as secondary in EHSA mode, will not be allowed.
	HARDWARE_REMOVAL	REMFAN	6	Fan %d removed.
	HARDWARE_INSERTION	INSFAN	6	Fan %d inserted.
	HARDWARE_INSERTION	PSINSERTED	4	Power supply inserted in slot %d.

Table 55-3 Call Home Alert Groups, Events, and Actions (continued)

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
Diagnostic				Events related to standard or intelligent line cards. (Sent to TAC.) CLI commands executed: **remote command switch show version** **show buffers** **show diagnostic result module <slot#> detail** **show diagnostic result module all** **show install running (ION only)** **show inventory** **show logging** **show logging system last 100** **show module** **show version**
		DIAG_OK		
		DIAG_BYPASS		
	DIAGNOSTIC S_FAILURE	DIAG_ERROR		
	DIAGNOSTIC S_FAILURE	DIAG_MINOR_ERRO R		
	DIAGNOSTIC S_FAILURE	DIAG_MAJOR_ERRO R		
Configuration				User-generated request for configuration. (Sent to TAC.) CLI commands executed: **remote command switch show version** **show install running (ION only)** **show module** **show running-config all** **show startup-config** **show version**
Test		TEST		User-generated test message. (Sent to TAC.) CLI commands executed: **show install running (ION only)** **show module** **show version**

Message Contents

The following tables display the content formats of alert group messages:

- Table 55-4 describes the content fields of a short text message.

- Table 55-5 describes the content fields that are common to all long text and XML messages. The fields specific to a particular alert group message are inserted at a point between the common fields. The insertion point is identified in the table.
- Table 55-6 describes the inserted content fields for reactive messages (system failures that require a TAC case) and proactive messages (issues that might result in degraded system performance).
- Table 55-7 describes the inserted content fields for an inventory message.

Table 55-4 Format for a Short Text Message

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to a system message

Table 55-5 Common Fields for All Long Text and XML Messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Time stamp	Date and time stamp of event in ISO time notation: *YYYY-MM-DDTHH:MM:SS*	CallHome/EventTime
Message name	Name of message. Specific event names are listed in the "Alert Group Trigger Events and Commands" section on page 55-15.	(for short text message only)
Message type	Specifically Call Home.	CallHome/Event/Type
Message subtype	Specific type of message: full, delta, or test.	CallHome/Event/SubType
Message group	Specifically reactive or proactive.	(for long text message only)
Severity level	Severity level of message (see Table 55-1 on page 55-8).	Body/Block/Severity
Source ID	Product type for routing. Specifically Catalyst 6500.	(for long text message only)
Device ID	Unique device identifier (UDI) for end device generating message. This field should be empty if the message is nonspecific to a fabric switch. The format is *type@Sid@serial.* • *type* is the product model number from backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number· • *serial* is the number identified by the Sid field. Example: WS-C6509@C@12345678	CallHome/CustomerData/ContractData/DeviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	CallHome/CustomerData/ContractData/CustomerId
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	CallHome/CustomerData/ContractData/ContractId
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	CallHome/CustomerData/ContractData/SiteId

Table 55-5 Common Fields for All Long Text and XML Messages (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Server ID	If the message is generated from the fabric switch, this is the unique device identifier (UDI) of the switch. The format is *type@Sid@serial.* • *type* is the product model number from backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number· • *serial* is the number identified by the Sid field. Example: WS-C6509@C@12345678	(for long text message only)
Message description	Short text describing the error.	CallHome/MessageDescription
Device name	Node that experienced the event. This is the host name of the device.	CallHome/CustomerData/SystemInfo/Name
Contact name	Name of person to contact for issues associated with the node experiencing the event.	CallHome/CustomerData/SystemInfo/Contact
Contact e-mail	E-mail address of person identified as contact for this unit.	CallHome/CustomerData/SystemInfo/ContactEmail
Contact phone number	Phone number of the person identified as the contact for this unit.	CallHome/CustomerData/SystemInfo/ContactPhoneNumber
Street address	Optional field containing street address for RMA part shipments associated with this unit.	CallHome/CustomerData/SystemInfo/StreetAddress
Model name	Model name of the switch. This is the specific model as part of a product family name.	CallHome/Device/Cisco_Chassis/Model
Serial number	Chassis serial number of the unit.	CallHome/Device/Cisco_Chassis/SerialNumber
Chassis part number	Top assembly number of the chassis.	CallHome/Device/Cisco_Chassis/AdditionalInformation/ AD@name="PartNumber"/
System Object ID	The System ObjectID that uniquely identifies the system.	CallHome/Device/Cisco_Chassis/AdditionalInformation/ AD@name="sysObjectID"
SysDesc	System description for the managed element.	CallHome/Device/Cisco_Chassis/AdditionalInformation/ AD@name="sysDescr"
The following fields may be repeated if multiple CLI commands are executed for this alert group.		
Command output name	The exact name of the issued CLI command.	/aml/Attachments/Attachment/Name
Attachment type	Type (usually inline).	/aml/Attachemtents/Attachment@type

Table 55-5 Common Fields for All Long Text and XML Messages (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
MIME type	Normally text/plain or encoding type.	/aml/attachments/attachment/Data@encoding
Command output text	Output of command automatically executed (see Table 55-3 on page 55-16).	/aml/attachments/attachment/atdata

Table 55-6 Fields for a Reactive or Proactive Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	CallHome/Device/Cisco_Chassis/HardwareVersion
Supervisor module software version	Top-level software version.	CallHome/Device/Cisco_Chassis/AdditionalInformation/AD@name="SoftwareVersion"
Affected FRU name	Name of the affected FRU generating the event message.	CallHome/Device/Cisco_Chassis/Cisco_Card/Model
Affected FRU serial number	Serial number of affected FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/SerialNumber
Affected FRU part number	Part number of affected FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/PartNumber
FRU slot	Slot number of FRU generating the event message.	CallHome/Device/Cisco_Chassis/Cisco_Card/LocationWithinContainer
FRU hardware version	Hardware version of affected FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/HardwareVersion
FRU software version	Software version(s) running on affected FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/SoftwareIdentity/VersionString
Process name	Name of process.	/aml/body/process/name
Process ID	Unique process ID.	/aml/body/process/id
Process state	State of process (for example, running or halted).	/aml/body/process/processState
Process exception	Exception or reason code.	/aml/body/process/exception

Table 55-7 Fields for an Inventory Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	CallHome/Device/Cisco_Chassis/HardwareVersion
Supervisor module software version	Top-level software version.	CallHome/Device/Cisco_Chassis/AdditionalInformation/AD@name="SoftwareVersion"
FRU name	Name of the affected FRU generating the event message.	CallHome/Device/Cisco_Chassis/Cisco_Card/Model
FRU s/n	Serial number of FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/SerialNumber
FRU part number	Part number of FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/PartNumber
FRU slot	Slot number of FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/LocationWithinContainer
FRU hardware version	Hardware version of FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/HardwareVersion
FRU software version	Software version(s) running on FRU.	CallHome/Device/Cisco_Chassis/Cisco_Card/SoftwareIdentity/VersionString

Table 55-8 Inserted Fields for a User-Generated Test Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Process ID	Unique process ID.	/aml/body/process/id
Process state	State of process (for example, running or halted).	/aml/body/process/processState
Process exception	Exception or reason code.	/aml/body/process/exception

Sample Syslog Alert Notification in Long-Text Format

```
source:MDS9000
Switch Priority:7
Device Id:WS-C6509@C@FG@07120011
Customer Id:Example.com
Contract Id:123
Site Id:San Jose
Server Id:WS-C6509@C@FG@07120011
Time of Event:2004-10-08T11:10:44
Message Name:SYSLOG_ALERT
```

```
Message Type:Syslog
Severity Level:2
System Name:10.76.100.177
Contact Name:User Name
Contact Email:admin@yourcompany.com
Contact Phone:+1 408 555-1212
Street Address:#1234 Picaboo Street, Any city, Any state, 12345
Event Description:2006 Oct  8 11:10:44 10.76.100.177 %PORT-5-IF_TRUNK_UP: %$VSAN 1%$
Interface fc2/5, vsan 1 is up

syslog_facility:PORT
start chassis information:
Affected Chassis:WS-C6509
Affected Chassis Serial Number:FG@07120011
Affected Chassis Hardware Version:0.104
Affected Chassis Software Version:3.1(1)
Affected Chassis Part No:73-8607-01
end chassis information:
```

Sample Syslog Alert Notification in XML Format

```
From: example
Sent: Wednesday, April 25, 2007 7:20 AM
To: User (user)
Subject: System Notification From Router - syslog - 2007-04-25 14:19:55
GMT+00:00

<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session xmlns:aml-session="http://www.example.com/2004/01/aml-session"
soap-env:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.example.com/services/DDCEService</aml-session:To>
<aml-session:Path>
<aml-session:Via>http://www.example.com/appliance/uri</aml-session:Via>
</aml-session:Path>
<aml-session:From>http://www.example.com/appliance/uri</aml-session:From>
<aml-session:MessageId>M2:69000101:C9D9E20B</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block xmlns:aml-block="http://www.example.com/2004/01/aml-block">
<aml-block:Header>
<aml-block:Type>http://www.example.com/2005/05/callhome/syslog</aml-block:Type>
<aml-block:CreationDate>2007-04-25 14:19:55 GMT+00:00</aml-block:CreationDate>
<aml-block:Builder>
<aml-block:Name>Cat6500</aml-block:Name>
<aml-block:Version>2.0</aml-block:Version>
</aml-block:Builder>
<aml-block:BlockGroup>
<aml-block:GroupId>G3:69000101:C9F9E20C</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>2</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
<ch:CallHome xmlns:ch="http://www.example.com/2005/05/callhome" version="1.0">
```

```
<ch:EventTime>2007-04-25 14:19:55 GMT+00:00</ch:EventTime>
<ch:MessageDescription>03:29:29: %CLEAR-5-COUNTERS: Clear counter on all interfaces by
console</ch:MessageDescription>
<ch:Event>
<ch:Type>syslog</ch:Type>
<ch:SubType></ch:SubType>
<ch:Brand>Cisco Systems</ch:Brand>
<ch:Series>Catalyst 6500 Series Switches</ch:Series>
</ch:Event>
<ch:CustomerData>
<ch:UserData>
<ch:Email>user@example.com</ch:Email>
</ch:UserData>
<ch:ContractData>
<ch:CustomerId>12345</ch:CustomerId>
<ch:SiteId>building 1</ch:SiteId>
<ch:ContractId>abcdefg12345</ch:ContractId>
<ch:DeviceId>WS-C6509@C@69000101</ch:DeviceId>
</ch:ContractData>
<ch:SystemInfo>
<ch:Name>Router</ch:Name>
<ch:Contact></ch:Contact>
<ch:ContactEmail>user@example.com</ch:ContactEmail>
<ch:ContactPhoneNumber>+1 408 555-1212</ch:ContactPhoneNumber>
<ch:StreetAddress>270 E. Tasman Drive, San Jose, CA</ch:StreetAddress>
</ch:SystemInfo>
</ch:CustomerData>
<ch:Device>
<rme:Chassis xmlns:rme="http://www.example.com/rme/4.0">
<rme:Model>WS-C6509</rme:Model>
<rme:HardwareVersion>1.0</rme:HardwareVersion>
<rme:SerialNumber>69000101</rme:SerialNumber>
<rme:AdditionalInformation>
<rme:AD name="PartNumber" value="73-3438-03 01" />
<rme:AD name="SoftwareVersion" value="12.2(20070421:012711)" />
</rme:AdditionalInformation>
</rme:Chassis>
</ch:Device>
</ch:CallHome>
</aml-block:Content>
<aml-block:Attachments>
<aml-block:Attachment type="inline">
<aml-block:Name>show logging</aml-block:Name>
<aml-block:Data encoding="plain">
<![CDATA[
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 53 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 53 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Trap logging: level informational, 72 message lines logged

Log Buffer (8192 bytes):

00:00:54: curr is 0x20000

00:00:54: RP: Currently running ROMMON from F2 region
00:01:05: %SYS-5-CONFIG_I: Configured from memory by console
00:01:09: %SYS-5-RESTART: System restarted --
```

```
Cisco IOS Software, s72033_rp Software (s72033_rp-ADVENTERPRISEK9_DBG-VM), Experimental
Version 12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 15:54 by xxx

Firmware compiled 11-Apr-07 03:34 by integ Build [100]


00:01:01: %PFREDUN-6-ACTIVE: Initializing as ACTIVE processor for this switch

00:01:01: %SYS-3-LOGGER_FLUSHED: System was paused for 00:00:00 to ensure console
debugging output.

00:03:00: SP: SP: Currently running ROMMON from F1 region
00:03:07: %C6K_PLATFORM-SP-4-CONFREG_BREAK_ENABLED: The default factory setting for config
register is 0x2102.It is advisable to retain 1 in 0x2102 as it prevents returning to
ROMMON when break is issued.

00:03:18: %SYS-SP-5-RESTART: System restarted --
Cisco IOS Software, s72033_sp Software (s72033_sp-ADVENTERPRISEK9_DBG-VM), Experimental
Version 12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 18:00 by xxx
00:03:18: %SYS-SP-6-BOOTTIME: Time taken to reboot after reload =  339 seconds
00:03:18: %OIR-SP-6-INSPS: Power supply inserted in slot 1
00:03:18: %C6KPWR-SP-4-PSOK: power supply 1 turned on.
00:03:18: %OIR-SP-6-INSPS: Power supply inserted in slot 2
00:01:09: %SSH-5-ENABLED: SSH 1.99 has been enabled
00:03:18: %C6KPWR-SP-4-PSOK: power supply 2 turned on.
00:03:18: %C6KPWR-SP-4-PSREDUNDANTMISMATCH: power supplies rated outputs do not match.
00:03:18: %C6KPWR-SP-4-PSREDUNDANTBOTHSUPPLY: in power-redundancy mode, system is
operating on both power supplies.
00:01:10: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
00:01:10: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
00:03:20: %C6KENV-SP-4-FANHIOUTPUT: Version 2 high-output fan-tray is in effect
00:03:22: %C6KPWR-SP-4-PSNOREDUNDANCY: Power supplies are not in full redundancy, power
usage exceeds lower capacity supply
00:03:26: %FABRIC-SP-5-FABRIC_MODULE_ACTIVE: The Switch Fabric Module in slot 6 became
active.
00:03:28: %DIAG-SP-6-RUN_MINIMUM: Module 6: Running Minimal Diagnostics...
00:03:50: %DIAG-SP-6-DIAG_OK: Module 6: Passed Online Diagnostics
00:03:50: %OIR-SP-6-INSCARD: Card inserted in slot 6, interfaces are now online
00:03:51: %DIAG-SP-6-RUN_MINIMUM: Module 3: Running Minimal Diagnostics...
00:03:51: %DIAG-SP-6-RUN_MINIMUM: Module 7: Running Minimal Diagnostics...
00:03:51: %DIAG-SP-6-RUN_MINIMUM: Module 9: Running Minimal Diagnostics...
00:01:51: %MFIB_CONST_RP-6-REPLICATION_MODE_CHANGE: Replication Mode Change Detected.
Current system replication mode is Ingress
00:04:01: %DIAG-SP-6-DIAG_OK: Module 3: Passed Online Diagnostics
00:04:01: %OIR-SP-6-DOWNGRADE: Fabric capable module 3 not at an appropriate hardware
revision level, and can only run in flowthrough mode
00:04:02: %OIR-SP-6-INSCARD: Card inserted in slot 3, interfaces are now online
00:04:11: %DIAG-SP-6-DIAG_OK: Module 7: Passed Online Diagnostics
00:04:14: %OIR-SP-6-INSCARD: Card inserted in slot 7, interfaces are now online
00:04:35: %DIAG-SP-6-DIAG_OK: Module 9: Passed Online Diagnostics
00:04:37: %OIR-SP-6-INSCARD: Card inserted in slot 9, interfaces are now online
00:00:09: DaughterBoard (Distributed Forwarding Card 3)

Firmware compiled 11-Apr-07 03:34 by integ Build [100]


00:00:22: %SYS-DFC4-5-RESTART: System restarted --
Cisco IOS Software, c6lc2 Software (c6lc2-SPDBG-VM), Experimental Version
12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
```

```
Compiled Thu 26-Apr-07 17:20 by xxx
00:00:23: DFC4: Currently running ROMMON from F2 region
00:00:25: %SYS-DFC2-5-RESTART: System restarted --
Cisco IOS Software, c6slc Software (c6slc-SPDBG-VM), Experimental Version
12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 16:40 by username1
00:00:26: DFC2: Currently running ROMMON from F2 region
00:04:56: %DIAG-SP-6-RUN_MINIMUM: Module 4: Running Minimal Diagnostics...
00:00:09: DaughterBoard (Distributed Forwarding Card 3)

Firmware compiled 11-Apr-07 03:34 by integ Build [100]

slot_id is 8

00:00:31: %FLASHFS_HES-DFC8-3-BADCARD: /bootflash:: The flash card seems to be corrupted
00:00:31: %SYS-DFC8-5-RESTART: System restarted --
Cisco IOS Software, c6lc2 Software (c6lc2-SPDBG-VM), Experimental Version
12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 17:20 by username1
00:00:31: DFC8: Currently running ROMMON from S (Gold) region
00:04:59: %DIAG-SP-6-RUN_MINIMUM: Module 2: Running Minimal Diagnostics...
00:05:12: %DIAG-SP-6-RUN_MINIMUM: Module 8: Running Minimal Diagnostics...
00:05:13: %DIAG-SP-6-RUN_MINIMUM: Module 1: Running Minimal Diagnostics...
00:00:24: %SYS-DFC1-5-RESTART: System restarted --
Cisco IOS Software, c6slc Software (c6slc-SPDBG-VM), Experimental Version
12.2(20070421:012711)
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Thu 26-Apr-07 16:40 by username1
00:00:25: DFC1: Currently running ROMMON from F2 region
00:05:30: %DIAG-SP-6-DIAG_OK: Module 4: Passed Online Diagnostics
00:05:31: %SPAN-SP-6-SPAN_EGRESS_REPLICATION_MODE_CHANGE: Span Egress HW Replication Mode
Change Detected. Current replication mode for unused asic session 0 is Centralized
00:05:31: %SPAN-SP-6-SPAN_EGRESS_REPLICATION_MODE_CHANGE: Span Egress HW Replication Mode
Change Detected. Current replication mode for unused asic session 1 is Centralized
00:05:31: %OIR-SP-6-INSCARD: Card inserted in slot 4, interfaces are now online
00:06:02: %DIAG-SP-6-DIAG_OK: Module 1: Passed Online Diagnostics
00:06:03: %OIR-SP-6-INSCARD: Card inserted in slot 1, interfaces are now online
00:06:31: %DIAG-SP-6-DIAG_OK: Module 2: Passed Online Diagnostics
00:06:33: %OIR-SP-6-INSCARD: Card inserted in slot 2, interfaces are now online
00:04:30: %XDR-6-XDRIPCNOTIFY: Message not sent to slot 4/0 (4) because of IPC error
timeout. Disabling linecard. (Expected during linecard OIR)
00:06:59: %DIAG-SP-6-DIAG_OK: Module 8: Passed Online Diagnostics
00:06:59: %OIR-SP-6-DOWNGRADE_EARL: Module 8 DFC installed is not identical to system PFC
and will perform at current system operating mode.
00:07:06: %OIR-SP-6-INSCARD: Card inserted in slot 8, interfaces are now online

Router#]]></aml-block:Data>
</aml-block:Attachment>
</aml-block:Attachments>
</aml-block:Block>
</soap-env:Body>
</soap-env:Envelope>
```

CHAPTER 56

Using the Mini Protocol Analyzer

This chapter describes how to use the Mini Protocol Analyzer on the Cisco 7600 series routers. Release 12.2(33)SRD and later releases support the Mini Protocol Analyzer feature.

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

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

This chapter consists of these sections:

Understanding How the Mini Protocol Analyzer Works

The Mini Protocol Analyzer captures network traffic from a SPAN session and stores the captured packets in a local memory buffer. Using the provided filtering options, you can limit the captured packets to:

- Packets from selected VLANs, ACLs, or MAC addresses.
- Packets of a specific EtherType
- Packets of a specified packet size

You can start and stop the capture using immediate commands, or you can schedule the capture to begin at a specified date and time.

The captured data can be displayed on the console, stored to a local file system, or exported to an external server using normal file transfer protocols. The format of the captured file is libpcap, which is supported by many packet analysis and sniffer programs. Details of this format can be found at the following URL:

http://www.tcpdump.org/

By default, only the first 68 bytes of each packet are captured.

Configuring the Mini Protocol Analyzer

To configure a capture session using the Mini Protocol Analyzer, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# [no] monitor session number type capture`	Configures a SPAN session number with packets directed to the processor for capture. Enters capture session configuration mode. The session number range is 1 to 80. The **no** prefix removes the session.
Step 3	`Router(config-mon-capture)# buffer-size buf_size`	(Optional) Sets the size in KB of the capture buffer. The range is 32-65535 KB; the default is 2048 KB.
Step 4	`Router(config-mon-capture)# description session_description`	(Optional) Describes the capture session. The description can be up to 240 characters and cannot contain special characters. If the description contains spaces, it must be enclosed in quotation marks(“”).
Step 5	`Router(config-mon-capture)# rate-limit pps`	(Optional) Sets a limit on the number of packets per second (*pps*) that can be captured. The range is 10-100000 packets per seconds; the default is 10000 packets per second.
Step 6	`Router(config-mon-capture)# source {{interface {single_interface | interface_list | interface_range | mixed_interface_list} | port-channel channel_id}} | {vlan {vlan_ID | vlan_list | vlan_range | mixed_vlan_list}}[rx | tx | both]`	Associates the capture session with source ports or VLANs, and selects the traffic direction to be monitored. The default traffic direction is both.
Step 7	`Router(config-mon-capture)# exit`	Exits the capture session configuration mode.

When configuring a capture session, note the following information:

- Only one capture session is supported; multiple simultaneous capture sessions cannot be configured.
- The **source interface** command argument is either a single interface, or a range of interfaces described by two interface numbers (the lesser one first, separated by a dash), or a comma-separated list of interfaces and ranges.

  Note When configuring a source interface list, you must enter a space before and after the comma. When configuring a source interface range, you must enter a space before and after the dash.

- The **source vlan** command argument is either a single VLAN number from 1 through 4094 (except reserved VLANs), or a range of VLANs described by two VLAN numbers (the lesser one first, separated by a dash), or a list of VLANs and ranges.

  Note When configuring a source VLAN list, do not enter a space before or after the comma. When configuring a source VLAN range, do not enter a space before or after the dash. Note that this requirement differs from the requirement for source interface lists and ranges.

- Data capture does not begin when the capture session is configured. The capture is started by the **monitor capture start** or **monitor capture schedule** command described in the "Starting and Stopping a Capture" section on page 56-4.
- Although the capture buffer is linear by default, it can be made circular as a run-time option in the **monitor capture start** or **monitor capture schedule** command.
- When no hardware rate limit registers are available, the capture session is disabled.
- The source VLAN cannot be changed if a VLAN filter is configured. Remove any VLAN filters before changing the source VLAN.

Filtering the Packets to be Captured

Several options are provided for filtering the packets to be captured. Filtering by ACL and VLAN is performed in hardware before any rate-limiting is applied; all other filters are executed in software. Software filtering can decrease the capture rate.

To filter the packets to be captured by the Mini Protocol Analyzer, perform this task in capture session configuration mode:

	Command	Purpose
Step 1	`Router(config-mon-capture)# [no] filter access-group {acl_number | acl_name}`	(Optional) Captures only packets from the specified ACL.
Step 2	`Router(config-mon-capture)# [no] filter vlan {vlan_ID | vlan_list | vlan_range | mixed_vlan_list}`	(Optional) Captures only packets from the specified source VLAN or VLANs.
Step 3	`Router(config-mon-capture)# [no] filter ethertype type`	(Optional) Captures only packets of the specified EtherType. The *type* can be specified in decimal, hex, or octal.
Step 4	`Router(config-mon-capture)# [no] filter length min_len [max_len]`	(Optional) Captures only packets whose size is between *min_len* and *max_len,* inclusive. If *max_len* is not specified, only packets of exactly size *min_len* will be captured. The range for *min_len* is 0 to 9216 bytes and the range for *max_len* is 1 to 9216 bytes.
Step 5	`Router(config-mon-capture)# [no] filter mac-address mac_addr`	(Optional) Captures only packets from the specified MAC address.
Step 6	`Router(config-mon-capture)# end`	Exits the configuration mode.

When configuring capture filtering, note the following information:

- The **filter vlan** argument is either a single VLAN number from 1 through 4094 (except reserved VLANs), or a range of VLANs described by two VLAN numbers (the lesser one first, separated by a dash), or a list of VLANs and ranges.

Note When configuring a filter VLAN list, you must enter a space before and after the comma. When configuring a filter VLAN range, you must enter a space before and after the dash. Note that this requirement differs from the requirement for source VLAN lists and ranges described in the preceding section.

- To enter an EtherType as a decimal number, enter the number (1 to 65535) with no leading zero. To enter a hexadecimal number, precede four hexadecimal characters with the prefix 0x. To enter an octal number, enter numeric digits (0 to 7) with a leading zero. For example, the 802.1Q EtherType can be entered in decimal notation as 33024, in hexadecimal as 0x8100, or in octal as 0100400.
- Enter a MAC address as three 2-byte values in dotted hexadecimal format. An example is 0123.4567.89ab.
- The **no** keyword removes the filter.

Note After removing a VLAN filter using the **no** keyword, you must exit configuration mode, reenter the capture configuration mode, and issue the **source vlan** command before making other capture configuration changes.

- When you configure a VLAN filter, the capture source or destination must be a VLAN. When you configure a port filter, the capture source or destination must be a port.

Starting and Stopping a Capture

The commands to start and stop a capture are not stored as configuration settings. These commands are executed from the console in EXEC mode. You can start a capture immediately or you can set a future date and time for the capture to start. The capture ends when one of the following conditions occurs:

- A stop or clear command is entered from the console.
- The capture buffer becomes full, unless it is configured as a circular buffer.
- The optionally specified number of seconds has elapsed.
- The optionally specified number of packets has been captured.

When the capture stops, the SPAN session is ended and no further capture session packets are forwarded to the processor.

When starting a packet capture, you have the option to override some configured settings.

To start, stop, or cancel a capture, perform this task:

	Command	Purpose
Step 1	`Router# monitor capture [buffer size buf_size][length cap_len][linear | circular][filter acl_number | acl_name] {start [for count (packets | seconds)] | schedule at time date}`	Starts a capture with optional run-time configuration changes. The capture can start immediately or it can start at a specified time and date. • The **buffer size** option overrides the configured or default capture buffer size. • The **length** option determines the number of bytes that will be captured from each packet. The range for *cap_len* is 0 to 9216 bytes; the default is 68 bytes. A value of 0 causes the entire packet to be captured. • The **circular** option specifies that the capture buffer will overwrite earlier entries once it fills. The **linear** option specifies that the capture will stop when the buffer fills. The default is **linear**. • The **filter** option applies the specified ACL. • The **for** option specifies that the capture will end after the specified number of seconds has elapsed or the specified number of packets has been captured.
Step 2	`Router# monitor capture stop`	Stops the capture.
Step 3	`Router# monitor capture clear [filter]`	Clears any run-time configuration settings, clears any pending scheduled capture, and clears the capture buffer. The **filter** option clears only the run-time filter settings.

When using these commands, note the following information:

- The format for *time* and *date* is hh:mm:ss dd mmm yyyy. The hour is specified in 24-hour notation, and the month is specified by a three-letter abbreviation. For example, to set a capture starting time of 7:30 pm on October 31, 2006, use the notation 19:30:00 31 oct 2006. The time zone is GMT.
- When you specify a capture filter ACL in the start command, the new ACL will not override any configured ACLs. The new ACL will execute in software.

Displaying and Exporting the Capture Buffer

To display the captured packets or information about the capture session, or to export the captured packets for analysis, perform this task:

	Command	Purpose
Step 1	`Router# show monitor capture`	Displays the capture session configuration.
Step 2	`Router# show monitor capture status`	Displays the capture session state, mode, and packet statistics.
Step 3	`Router# show monitor capture buffer [start [end]] [detail][dump [nowrap [dump_length]] [acl acl_number | acl_name]]`	Displays the capture buffer contents. • The *start* and *end* parameters specify packet number indices in the capture buffer. When a *start* index is specified with no *end* index, only the single packet at the *start* index is displayed. When both the *start* and *end* indices are specified, all packets between these indices are displayed. The range is 1 to 4294967295. • The **detail** option adds expanded and formatted protocol and envelope information for each packet, including the packet arrival time. • The **dump** option displays the hexadecimal contents of the packet. If *nowrap* is specified with *dump_length*, one line of hexadecimal packet content of *dump_length* characters will be displayed for each packet. If *dump_length* is not specified, a line of 72 characters will be displayed. The range of *dump_length* is 14 to 256. • The **acl** option causes the display of only those packets that match the specified ACL.
Step 4	`Router# show monitor capture buffer [start [end]] brief [acl acl_number | acl_name]`	Displays only packet header information.
Step 5	`Router# monitor capture export buffer url`	Copies the contents of the capture buffer to the specified file system or file transfer mechanism.

Mini Protocol Analyzer Configuration, Operation, and Display Examples

This section provides examples for configuring the Mini Protocol Analyzer, for starting and stopping a capture session, and for displaying the results of a capture session.

General Configuration Examples

This example shows how to minimally configure the Mini Protocol Analyzer:

```
Router#
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# monitor session 1 type capture
Router(config-mon-capture)# end

Router# show mon cap
Capture instance [1] :
======================
Capture Session ID : 1
Session status     : up
rate-limit value   : 10000
redirect index     : 0x807
buffer-size        : 2097152
capture state      : OFF
capture mode       : Linear
capture length     : 68

Router#
```

This example shows how to configure the buffer size, session description, and rate limit:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# monitor session 1 type capture
Router(config-mon-capture)# buffer-size 4096
Router(config-mon-capture)# description "Capture from ports, no filtering."
Router(config-mon-capture)# rate-limit 20000
Router(config-mon-capture)# end
Router#
Router# show monitor capture
Capture instance [1] :
======================
Capture Session ID : 1
Session status     : up
rate-limit value   : 20000
redirect index     : 0x807
buffer-size        : 4194304
capture state      : OFF
capture mode       : Linear
capture length     : 68

Router#
```

This example shows how to configure the source as a mixed list of ports:

```
Router(config-mon-capture)# source interface gig 3/1 - 3 , gig 3/5
```

This example shows how to configure the source as a mixed list of VLANs:

```
Router(config-mon-capture)# source vlan 123,234-245
```

Filtering Configuration Examples

This example shows how to configure for capturing packets with the following attributes:

- The packets belong to VLANs 123 or 234 through 245
- The packets are of 802.1Q EtherType (hexadecimal 0x8100, decimal 33024)
- The packet size is exactly 8192 bytes
- The source MAC address is 01:23:45:67:89:ab
- The packets conform to ACL number 99

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# monitor session 1 type capture
Router(config-mon-capture)# source vlan 123,234-245
Router(config-mon-capture)# filter ethertype 0x8100
Router(config-mon-capture)# filter length 8192
Router(config-mon-capture)# filter mac-address 0123.4567.89ab
Router(config-mon-capture)# filter access-group 99
Router(config-mon-capture)# end

Router# show monitor capture
Capture instance [1] :
======================
Capture Session ID : 1
Session status     : up
rate-limit value   : 20000
redirect index     : 0x7E07
Capture vlan       : 1019
buffer-size        : 4194304
capture state      : OFF
capture mode       : Linear
capture length     : 68
Sw Filters          :
     ethertype   : 33024
     src mac     : 0123.4567.89ab
  Hw  acl          : 99

Router# show monitor session 1
Session 1
---------
Type                   : Capture Session
Description            : capture from ports
Source VLANs           :
    Both               : 123,234-245
Capture buffer size    : 4096 KB
Capture rate-limit
             value     : 20000
Capture filters        :
     ethertype         : 33024
     src mac           : 0123.4567.89ab
     acl               : 99

Egress SPAN Replication State:
Operational mode       : Centralized
Configured mode        : Distributed (default)
```

```
Router#
```

This example shows how to capture packets whose size is less than 128 bytes:

```
Router(config-mon-capture)# filter length 0 128
```

This example shows how to capture packets whose size is more than 256 bytes:

```
Router(config-mon-capture)# filter length 256 9216
```

Operation Examples

This example shows how to start and stop a capture:

```
Router# monitor capture start
Router# monitor capture stop
Router#
```

This example shows how to start a capture to end after 60 seconds:

```
Router# monitor capture start for 60 seconds
Router#
```

This example shows how to start a capture at a future date and time:

```
Router# monitor capture schedule at 11:22:33 30 jun 2008
capture will start at : <11:22:33 UTC Mon Jun 30 2008>  after 32465825 secs
Router#
```

This example shows how to start a capture with options to override the buffer size and to change to a circular buffer:

```
Router# monitor capture buffer size 65535 circular start
Router#
```

This example shows how to export the capture buffer to an external server and a local disk:

```
Router# monitor capture export buffer tftp://server/user/capture_file.cap
Router# monitor capture export buffer disk1:capture_file.cap
```

Display Examples

These examples show how to display configuration information, session status, and capture buffer contents.

Displaying the Configuration

To display the capture session configuration, enter the **show monitor capture** command.

```
Router# show monitor capture
Capture instance [1] :
=====================
Capture Session ID : 1
Session status     : up
rate-limit value   : 10000
redirect index     : 0x807
buffer-size        : 2097152
```

```
capture state      : OFF
capture mode       : Linear
capture length     : 68
```

This example shows how to display more details using the **show monitor session** *n* command:

```
Router# show monitor session 1
Session 1
---------
Type                   : Capture Session
Source Ports           :
    Both               : Gi3/1-3,Gi3/5
Capture buffer size    : 32 KB
Capture filters        : None

Egress SPAN Replication State:
Operational mode       : Centralized
Configured mode        : Distributed (default)
```

This example shows how to display the full details using the **show monitor session** *n* **detail** command:

```
Router# show monitor session 1 detail
Session 1
---------
Type                   : Capture Session
Description            : -
Source Ports           :
    RX Only            : None
    TX Only            : None
    Both               : Gi3/1-3,Gi3/5
Source VLANs           :
    RX Only            : None
    TX Only            : None
    Both               : None
Source RSPAN VLAN      : None
Destination Ports      : None
Filter VLANs           : None
Dest RSPAN VLAN        : None
Source IP Address      : None
Source IP VRF          : None
Source ERSPAN ID       : None
Destination IP Address : None
Destination IP VRF     : None
Destination ERSPAN ID  : None
Origin IP Address      : None
IP QOS PREC            : 0
IP TTL                 : 255
Capture dst_cpu_id     : 1
Capture vlan           : 0
Capture buffer size    : 32 KB
Capture rate-limit
             value     : 10000
Capture filters        : None

Egress SPAN Replication State:
Operational mode       : Centralized
Configured mode        : Distributed (default)
```

Displaying the Capture Session Status

To display the capture session status, enter the **show monitor capture status** command.

```
Router# show monitor capture status
capture state      : ON
capture mode       : Linear
Number  of packets
          received : 253
           dropped : 0
          captured : 90
```

Displaying the Capture Buffer Contents

To display the capture session contents, enter the **show monitor capture buffer** command. These examples show the resulting display using several options of this command:

```
Router# show monitor capture buffer
 1     IP: s=10.12.0.5 , d=224.0.0.10, len 60
 2     346   0180.c200.000e  0012.44d8.5000  88CC 020707526F7
 3     60    0180.c200.0000  0004.c099.06c5  0026 42420300000
 4     60    ffff.ffff.ffff  0012.44d8.5000  0806 00010800060
 5     IP: s=7.0.84.23 , d=224.0.0.5, len 116
 6     IP: s=10.12.0.1 , d=224.0.0.10, len 60


Router# show monitor capture buffer detail
 1      Arrival time : 09:44:30 UTC Fri Nov 17 2006
        Packet Length : 74 , Capture Length : 68
        Ethernet II :  0100.5e00.000a  0008.a4c8.c038  0800
        IP: s=10.12.0.5 , d=224.0.0.10, len 60, proto=88
 2      Arrival time : 09:44:31 UTC Fri Nov 17 2006
        Packet Length : 346 , Capture Length : 68
346   0180.c200.000e  0012.44d8.5000  88CC 020707526F757463031


Router# show monitor capture buffer dump
 1     IP: s=10.12.0.5 , d=224.0.0.10, len 60
08063810:                       0100 5E00000A            ..^...
08063820: 0008A4C8 C0380800 45C0003C 00000000  ..$H@8..E@.<....
08063830: 0258CD8F 0A0C0005 E000000A 0205EE6A  .XM.....`.....nj
08063840: 00000000 00000000 00000000 00000064  ...............d
08063850: 0001000C 01000100 0000000F 0004      ..............
 2     346   0180.c200.000e  0012.44d8.5000  88CC 020707526F757465720415
 3     60    0180.c200.0000  0004.c099.06c5  0026 4242030000000000800000
 4     60    ffff.ffff.ffff  0012.44d8.5000  0806 0001080006040001001244
 5     IP: s=7.0.84.23 , d=224.0.0.5, len 116
0806FCB0:                       0100 5E000005            ..^...
0806FCC0: 0015C7D7 AC000800 45C00074 00000000  ..GW,...E@.t....
0806FCD0: 01597D55 07005417 E0000005 0201002C  .Y}U..T.`......,
0806FCE0: 04040404 00000000 00000002 00000010  ................
0806FCF0: 455D8A10 FFFF0000 000A1201 0000      E]............


Router# show monitor capture buffer dump nowrap
 1     74    0100.5e00.000a  0008.a4c8.c038  0800 45C0003C000000
 2     346   0180.c200.000e  0012.44d8.5000  88CC 020707526F7574
 3     60    0180.c200.0000  0004.c099.06c5  0026 42420300000000
 4     60    ffff.ffff.ffff  0012.44d8.5000  0806 00010800060400
```

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

The global health monitoring tests consist of the following tests:

TestSPRPInbandPing

The TestSPRPInbandPing 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(33)SRA and later releases.
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-2 TestScratchRegister Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not disable.
Default	On.
Release	12.2(33)SRA and later releases.
Corrective action	Reset the malfunctioning supervisor engine or power down the module.
Hardware support	Supervisor Engine 720, DFC-equipped modules, WS-X6148-FE-SFP, WS-X6148A-GE-TX, and WS-X6148A-RJ-45.

TestMacNotification

The TestMacNotification test verifies that the data and control path between DFC modules and supervisor engines is working properly. This test also ensures Layer 2 MAC address consistency across Layer 2 MAC address tables. The test runs every six seconds. Ten consecutive failures causes the module to reset during bootup or runtime (default). After three consecutive resets, the module powers down.

Table A-3 TestMacNotification Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not disable.
Default	On.
Release	12.2(33)SRA and later releases.
Corrective action	Reset the module. After the module has ten consecutive failures or three consecutive resets, it powers down.
Hardware support	DFC-equipped modules.

Per-Port Tests

The per-port tests consist of the following tests:

TestNonDisruptiveLoopback, page A-C

TestLoopback, page A-D

TestActiveToStandbyLoopback, page A-D

TestTransceiverIntegrity, page A-E

TestNetflowInlineRewrite, page A-E

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.
Release	12.2(33)SRA and later releases.

Table A-4 TestNonDisruptiveLoopback Test Attributes (continued)

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(33)SRA and later releases.
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.).
Recommendation	Schedule during downtime.
Default	Runs at bootup or after OIR.

Table A-6 TestActiveToStandbyLoopback Test Attributes (continued)

Release	12.2(33)SRA and later releases.
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(33)SRA and later releases.
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(33)SRA and later releases.
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 is a combination of the TestNewLearn and the TestIndexLearn tests, which are described in the "DFC Layer 2 Forwarding Engine Tests" section on page A-H.

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(33)SRA and later releases.
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, which are described in the "DFC Layer 2 Forwarding Engine Tests" section on page A-H.

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(33)SRA and later releases.
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, which are described in the "DFC Layer 2 Forwarding Engine Tests" section on page A-H.

Table A-11 TestBadBpduTrap 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(33)SRA and later releases.
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, which are described in the "DFC Layer 2 Forwarding Engine Tests" section on page A-H.

Table A-12 TestMatchCapture Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Run this test on-demand to verify the Layer 2 learning functionality. This test can also be used as a health-monitoring test.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(33)SRA and later releases.
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	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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

DFC Layer 2 Forwarding Engine Tests

The DFC Layer 2 Forwarding Engine tests consists of the following tests:

TestDontLearn, page A-I

TestNewLearn, page A-I

TestIndexLearn, page A-J

TestConditionalLearn, page A-J

TestTrap, page A-K

TestBadBpdu, page A-K

TestProtocolMatchChannel, page A-L

TestCapture, page A-L

TestStaticEntry, page A-M

TestDontLearn

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. For DFC-enabled modules, the diagnostic packet is sent from the supervisor engine inband port through the switch fabric and looped back from one of the ports on the DFC-enabled module. The “don't learn” feature is verified during diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-14 TestDontLearn 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	Schedule during downtime.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

TestNewLearn

The TestNewLearn test verifies the Layer 2 source MAC address learning functionality of the Layer 2 forwarding engine. For supervisor engines, a diagnostic packet is sent from the supervisor engine inband port to verify that the Layer 2 forwarding engine is learning the new source MAC address from the diagnostic packet. For DFC-enabled modules, a diagnostic packet is sent from the supervisor engine inband port through the switch fabric and looped backed from one of the ports on the DFC-enabled module. The Layer 2 learning functionality is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-15 TestNewLearn 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

TestIndexLearn

The TestIndexLearn test ensures that existing MAC address table entries can be updated. This test verifies the Index Learn feature of the Layer 2 forwarding engine is working properly. When running the test on the supervisor engine, the diagnostic packet is sent from the supervisor engine's inband port and performs a packet lookup using the supervisor engine Layer 2 forwarding engine. For DFC-enabled modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Index Learn feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-16 TestIndexLearn 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

TestConditionalLearn

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. For DFC-enabled modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Conditional Learn feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-17 TestConditionalLearn 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

TestTrap

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. For DFC-enabled modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Trap feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-18 TestTrap 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

TestBadBpdu

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. For DFC-enabled modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The BPDU feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-19 TestBadBpdu 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

TestProtocolMatchChannel

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. For DFC-enabled modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. T he Match feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-20 TestProtocolMatchChannel 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

TestCapture

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. For DFC-enabled modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Capture feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-21 TestCapture 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	Schedule during downtime.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

TestStaticEntry

The TestStaticEntry test verifies the ability to populate static entries in the Layer 2 MAC address table. When running the test on the supervisor engine, the diagnostic packet is sent from the supervisor engine's inband port and performs a packet lookup using the supervisor engine's Layer 2 forwarding engine. For DFC-enabled modules, the diagnostic packet is sent from the supervisor engine's inband port through the switch fabric and looped back from one of the DFC ports. The Static Entry feature is verified during the diagnostic packet lookup by the Layer 2 forwarding engine.

Table A-22 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	DFC-enabled modules.

PFC Layer 3 Forwarding Engine Tests

The PFC Layer 3 Forwarding Engine tests consists of the following tests:

TestFibDevices, page A-N
TestIPv4FibShortcut, page A-N
TestIPv6FibShortcut, page A-O
TestMPLSFibShortcut, page A-O
TestNATFibShortcut, page A-P
TestL3Capture2, page A-P
TestAclPermit, page A-Q
TestAclDeny, page A-Q
TestQoS, page A-R

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-23 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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-24 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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-25 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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-26 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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-27 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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-28 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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-29 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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-30 TestACLDeny Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not disable.
Default	On.
Release	12.2(33)SRA and later releases.
Corrective action	Automatic ASIC reset for recovery.
Hardware support	Supervisor engines and DFC-enabled modules.

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-31 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(33)SRA and later releases.
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-32 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

DFC Layer 3 Forwarding Engine Tests

The DFC Layer 3 Forwarding Engine tests consists of the following tests:

TestFibDevices, page A-S

TestIPv4FibShortcut, page A-S

TestIPv6FibShortcut, page A-T

TestMPLSFibShortcut, page A-T

TestNATFibShortcut, page A-U

TestL3Capture2, page A-U

TestFibDevices

The TestFibDevices test verifies that the FIB TCAM and adjacency devices are functional. One FIB entry is installed on each FIBTCAM device and 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, the TestFibDevices test tests all FIB and adjacency devices using IPv4 or IPv6 packets, depending on your configuration.

Table A-33 TestFibDevices 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	Schedule during downtime.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

TestIPv4FibShortcut

The TestIPv4FibShortcut test verifies that the IPv4 FIB forwarding functionality 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-34 TestIPv4FibShortcut 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.

Table A-34 TestIPv4FibShortcut Test Attributes (continued)

Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

TestIPv6FibShortcut

The TestIPv6FibShortcut test verifies that the IPv6 FIB forwarding functionality 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 that the diagnostic packet is forwarded according to rewritten MAC and VLAN information.

Table A-35 TestIPv6FibShortcut 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

TestMPLSFibShortcut

The TestMPLSFibShortcut test verifies that the MPLS forwarding functionality 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 using the MPLS label from the adjacency entry.

Table A-36 TestMPLSFibShortcut 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

TestNATFibShortcut

The TestNATFibShortcut test verifies the ability to rewrite a packet based on NAT adjacency information, such as the rewrite destination IP address. One diagnostic NAT FIB and adjacency entry is installed and a diagnostic packet is sent to make sure the diagnostic packet is forwarded according to the rewritten IP address.

Table A-37 TestNATFibShortcut 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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 capture bit information.

Table A-38 TestL3Capture2 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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 is looked up by the Layer 3 forwarding engine to make sure it hits the ACL TCAM entry and gets permitted and forwarded correctly.

Table A-39 TestACLPermit 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	This test runs by default during bootup or after a reset or OIR.
Default	Off.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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 and output Layer 2 redirect, Layer 3 redirect, and Layer 3 bridges.

Table A-40 TestACLDeny 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 the looped-back port (for example, Spanning Tree Protocol).
Recommendation	Schedule during downtime if you are using ACLs.
Default	Off.
Release	12.2(33)SRA and later releases.
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-41 TestQoS Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive for looped-back ports. The disruption is typically less than one second.
Recommendation	Schedule during downtime.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

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-42 TestNetflowShortcut Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive for looped-back ports. Disruption is typically less than one second.
Recommendation	Run this test on-demand if you suspect that NetFlow is not working properly.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and DFC-enabled modules.

Replication Engine Tests

The Replication Engine tests consists of the following tests:

TestL3VlanMet, page A-X

TestIngressSpan, page A-X

TestEgressSpan, page A-Y

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-43 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(33)SRA and later releases.
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-44 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(33)SRA and later releases.
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-45 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	Supervisor engines and WS-65xx and WS-67xx modules.

Fabric Tests

The Fabric tests consists of the following tests:

TestFabricSnakeForward, page A-Y
TestFabricSnakeBackward, page A-Z
TestSynchedFabChannel, page A-Z
TestFabricCh0Health, page A-AA
TestFabricCh1Health, page A-AA

TestFabricSnakeForward

The TestFabricSnakeForward test consists of two test cases: the internal snake test and the external snake test. The internal snake test generates the test packets inside the fabric ASIC and the test data path is limited so that it stays inside the fabric ASIC. The external snake test generates the test packet using the supervisor engine inband port; the test data path involves the port ASIC, the rewrite engine ASIC inside the supervisor engine, and the fabric ASIC. Whether or not the supervisor engine local channel is synchronized to the fabric ASIC determines which test is used. If it is synchronized, the external snake test is used; if it is not, the internal snake test is used. For both tests, only the channels that are not synchronized to any modules are involved in the test. The Forward direction indicates that the snaking direction is from the low-numbered channel to the high-numbered channel.

Table A-46 TestFabricSnakeForward Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Run on-demand. This test can result in high CPU utilization.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(33)SRA and later releases.

Table A-46 TestFabricSnakeForward Test Attributes (continued)

Corrective action	Supervisor engines crash to ROMMON; SFMs reset.
Hardware support	Supervisor Engine 720 and SFM.

TestFabricSnakeBackward

The TestFabricSnakeBackward test consists of two test cases: the internal snake test and the external snake test. The internal snake test generates the test packets inside the fabric ASIC, and the test data path is limited so that it stays inside the fabric ASIC. The external snake test generates the test packet using the supervisor engine inband port and the test data path involves the port ASIC, the rewrite engine ASIC inside the supervisor engine, and the fabric ASIC. Whether or not the supervisor engine local channel is synchronized to the fabric ASIC determines which test is used. If it is synchronized, the external snake test is used; if it is not, internal snake test is used. For both tests, only the channels that are not synchronized to any modules are involved in the test. The backward direction indicates that the snaking direction is from the high-numbered channel to the low-numbered channel.

Table A-47 TestFabricSnakeBackward Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Run on-demand. This test can result in high CPU utilization.
Default	This test runs by default during bootup or after a reset or OIR.
Release	12.2(33)SRA and later releases.
Corrective action	Supervisor engines crash to ROMMON; SFMs reset.
Hardware support	Supervisor Engine 720 and SFM.

TestSynchedFabChannel

The TestSynchedFabChannel test periodically checks the fabric synchronization status for both the module and the fabric. This test is available only for fabric-enabled modules. This test is not a packet-switching test so it does not involve the data path. This test sends an SCP control message to the module and fabric to query the synchronization status.

Table A-48 TestSynchedFabChannel Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not turn this test off. Use as a health-monitoring test.
Default	On.
Release	12.2(33)SRA and later releases.
Corrective action	The module resets after five consecutive failures. Three consecutive reset cycles results in the module powering down. A fabric switchover may be triggered, depending on the type of failure.
Hardware support	All fabric-enabled modules.

TestFabricCh0Health

The TestFabricCh0Health test constantly monitors the health of the ingress and egress data paths for fabric channel 0 on 10-gigabit modules. The test runs every five seconds. Ten consecutive failures are treated as fatal and the module resets; three consecutive reset cycles may result in a fabric switchover.

Table A-49 TestFabricSCh0Health Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not turn this test off. Use as a health-monitoring test.
Default	On.
Release	12.2(33)SRA and later releases.
Corrective action	The module resets after 10 consecutive failures. Three consecutive resets powers down the module.
Hardware support	WS-X6704-10GE and WS-6702-10GE.

TestFabricCh1Health

The TestFabricCh1Health test constantly monitors the health of the ingress and egress data paths for fabric channel 1 on 10-gigabit modules. The test runs every five seconds. Ten consecutive failures are treated as fatal and the module resets; three consecutive reset cycles may result in a fabric switchover.

Table A-50 TestFabricCh1Health Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not turn this test off. Use as a health-monitoring test.
Default	On.
Release	12.2(33)SRA and later releases.
Corrective action	The module resets after 10 consecutive failures. Three consecutive failures resets powers down the module.
Hardware support	WS-X6704-10GE module.

Exhaustive Memory Tests

The exhaustive memory tests include the following tests:

TestFibTcamSSRAM, page A-AB

TestAsicMemory, page A-AB

TestAclQosTcam, page A-AC

TestNetflowTcam, page A-AC

TestQoSTcam, page A-AD

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-51 TestFibTcamSSRAM Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive. Disruption is several hours.
Recommendation	Use this test only if you suspect a problem with the hardware or before putting the hardware into a live network. Do not run any traffic in the background on the module that you are testing. The supervisor engine must be rebooted after running this test.
Default	Off.
Release	12.2(33)SRA and later releases.
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-52 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(33)SRA and later releases.
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, PFC3BXL, PFC3C, and PFC3CXL. It is not supported on the PFC3A.

Table A-53 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.
Default	Off.
Release	12.2(33)SRA and later releases.
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-54 TestNetflowTcam Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive. Disruption is several minutes and can vary depending on the type of PFC you are testing.
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(33)SRA and later releases.
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-55 TestQoSTcam Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Disruptive. Disruption is several minutes and can vary depending on what type of PFC you are testing.
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(33)SRA and later releases.
Corrective action	Not applicable.
Hardware support	All modules including supervisor engines.

IPSEC Services Modules Tests

The IPSEC Services Modules Tests include the following tests:

TestIPSecClearPkt, page A-AD

TestHapiEchoPkt, page A-AE

TestIPSecEncryptDecryptPkt, page A-AE

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-56 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(33)SRA and later releases.
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-57 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(33)SRA and later releases.
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-58 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(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	VPN services module.

Stress Tests

The stress tests consist of the following tests:

TestTrafficStress, page A-AF

TestEobcStressPing, page A-AF

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-59 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(33)SRA and later releases.
Corrective action	Not applicable.
Hardware support	Supervisor Engine 720 and Supervisor Engine 32.

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-60 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(33)SRA and later releases.
Corrective action	Not applicable.
Hardware support	Supervisor Engine 720 and Supervisor Engine 32.

Critical Recovery Tests

The critical recovery tests consist of the following tests:

- TestL3HealthMonitoring, page A-AG
- TestTxPathMonitoring, page A-AG
- TestSynchedFabChannel, page A-AH

The TestFabricCh0Health and TestFabricCh1Health tests are also considered critical recovery tests. See the "Fabric Tests" section on page A-Y for a description of these tests.

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-61 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(33)SRA and later releases.
Corrective action	Not applicable.
Hardware support	DFC-equipped modules

TestTxPathMonitoring

The TestTxPathMonitoring test sends index-directed packets periodically to each port on the Supervisor Engine 720 and WS-X67xx series modules to verify ASIC synchronization and correct any related problems. The test runs every two seconds.

Table A-62 TestTxPathMonitoring Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not change the default settings.
Default	On.
Release	12.2(33)SRA and later releases.
Corrective action	Not applicable (self-recovering).
Hardware support	Supervisor Engine 720 and WS-67xx series modules.

TestSynchedFabChannel

The TestSynchedFabChannel test periodically checks the fabric synchronization status for both the module and the fabric. This test is available only for fabric-enabled modules. This test is not a packet-switching test so it does not involve the data path. This test sends an SCP control message to the module and fabric to query the synchronization status.

Table A-63 TestSynchedFabChannel Test Attributes

Attribute	Description
Disruptive/Nondisruptive	Nondisruptive.
Recommendation	Do not turn off. Use as a health-monitoring test.
Default	On.
Release	12.2(33)SRA and later releases.
Corrective action	None. See the system message guide for more information.
Hardware support	All fabric-enabled modules.

General Tests

The general tests consist of the following tests:

ScheduleSwitchover, page A-AH

TestFirmwareDiagStatus, page A-AI

ScheduleSwitchover

The ScheduleSwitchover test allows you to trigger a switchover at any time using the online diagnostics scheduling capability.

Table A-64 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(33)SRA and later releases.
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-65 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(33)SRA and later releases.
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
MSFC	Multilayer Switch Feature Card
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
MSFC	Multilayer Switching Feature Card
MSM	Multilayer Switch Module
MST	multiple spanning tree
MTU	maximum transmission unit
MVAP	multiple VLAN access port

Table B-1 List of Acronyms (continued)

Acronym	Expansion
NAM	Network Analysis Module
NBP	Name Binding Protocol
NCIA	Native Client Interface Architecture
NDE	NetFlow Data Export
NET	network entity title
NetBIOS	Network Basic Input/Output System
NFFC	NetFlow Feature Card
NMP	Network Management Processor
NSAP	network service access point
NSF	Nonstop Forwarding
NTP	Network Time Protocol
NVRAM	nonvolatile RAM
OAM	Operation, Administration, and Maintenance
ODM	order dependent merge
OSI	Open System Interconnection
OSM	Optical Services Module
OSPF	open shortest path first
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
PPP	Point-to-Point Protocol
PRID	Policy Rule Identifiers
PVST+	Per VLAN Spanning Tree+
QDM	QoS device manager
QM	QoS manager

Table B-1 List of Acronyms (continued)

Acronym	Expansion
QoS	quality of service
RACL	router interface access control list
RADIUS	Remote Access Dial-In User Service
RAM	random-access memory
RCP	Remote Copy Protocol
RGMP	Router-Ports Group Management Protocol
RIB	routing information base
RIF	Routing Information Field
RMON	remote network monitor
ROM	read-only memory
ROMMON	ROM monitor
RP	route processor or rendezvous point
RPC	remote procedure call
RPF	reverse path forwarding
RPR	route processor redundancy
RPR+	route processor redundancy plus
RSPAN	remote SPAN
RST	reset
RSVP	ReSerVation Protocol
SAID	Security Association Identifier
SAP	service access point
SCM	service connection manager
SCP	Switch-Module Configuration Protocol
SDLC	Synchronous Data Link Control
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

Table B-1 List of Acronyms (continued)

Acronym	Expansion
SRM	single router mode
SSO	stateful switchover
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

Table B-1 List of Acronyms (continued)

Acronym	Expansion
WCCP	Web Cache Communications Protocol
WFQ	weighted fair queueing
WRED	weighted random early detection
WRR	weighted round-robin
XNS	Xerox Network System

APPENDIX C

Cisco IOS Release 12.2SRB Software Images

Table C-1 lists the software images that are provided with Cisco IOS Release 12.2SRB. These images are available only for the Cisco 7600 router. If you attempt to run any of these images on another platform (such as a Catalyst 6500 series switch), the system displays one of the error messages shown in the following section.

Note Special images are available to support both the Cisco 7600 router and the Catalyst 6500 switch (rsp72043-adventerprisek9-mz, s3223-adventerprisek9-mz, and s72033-adventerprisek9-mz). Contact your Cisco account representative to determine how to obtain one of these images or if an alternative image is available.

Table C-1 Cisco 7600 Release 12.2SRB Software Images

Image Filename	Description
c7600rsp72043-adventerprisek9-mz (RSP720) c7600s72033-adventerprisek9-mz (Sup720) c7600s3223-adventerprisek9-mz (Sup32)	Advanced enterprise services software images
c7600rsp72043-advipservicesk9-mz (RSP720) c7600s72033-advipservicesk9-mz (Sup720) c7600s3223-advipservicesk9-mz (Sup32)	Advanced IP services software images
c7600rsp72043-ipservices-mz (RSP720) c7600s72033-ipservices-mz (Sup720) c7600s3223-ipservices-mz (Sup32)	IP services software images
c7600rsp72043-ipservicesk9-mz (RSP720) c7600s72033-ipservicesk9-mz (Sup720) c7600s3223-ipservicesk9-mz (Sup32)	IP services SSH software images
c7600rsp720_rp-rm2.srec (RSP720)	ROMmon software image for route processor (available only for RSP720 in 12.2SRB)
c7600rsp720_sp-rm2.srec (RSP720)	ROMmon software image for switch processor (available only for RSP720 in 12.2SRB)

For information about the features in each image, see the Feature Navigator tool at:

http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp

Software Image Messages for Non-Compliant Platform

The Cisco IOS Release 12.2SRB images are available only for the Cisco 7600 router. Previously, Cisco 7600 router images would also run on the Catalyst 6500 series switch. Now, however, one of the following error messages is displayed if you attempt to run a Cisco 7600 image on another platform.

- If you attempt to run the 12.2SRB software image on a platform other than the Cisco 7600 router, the following message is displayed:

```
%C6K_PLATFORM-1- DETECTED_NON_C7600_CHASSIS: Software has detected an attempt to run a
c7600 image on an incompatible chassis ([char])
Explanation: You are loading a special 7600 12.2SR image on a Catalyst 6500 Series
Switch .
Recommended Action: You are loading a special Cisco 7600 Series Router image on a
Catalyst 6500 Series Switch. Thank you for registering your chassis for approval to
use this image.  If you have not registered this chassis, please contact your Cisco
account representative in order to gain approval for use of this image.
```

- If you attempt to run a Cisco 7600 image on another platform (such as the Catalyst 6500 switch), the following message is displayed:

```
%C6K_PLATFORM-1-DISABLE_NON_C7600_CHASSIS: Software has detected an attempt to run a
c7600 image on an incompatible chassis ([char]). This image may not boot.

Explanation: Software detected an attempt to load a Cisco 7600 series router image on
an incompatible chassis.
Recommended Action: Verify that you are using a Cisco 7600 series chassis; if you are
attempting to run this on a Catalyst 6500 series switch please contact your account
representative about selecting an appropriate image for your platform.
```

INDEX

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X