

You'll be entered into a quarterly drawing for **free** Cisco Press books by returning this survey! Cisco is dedicated to customer satisfaction and would like to hear your thoughts on these printed manuals. Please visit the Cisco Product Comments on-line survey at **www.cisco.com/go/crc** to submit your comments about accessing Cisco technical manuals. Thank you for your time

General Information

1	Years of networking experience:	_____	Years of experience with Cisco products:	_____
2	I have these network types:	____ LAN	____ Backbone	____ WAN
	Other:	_____		
3	I have these Cisco products:	____ Switches	____ Routers	
	Other (specify models):	_____		
4	I perform these types of tasks:	____ H/W installation and/or maintenance	____ S/W configuration	
	Network management	____ Other:	_____	
5	I use these types of documentation:	____ H/W installation	____ H/W configuration	____ S/W configuration
	Command reference	____ Quick reference	____ Release notes	____ Online help
	Other:	_____		
6	I access this information through:	____ % Cisco.com	____ % CD-ROM	____ % Printed manuals
	% Other:	_____		
7	I prefer this access method:	____ Cisco.com	____ CD-ROM	____ Printed manuals
	Other:	_____		
8	I use the following three product features the most:	_____		

Document Information

Document Title:	Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide		
Part Number:	OL-3999-07	S/W Release (if applicable):	12.2(18)SXF
On a scale of 1–5 (5 being the best), please let us know how we rate in the following areas:			
____ The document is complete.	____ The information is accurate.		
____ The information is well organized.	____ The information I wanted was easy to find.		
____ The document is written at my technical level of understanding.	____ The information I found was useful to my job.		
Please comment on our lowest scores: _____			

Mailing Information

Organization	Date	
Contact Name		
Mailing Address		
City	State/Province	Zip/Postal Code
Country	Phone ()	Extension
E-mail	Fax ()	
May we contact you further concerning our documentation? ____ Yes ____ No		

You can also send us your comments by e-mail to **bug-doc@cisco.com**, or by fax to **408-527-8089**.

When mailing this card from outside of the United States, please enclose in an envelope addressed to the location on the back of this card with the required postage or fax to 1-408-527-8089.

BUSINESS REPLY MAIL

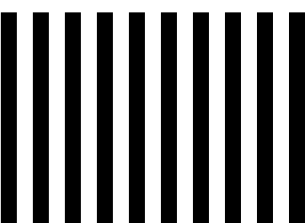
FIRST-CLASS MAIL PERMIT NO. 4631 SAN JOSE CA

POSTAGE WILL BE PAID BY ADDRESSEE

DOCUMENT RESOURCE CONNECTION
CISCO SYSTEMS INC
170 WEST TASMAN DR
SAN JOSE CA 95134-9916



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES





Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide

Release 12.2SX

Corporate 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: 408 526-4100

Text Part Number: OL-3999-07



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

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 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.

CCIP, CCSP, the Cisco Arrow logo, the Cisco *Powered* Network mark, Cisco Unity, Follow Me Browsing, FormShare, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, the Cisco IOS logo, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherSwitch, Fast Step, GigaStack, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, LightStream, MGX, MICA, the Networkers logo, Networking Academy, Network Registrar, *Packet*, PIX, Post-Routing, Pre-Routing, RateMUX, Registrar, ScriptShare, SlideCast, SMARTnet, StrataView Plus, Stratm, SwitchProbe, TeleRouter, The Fastest Way to Increase Your Internet Quotient, TransPath, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website 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. (0401R)

Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide, Release 12.2SX
© 2001–2005, Cisco Systems, Inc. All rights reserved.



Preface i

Audience	i
Organization	i
Related Documentation	iv
Conventions	v
Obtaining Documentation	vi
Cisco.com	vi
Product Documentation DVD	vi
Ordering Documentation	vi
Documentation Feedback	vii
Cisco Product Security Overview	vii
Reporting Security Problems in Cisco Products	vii
Obtaining Technical Assistance	viii
Cisco Technical Support & Documentation Website	viii
Submitting a Service Request	ix
Definitions of Service Request Severity	ix
Obtaining Additional Publications and Information	ix

Product Overview 1-1

Supported Hardware and Software	1-1
User Interfaces	1-1
Configuring Embedded CiscoView Support	1-2
Understanding Embedded CiscoView	1-2
Installing and Configuring Embedded CiscoView	1-2
Displaying Embedded CiscoView Information	1-3
Software Features Supported in Hardware by the PFC and DFC	1-3

Command-Line Interfaces 2-1

Accessing the CLI	2-1
Accessing the CLI through the EIA/TIA-232 Console Interface	2-2
Accessing the CLI through Telnet	2-2
Performing Command Line Processing	2-3

Performing History Substitution	2-3
Cisco IOS Command Modes	2-4
Displaying a List of Cisco IOS Commands and Syntax	2-5
ROM-Monitor Command-Line Interface	2-6
Configuring the Switch for the First Time	3-1
Default Configuration	3-1
Configuring the Switch	3-2
Using the Setup Facility or the setup Command	3-2
Using Configuration Mode	3-10
Checking the Running Configuration Before Saving	3-10
Saving the Running Configuration Settings	3-11
Reviewing the Configuration	3-11
Configuring a Default Gateway	3-11
Configuring a Static Route	3-12
Configuring a BOOTP Server	3-13
Protecting Access to Privileged EXEC Commands	3-15
Setting or Changing a Static Enable Password	3-15
Using the enable password and enable secret Commands	3-15
Setting or Changing a Line Password	3-16
Setting TACACS+ Password Protection for Privileged EXEC Mode	3-16
Encrypting Passwords	3-17
Configuring Multiple Privilege Levels	3-17
Recovering a Lost Enable Password	3-19
Modifying the Supervisor Engine Startup Configuration	3-20
Understanding the Supervisor Engine Boot Configuration	3-20
Configuring the Software Configuration Register	3-21
Specifying the Startup System Image	3-24
Understanding Flash Memory	3-24
CONFIG_FILE Environment Variable	3-25
Controlling Environment Variables	3-26
Configuring a Supervisor Engine 720	4-1
Using the Slots on a Supervisor Engine 720	4-1
Configuring Supervisor Engine 720 Ports	4-1
Configuring and Monitoring the Switch Fabric Functionality	4-2
Understanding How the Switch Fabric Functionality Works	4-2

Configuring the Switch Fabric Functionality	4-4
Monitoring the Switch Fabric Functionality	4-4
Configuring a Supervisor Engine 32	5-1
Flash Memory on a Supervisor Engine 32	5-1
Supervisor Engine 32 Ports	5-2
Configuring the Supervisor Engine 2 and the Switch Fabric Module	6-1
Using the Slots on a Supervisor Engine 2	6-1
Understanding How the Switch Fabric Module Works	6-1
Switch Fabric Module Overview	6-2
Switch Fabric Module Slots	6-2
Switch Fabric Redundancy	6-2
Forwarding Decisions for Layer 3-Switched Traffic	6-2
Switching Modes	6-3
Configuring the Switch Fabric Module	6-3
Configuring the Switching Mode	6-4
Configuring Fabric-Required Mode	6-4
Configuring an LCD Message	6-5
Monitoring the Switch Fabric Module	6-5
Displaying the Module Information	6-6
Displaying the Switch Fabric Module Redundancy Status	6-6
Displaying Fabric Channel Switching Modes	6-6
Displaying the Fabric Status	6-7
Displaying the Fabric Utilization	6-7
Displaying Fabric Errors	6-7
Configuring NSF with SSO Supervisor Engine Redundancy	7-1
Understanding NSF with SSO Supervisor Engine Redundancy	7-1
NSF with SSO Supervisor Engine Redundancy Overview	7-2
SSO Operation	7-2
NSF Operation	7-3
Cisco Express Forwarding	7-3
Multicast MLS NSF with SSO	7-4
Routing Protocols	7-4
NSF Benefits and Restrictions	7-8
Supervisor Engine Configuration Synchronization	7-9
Supervisor Engine Redundancy Guidelines and Restrictions	7-9

Redundancy Configuration Guidelines and Restrictions	7-10
Hardware Configuration Guidelines and Restrictions	7-10
Configuration Mode Restrictions	7-11
NSF Configuration Tasks	7-11
Configuring SSO	7-11
Configuring Multicast MLS NSF with SSO	7-12
Verifying Multicast NSF with SSO	7-13
Configuring CEF NSF	7-13
Verifying CEF NSF	7-13
Configuring BGP NSF	7-14
Verifying BGP NSF	7-14
Configuring OSPF NSF	7-15
Verifying OSPF NSF	7-15
Configuring IS-IS NSF	7-16
Verifying IS-IS NSF	7-17
Configuring EIGRP NSF	7-19
Verifying EIGRP NSF	7-19
Synchronizing the Supervisor Engine Configurations	7-20
Copying Files to the Redundant Supervisor Engine	7-20
Configuring SRM with SSO Supervisor Engine Redundancy	8-1
Understanding SRM with SSO	8-1
Supervisor Engine Redundancy Overview	8-2
SRM with SSO Operation	8-2
Supervisor Engine 720 Configuration Synchronization	8-4
Supervisor Engine 720 Redundancy Guidelines and Restrictions	8-4
Redundancy Guidelines and Restrictions	8-5
Hardware Configuration Guidelines and Restrictions	8-5
Configuration Mode Restrictions	8-6
Configuring Supervisor Engine 720 Redundancy	8-6
Configuring SRM with SSO Redundancy	8-6
Configuring the SRM with SSO Route Convergence Interval	8-7
Synchronizing the Supervisor Engine Configurations	8-8
Displaying the Redundancy States	8-8
Copying Files to the Redundant Supervisor Engine	8-8

Configuring RPR and RPR+ Supervisor Engine Redundancy 9-1

- Understanding RPR and RPR+ 9-1
 - Supervisor Engine Redundancy Overview 9-2
 - RPR Operation 9-2
 - RPR+ Operation 9-3
 - Supervisor Engine Configuration Synchronization 9-4
- Supervisor Engine Redundancy Guidelines and Restrictions 9-4
 - Redundancy Guidelines and Restrictions 9-5
 - RPR+ Guidelines and Restrictions 9-5
 - Hardware Configuration Guidelines and Restrictions 9-6
 - Configuration Mode Restrictions 9-6
- Configuring Supervisor Engine Redundancy 9-7
 - Configuring Redundancy 9-7
 - Synchronizing the Supervisor Engine Configurations 9-8
 - Displaying the Redundancy States 9-9
- Performing a Fast Software Upgrade 9-9
- Copying Files to the Redundant Supervisor Engine 9-11

Configuring Interfaces 10-1

- Understanding Interface Configuration 10-1
- Using the Interface Command 10-2
- Configuring a Range of Interfaces 10-4
- Defining and Using Interface-Range Macros 10-5
- Configuring Optional Interface Features 10-6
 - Configuring Ethernet Interface Speed and Duplex Mode 10-7
 - Configuring Jumbo Frame Support 10-10
 - Configuring IEEE 802.3Z Flow Control 10-13
 - Configuring the Port Debounce Timer 10-14
 - Adding a Description for an Interface 10-15
- Understanding Online Insertion and Removal 10-16
- Monitoring and Maintaining Interfaces 10-16
 - Monitoring Interface Status 10-17
 - Clearing Counters on an Interface 10-17
 - Resetting an Interface 10-18
 - Shutting Down and Restarting an Interface 10-18
- Checking the Cable Status Using the TDR 10-19

Configuring LAN Ports for Layer 2 Switching	11-1
Understanding How Layer 2 Switching Works	11-1
Understanding Layer 2 Ethernet Switching	11-1
Understanding VLAN Trunks	11-2
Layer 2 LAN Port Modes	11-4
Default Layer 2 LAN Interface Configuration	11-5
Layer 2 LAN Interface Configuration Guidelines and Restrictions	11-5
Configuring LAN Interfaces for Layer 2 Switching	11-6
Configuring a LAN Port for Layer 2 Switching	11-7
Configuring a Layer 2 Switching Port as a Trunk	11-7
Configuring a LAN Interface as a Layer 2 Access Port	11-14
Configuring a Custom IEEE 802.1Q EtherType Field Value	11-15
Configuring Flex Links	12-1
Understanding Flex Links	12-1
Configuring Flex Links	12-2
Flex Links Default Configuration	12-2
Flex Links Configuration Guidelines and Restrictions	12-2
Configuring Flex Links	12-3
Monitoring Flex Links	12-3
Configuring EtherChannels	13-1
Understanding How EtherChannels Work	13-1
EtherChannel Feature Overview	13-1
Understanding How EtherChannels Are Configured	13-2
Understanding Port Channel Interfaces	13-4
Understanding Load Balancing	13-5
EtherChannel Feature Configuration Guidelines and Restrictions	13-5
Configuring EtherChannels	13-6
Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels	13-7
Configuring Channel Groups	13-8
Configuring the LACP System Priority and System ID	13-10
Configuring EtherChannel Load Balancing	13-11
Configuring the EtherChannel Min-Links Feature	13-12
Configuring VTP	14-1
Understanding How VTP Works	14-1

Understanding the VTP Domain	14-2
Understanding VTP Modes	14-2
Understanding VTP Advertisements	14-3
Understanding VTP Version 2	14-3
Understanding VTP Pruning	14-3
VTP Default Configuration	14-5
VTP Configuration Guidelines and Restrictions	14-5
Configuring VTP	14-6
Configuring VTP Global Parameters	14-6
Configuring the VTP Mode	14-8
Displaying VTP Statistics	14-10
Configuring VLANs	15-1
Understanding How VLANs Work	15-1
VLAN Overview	15-1
VLAN Ranges	15-2
Configurable VLAN Parameters	15-3
Understanding Token Ring VLANs	15-3
VLAN Default Configuration	15-6
VLAN Configuration Guidelines and Restrictions	15-8
Configuring VLANs	15-9
VLAN Configuration Options	15-9
Creating or Modifying an Ethernet VLAN	15-10
Assigning a Layer 2 LAN Interface to a VLAN	15-12
Configuring the Internal VLAN Allocation Policy	15-12
Configuring VLAN Translation	15-13
Mapping 802.1Q VLANs to ISL VLANs	15-16
Configuring Private VLANs	16-1
Understanding How Private VLANs Work	16-1
Private VLAN Configuration Guidelines and Restrictions	16-2
Configuring Private VLANs	16-5
Configuring a VLAN as a Private VLAN	16-6
Associating Secondary VLANs with a Primary VLAN	16-7
Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN	16-8
Configuring a Layer 2 Interface as a Private VLAN Host Port	16-9
Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port	16-10

Configuring Cisco IP Phone Support	17-1
Understanding Cisco IP Phone Support	17-1
Cisco IP Phone Connections	17-1
Cisco IP Phone Voice Traffic	17-2
Cisco IP Phone Data Traffic	17-3
Cisco IP Phone Power Configurations	17-3
Default Cisco IP Phone Support Configuration	17-4
Cisco IP Phone Support Configuration Guidelines and Restrictions	17-4
Configuring Cisco IP Phone Support	17-5
Configuring Voice Traffic Support	17-5
Configuring Data Traffic Support	17-7
Configuring Inline Power Support	17-8
Configuring IEEE 802.1Q Tunneling	18-1
Understanding How 802.1Q Tunneling Works	18-1
802.1Q Tunneling Configuration Guidelines and Restrictions	18-3
Configuring 802.1Q Tunneling	18-6
Configuring 802.1Q Tunnel Ports	18-6
Configuring the Switch to Tag Native VLAN Traffic	18-6
Configuring Layer 2 Protocol Tunneling	19-1
Understanding How Layer 2 Protocol Tunneling Works	19-1
Configuring Support for Layer 2 Protocol Tunneling	19-2
Configuring Standard-Compliant IEEE MST	20-1
Understanding MST	20-1
MST Overview	20-2
MST Regions	20-2
IST, CIST, and CST	20-3
Hop Count	20-6
Boundary Ports	20-6
Standard-Compliant MST Implementation	20-7
Interoperability with IEEE 802.1D-1998 STP	20-9
Understanding RSTP	20-9
Port Roles and the Active Topology	20-10
Rapid Convergence	20-11

Synchronization of Port Roles	20-12
Bridge Protocol Data Unit Format and Processing	20-13
Topology Changes	20-15
Configuring MST	20-15
Default MST Configuration	20-16
MST Configuration Guidelines and Restrictions	20-16
Specifying the MST Region Configuration and Enabling MST	20-17
Configuring the Root Bridge	20-19
Configuring a Secondary Root Bridge	20-20
Configuring Port Priority	20-21
Configuring Path Cost	20-22
Configuring the Switch Priority	20-23
Configuring the Hello Time	20-24
Configuring the Forwarding-Delay Time	20-25
Configuring the Transmit Hold Count	20-25
Configuring the Maximum-Aging Time	20-26
Configuring the Maximum-Hop Count	20-26
Specifying the Link Type to Ensure Rapid Transitions	20-26
Designating the Neighbor Type	20-27
Restarting the Protocol Migration Process	20-28
Displaying the MST Configuration and Status	20-28
Configuring STP and Prestandard IEEE 802.1s MST	21-1
Understanding How STP Works	21-1
STP Overview	21-2
Understanding the Bridge ID	21-2
Understanding Bridge Protocol Data Units	21-4
Election of the Root Bridge	21-4
STP Protocol Timers	21-5
Creating the Spanning Tree Topology	21-5
STP Port States	21-6
STP and IEEE 802.1Q Trunks	21-12
Understanding How IEEE 802.1w RSTP Works	21-13
IEEE 802.1w RSTP Overview	21-13
RSTP Port Roles	21-13
RSTP Port States	21-14
Rapid-PVST	21-14
Understanding How Prestandard IEEE 802.1s MST Works	21-14

IEEE 802.1s MST Overview	21-15
MST-to-PVST Interoperability	21-16
Common Spanning Tree	21-17
MST Instances	21-18
MST Configuration Parameters	21-18
MST Regions	21-18
Message Age and Hop Count	21-20
Default STP Configuration	21-20
STP and MST Configuration Guidelines and Restrictions	21-21
Configuring STP	21-21
Enabling STP	21-22
Enabling the Extended System ID	21-23
Configuring the Root Bridge	21-24
Configuring a Secondary Root Bridge	21-25
Configuring STP Port Priority	21-26
Configuring STP Port Cost	21-28
Configuring the Bridge Priority of a VLAN	21-29
Configuring the Hello Time	21-31
Configuring the Forward-Delay Time for a VLAN	21-31
Configuring the Maximum Aging Time for a VLAN	21-32
Enabling Rapid-PVST	21-32
Configuring Prestandard IEEE 802.1s MST	21-33
Enabling MST	21-33
Displaying MST Configurations	21-35
Configuring MST Instance Parameters	21-38
Configuring MST Instance Port Parameters	21-39
Restarting Protocol Migration	21-39
Configuring Optional STP Features	22-1
Understanding How PortFast Works	22-2
Understanding How BPDU Guard Works	22-2
Understanding How PortFast BPDU Filtering Works	22-2
Understanding How UplinkFast Works	22-3
Understanding How BackboneFast Works	22-4
Understanding How EtherChannel Guard Works	22-6
Understanding How Root Guard Works	22-6
Understanding How Loop Guard Works	22-6
Enabling PortFast	22-8

Enabling PortFast BPDU Filtering	22-10
Enabling BPDU Guard	22-11
Enabling UplinkFast	22-12
Enabling BackboneFast	22-13
Enabling EtherChannel Guard	22-14
Enabling Root Guard	22-14
Enabling Loop Guard	22-15
Configuring Layer 3 Interfaces	23-1
Layer 3 Interface Configuration Guidelines and Restrictions	23-1
Configuring Subinterfaces on Layer 3 Interfaces	23-2
Configuring IPv4 Routing and Addresses	23-4
Configuring IPX Routing and Network Numbers	23-7
Configuring AppleTalk Routing, Cable Ranges, and Zones	23-8
Configuring Other Protocols on Layer 3 Interfaces	23-9
Configuring UDE and UDLR	24-1
Understanding UDE and UDLR	24-1
UDE and UDLR Overview	24-1
Supported Hardware	24-2
Understanding UDE	24-2
Understanding UDLR	24-3
Configuring UDE and UDLR	24-3
Configuring UDE	24-3
Configuring UDLR	24-6
Configuring PFC3BXL and PFC3B Mode Multiprotocol Label Switching	25-1
PFC3BXL and PFC3B Mode MPLS Label Switching	25-1
Understanding MPLS	25-2
Understanding PFC3BXL and PFC3B Mode MPLS Label Switching	25-2
Supported Hardware Features	25-4
Supported Cisco IOS Features	25-5
MPLS Guidelines and Restrictions	25-7
PFC3BXL and PFC3B Mode MPLS Supported Commands	25-7
Configuring MPLS	25-7
MPLS Per-Label Load Balancing	25-7

MPLS Configuration Examples	25-8
PFC3BXL or PFC3B Mode VPN Switching	25-9
PFC3BXL or PFC3B Mode VPN Switching Operation	25-10
MPLS VPN Guidelines and Restrictions	25-11
PFC3BXL or PFC3B Mode MPLS VPN Supported Commands	25-11
Configuring MPLS VPN	25-11
MPLS VPN Sample Configuration	25-12
Any Transport over MPLS	25-13
AToM Load Balancing	25-14
Understanding EoMPLS	25-14
EoMPLS Guidelines and Restrictions	25-14
Configuring EoMPLS	25-16
Configuring IPv4 Multicast VPN Support	26-1
Understanding How MVPN Works	26-1
MVPN Overview	26-1
Multicast Routing and Forwarding and Multicast Domains	26-2
Multicast Distribution Trees	26-2
Multicast Tunnel Interfaces	26-5
PE Router Routing Table Support for MVPN	26-6
Multicast Distributed Switching Support	26-6
Hardware-Assisted IPv4 Multicast	26-6
MVPN Configuration Guidelines and Restrictions	26-7
Configuring MVPN	26-8
Forcing Ingress Multicast Replication Mode (Optional)	26-8
Configuring a Multicast VPN Routing and Forwarding Instance	26-9
Configuring Multicast VRF Routing	26-15
Configuring Interfaces for Multicast Routing to Support MVPN	26-20
Sample Configurations for MVPN	26-22
MVPN Configuration with Default MDTs Only	26-22
MVPN Configuration with Default and Data MDTs	26-24
Configuring IP Unicast Layer 3 Switching	27-1
Understanding How Layer 3 Switching Works	27-1
Understanding Hardware Layer 3 Switching	27-2
Understanding Layer 3-Switched Packet Rewrite	27-2
Default Hardware Layer 3 Switching Configuration	27-4

Configuration Guidelines and Restrictions	27-4
Configuring Hardware Layer 3 Switching	27-4
Displaying Hardware Layer 3 Switching Statistics	27-5
Configuring IPv6 Multicast PFC3 and DFC3 Layer 3 Switching	28-1
Features that Support IPv6 Multicast	28-1
IPv6 Multicast Guidelines and Restrictions	28-2
New or Changed IPv6 Multicast Commands	28-3
Configuring IPv6 Multicast Layer 3 Switching	28-3
Using show Commands to Verify IPv6 Multicast Layer 3 Switching	28-3
Verifying MFIB Clients	28-4
Displaying the Switching Capability	28-4
Verifying the (S,G) Forwarding Capability	28-4
Verifying the (*,G) Forwarding Capability	28-5
Verifying the Subnet Entry Support Status	28-5
Verifying the Current Replication Mode	28-5
Displaying the Replication Mode Auto Detection Status	28-5
Displaying the Replication Mode Capabilities	28-5
Displaying Subnet Entries	28-6
Displaying the IPv6 Multicast Summary	28-6
Displaying the NetFlow Hardware Forwarding Count	28-6
Displaying the FIB Hardware Bridging and Drop Counts	28-7
Displaying the Shared and Well-Known Hardware Adjacency Counters	28-7
Configuring IPv4 Multicast Layer 3 Switching	29-1
Understanding How IPv4 Multicast Layer 3 Switching Works	29-1
IPv4 Multicast Layer 3 Switching Overview	29-2
Multicast Layer 3 Switching Cache	29-2
Layer 3-Switched Multicast Packet Rewrite	29-3
Partially and Completely Switched Flows	29-3
Non-RPF Traffic Processing	29-5
Understanding How IPv4 Bidirectional PIM Works	29-6
Default IPv4 Multicast Layer 3 Switching Configuration	29-7
IPv4 Multicast Layer 3 Switching Configuration Guidelines and Restrictions	29-7
Restrictions	29-7
Unsupported Features	29-8

Configuring IPv4 Multicast Layer 3 Switching	29-8
Source-Specific Multicast with IGMPv3, IGMP v3lite, and URD	29-9
Enabling IPv4 Multicast Routing Globally	29-9
Enabling IPv4 PIM on Layer 3 Interfaces	29-10
Enabling IP Multicast Layer 3 Switching Globally	29-10
Enabling IP Multicast Layer 3 Switching on Layer 3 Interfaces	29-10
Configuring the Replication Mode	29-11
Enabling Local Egress Replication	29-12
Specifying the Maximum Number of Multicast Routes	29-14
Configuring the Layer 3 Switching Global Threshold	29-14
Enabling Installation of Directly Connected Subnets	29-15
Specifying the Flow Statistics Message Interval	29-15
Enabling Shortcut-Consistency Checking	29-15
Configuring ACL-Based Filtering of RPF Failures	29-16
Displaying RPF Failure Rate-Limiting Information	29-16
Displaying IPv4 Multicast Layer 3 Hardware Switching Summary	29-17
Displaying the IPv4 Multicast Routing Table	29-20
Displaying IPv4 Multicast Layer 3 Switching Statistics	29-21
Configuring IPv4 Bidirectional PIM	29-22
Enabling IPv4 Bidirectional PIM Globally	29-22
Configuring the Rendezvous Point for IPv4 Bidirectional PIM Groups	29-23
Setting the IPv4 Bidirectional PIM Scan Interval	29-24
Displaying IPv4 Bidirectional PIM Information	29-24
Using IPv4 Debug Commands	29-26
Clearing IPv4 Multicast Layer 3 Switching Statistics	29-27
Configuring MLDv2 Snooping for IPv6 Multicast Traffic	30-1
Understanding How MLDv2 Snooping Works	30-1
MLDv2 Snooping Overview	30-2
MLDv2 Messages	30-2
Source-Based Filtering	30-3
Explicit Host Tracking	30-3
MLDv2 Snooping Proxy Reporting	30-3
Joining an IPv6 Multicast Group	30-4
Leaving a Multicast Group	30-6
Understanding the MLDv2 Snooping Querier	30-7
Default MLDv2 Snooping Configuration	30-7
MLDv2 Snooping Configuration Guidelines and Restrictions	30-7

MLDv2 Snooping Querier Configuration Guidelines and Restrictions	30-8
Enabling the MLDv2 Snooping Querier	30-8
Configuring MLDv2 Snooping	30-9
Enabling MLDv2 Snooping	30-9
Configuring a Static Connection to a Multicast Receiver	30-10
Configuring a Multicast Router Port Statically	30-11
Configuring the MLD Snooping Query Interval	30-11
Enabling Fast-Leave Processing	30-12
Enabling SSM Safe Reporting	30-12
Configuring Explicit Host Tracking	30-13
Configuring Report Suppression	30-13
Displaying MLDv6 Snooping Information	30-14
Configuring IGMP Snooping for IPv4 Multicast Traffic	31-1
Understanding How IGMP Snooping Works	31-1
IGMP Snooping Overview	31-2
Joining a Multicast Group	31-2
Leaving a Multicast Group	31-4
Understanding the IGMP Snooping Querier	31-5
Understanding IGMP Version 3 Support	31-5
Default IGMP Snooping Configuration	31-7
IGMP Snooping Configuration Guidelines and Restrictions	31-8
IGMP Snooping Querier Configuration Guidelines and Restrictions	31-8
Enabling the IGMP Snooping Querier	31-9
Configuring IGMP Snooping	31-9
Enabling IGMP Snooping	31-10
Configuring a Static Connection to a Multicast Receiver	31-11
Configuring a Multicast Router Port Statically	31-11
Configuring the IGMP Snooping Query Interval	31-11
Enabling IGMP Fast-Leave Processing	31-12
Configuring Source Specific Multicast (SSM) Mapping	31-12
Enabling SSM Safe Reporting	31-13
Configuring IGMPv3 Explicit Host Tracking	31-13
Displaying IGMP Snooping Information	31-14

Configuring PIM Snooping	32-1
Understanding How PIM Snooping Works	32-1
Default PIM Snooping Configuration	32-4
PIM Snooping Configuration Guidelines and Restrictions	32-4
Configuring PIM Snooping	32-4
Enabling PIM Snooping Globally	32-5
Enabling PIM Snooping in a VLAN	32-5
Disabling PIM Snooping Designated-Router Flooding	32-6
Configuring RGMP	33-1
Understanding How RGMP Works	33-1
Default RGMP Configuration	33-2
RGMP Configuration Guidelines and Restrictions	33-2
Enabling RGMP on Layer 3 Interfaces	33-3
Configuring Network Security	34-1
Configuring MAC Address-Based Traffic Blocking	34-1
Configuring TCP Intercept	34-2
Configuring Unicast Reverse Path Forwarding Check	34-2
Understanding PFC3 Unicast RPF Check Support	34-2
Understanding PFC2 Unicast RPF Check Support	34-3
Unicast RPF Check Guidelines and Restrictions	34-3
Configuring Unicast RPF Check	34-3
Understanding Cisco IOS ACL Support	35-1
Cisco IOS ACL Configuration Guidelines and Restrictions	35-1
Hardware and Software ACL Support	35-2
Optimized ACL Logging on Supervisor Engine 720	35-3
Understanding OAL	35-3
OAL Guidelines and Restrictions	35-3
Configuring OAL	35-4
Guidelines and Restrictions for Using Layer 4 Operators in ACLs	35-5
Determining Layer 4 Operation Usage	35-6
Determining Logical Operation Unit Usage	35-6
Configuring VLAN ACLs	36-1
Understanding VACLs	36-1
VACL Overview	36-2

Bridged Packets	36-2
Routed Packets	36-3
Multicast Packets	36-4
Configuring VACLs	36-4
VACL Configuration Overview	36-5
Defining a VLAN Access Map	36-5
Configuring a Match Clause in a VLAN Access Map Sequence	36-6
Configuring an Action Clause in a VLAN Access Map Sequence	36-7
Applying a VLAN Access Map	36-8
Verifying VLAN Access Map Configuration	36-8
VLAN Access Map Configuration and Verification Examples	36-9
Configuring a Capture Port	36-9
Configuring VACL Logging	36-11
Configuring Denial of Service Protection	37-1
Understanding How DoS Protection Works	37-2
DoS Protection for the Supervisor Engine 2	37-2
DoS Protection on the Supervisor Engine 720	37-10
DoS Protection Default Configuration	37-22
DoS Protection Configuration Guidelines and Restrictions	37-22
Supervisor Engine 2	37-23
Supervisor Engine 720	37-23
Monitoring Packet Drop Statistics	37-24
Displaying Rate-Limiter Information	37-27
Understanding How Control Plane Policing Works	37-29
CoPP Default Configuration	37-29
CoPP Configuration Guidelines and Restrictions	37-30
Configuring CoPP	37-31
Monitoring CoPP	37-32
Defining Traffic Classification	37-33
Traffic Classification Overview	37-33
Traffic Classification Guidelines	37-34
Sample Basic ACLs for CoPP Traffic Classification	37-34
Configuring Sticky ARP	37-36
Configuring DHCP Snooping	38-1
Overview of DHCP Snooping	38-1

DHCP Snooping Option-82 Data Insertion	38-2
Overview of the DHCP Snooping Database Agent	38-4
Default Configuration for DHCP Snooping	38-5
DHCP Snooping Configuration Guidelines and Restrictions	38-5
Configuring DHCP Snooping	38-6
Enabling DHCP Snooping Globally	38-7
Enabling DHCP Option-82 Data Insertion	38-7
Enabling DHCP Snooping MAC Address Verification	38-8
Enabling DHCP Snooping on VLANs	38-8
Configuring the DHCP Trust State on Layer 2 LAN Interfaces	38-9
Configuring DHCP Snooping Rate Limiting on Layer 2 LAN Interfaces	38-10
Configuring the DHCP Snooping Database Agent	38-11
Configuration Examples for the Database Agent	38-11
Displaying a Binding Table	38-14
Configuring Dynamic ARP Inspection	39-1
Understanding DAI	39-1
Understanding ARP	39-1
Understanding ARP Spoofing Attacks	39-2
Understanding DAI and ARP Spoofing Attacks	39-2
Interface Trust States and Network Security	39-3
Rate Limiting of ARP Packets	39-4
Relative Priority of ARP ACLs and DHCP Snooping Entries	39-4
Logging of Dropped Packets	39-4
Default DAI Configuration	39-5
DAI Configuration Guidelines and Restrictions	39-5
Configuring DAI	39-6
Enabling DAI on VLANs	39-7
Configuring the DAI Interface Trust State	39-7
Applying ARP ACLs for DAI Filtering	39-8
Configuring ARP Packet Rate Limiting	39-9
Enabling DAI Error-Disabled Recovery	39-10
Enabling Additional Validation	39-11
Configuring DAI Logging	39-12
Displaying DAI Information	39-15
DAI Configuration Samples	39-16

Sample One: Two Switches Support DAI 39-16

Sample Two: One Switch Supports DAI 39-20

Configuring Traffic Storm Control 40-1

Understanding Traffic Storm Control 40-1

Default Traffic Storm Control Configuration 40-3

Enabling Traffic Storm Control 40-3

Displaying Traffic Storm Control Settings 40-5

Unknown Unicast Flood Blocking 41-1

Understanding UUFB 41-1

Configuring UUFB 41-1

Configuring PFC QoS 42-1

Understanding How PFC QoS Works 42-2

Hardware Supported by PFC QoS 42-2

QoS Terminology 42-3

PFC QoS Feature Flowcharts 42-5

PFC QoS Feature Summary 42-11

Ingress LAN Port Features 42-12

PFC Marking and Policing 42-15

Egress LAN Port Features 42-22

PFC QoS Default Configuration 42-24

PFC QoS Global Settings 42-24

Default Values with PFC QoS Enabled 42-25

Default Values With PFC QoS Disabled 42-36

PFC QoS Configuration Guidelines and Restrictions 42-37

Configuring PFC QoS 42-41

Enabling PFC QoS Globally 42-41

Preserving the Received ToS Byte 42-42

Enabling Queueing-Only Mode 42-43

Enabling Microflow Policing of Bridged Traffic 42-43

Enabling VLAN-Based PFC QoS on Layer 2 LAN Ports 42-44

Enabling Egress ACL Support for Remarked DSCP 42-45

Creating Named Aggregate Policers 42-46

Configuring a PFC QoS Policy 42-48

Configuring Egress DSCP Mutation on a PFC3 42-65

Configuring Ingress CoS Mutation on IEEE 802.1Q Tunnel Ports 42-67

Configuring DSCP Value Maps	42-70
Configuring the Trust State of Ethernet LAN and OSM Ingress Ports	42-74
Configuring the Ingress LAN Port CoS Value	42-75
Configuring Standard-Queue Drop Threshold Percentages	42-76
Mapping CoS Values to Drop Thresholds	42-82
Allocating Bandwidth Between LAN-Port Standard Transmit Queues	42-86
Setting the Receive-Queue Size Ratio on 1p1q0t and 1p1q8t Ingress LAN Ports	42-87
Setting the LAN-Port Transmit-Queue Size Ratio	42-88
Configuring PFC3BXL or PFC3B Mode MPLS QoS	43-1
Terminology	43-2
PFC3BXL or PFC3B Mode MPLS QoS Features	43-3
MPLS Experimental Field	43-3
Trust	43-3
Classification	43-3
Policing and Marking	43-4
Preserving IP ToS	43-4
EXP Mutation	43-4
MPLS DiffServ Tunneling Modes	43-4
PFC3BXL or PFC3B Mode MPLS QoS Overview	43-4
Specifying the QoS in the IP Precedence Field	43-5
PFC3BXL or PFC3B Mode MPLS QoS	43-5
LERs at the Input Edge of an MPLS Network	43-6
LSRs in the Core of an MPLS Network	43-6
LERs at the Output Edge of an MPLS Network	43-7
Understanding PFC3BXL or PFC3B Mode MPLS QoS	43-7
LERs at the EoMPLS Edge	43-8
LERs at the IP Edge (MPLS, MPLS VPN)	43-9
LSRs at the MPLS Core	43-12
PFC3BXL or PFC3B MPLS QoS Default Configuration	43-15
MPLS QoS Commands	43-16
PFC3BXL or PFC3B Mode MPLS QoS Restrictions and Guidelines	43-17
Configuring PFC3BXL or PFC3B Mode MPLS QoS	43-17
Enabling QoS Globally	43-18
Enabling Queueing-Only Mode	43-19
Configuring a Class Map to Classify MPLS Packets	43-20
Configuring a Policy Map	43-22

Displaying a Policy Map	43-27
Configuring PFC3BXL or PFC3B mode MPLS QoS Egress EXP Mutation	43-28
Configuring EXP Value Maps	43-29
MPLS DiffServ Tunneling Modes	43-30
Short Pipe Mode	43-31
Uniform Mode	43-32
MPLS DiffServ Tunneling Restrictions and Usage Guidelines	43-33
Configuring Short Pipe Mode	43-34
Ingress PE Router—Customer Facing Interface	43-34
Configuring Ingress PE Router—P Facing Interface	43-35
Configuring the P Router—Output Interface	43-36
Configuring the Egress PE Router—Customer Facing Interface	43-37
Configuring Uniform Mode	43-38
Configuring the Ingress PE Router—Customer Facing Interface	43-38
Configuring the Ingress PE Router—P Facing Interface	43-40
Configuring the Egress PE Router—Customer Facing Interface	43-41
Configuring PFC QoS Statistics Data Export	44-1
Understanding PFC QoS Statistics Data Export	44-1
PFC QoS Statistics Data Export Default Configuration	44-2
Configuring PFC QoS Statistics Data Export	44-2
Configuring the Cisco IOS Firewall Feature Set	45-1
Cisco IOS Firewall Feature Set Support Overview	45-1
Cisco IOS Firewall Guidelines and Restrictions	45-2
Additional CBAC Configuration	45-3
Configuring IEEE 802.1X Port-Based Authentication	46-1
Understanding 802.1X Port-Based Authentication	46-1
Device Roles	46-2
Authentication Initiation and Message Exchange	46-3
Ports in Authorized and Unauthorized States	46-4
Supported Topologies	46-4
Default 802.1X Port-Based Authentication Configuration	46-5
802.1X Port-Based Authentication Guidelines and Restrictions	46-6
Configuring 802.1X Port-Based Authentication	46-7

Enabling 802.1X Port-Based Authentication	46-7
Configuring Switch-to-RADIUS-Server Communication	46-8
Enabling Periodic Reauthentication	46-10
Manually Reauthenticating the Client Connected to a Port	46-11
Initializing Authentication for the Client Connected to a Port	46-11
Changing the Quiet Period	46-11
Changing the Switch-to-Client Retransmission Time	46-12
Setting the Switch-to-Client Retransmission Time for EAP-Request Frames	46-13
Setting the Switch-to-Authentication-Server Retransmission Time for Layer 4 Packets	46-13
Setting the Switch-to-Client Frame Retransmission Number	46-14
Enabling Multiple Hosts	46-14
Resetting the 802.1X Configuration to the Default Values	46-15
Displaying 802.1X Status	46-15
Configuring Port Security	47-1
Understanding Port Security	47-1
Port Security with Dynamically Learned and Static MAC Addresses	47-1
Port Security with Sticky MAC Addresses	47-2
Default Port Security Configuration	47-2
Port Security Guidelines and Restrictions	47-3
Configuring Port Security	47-3
Enabling Port Security	47-4
Configuring the Port Security Violation Mode on a Port	47-6
Configuring the Maximum Number of Secure MAC Addresses on a Port	47-7
Enabling Port Security with Sticky MAC Addresses on a Port	47-8
Configuring a Static Secure MAC Address on a Port	47-9
Configuring Secure MAC Address Aging on a Port	47-10
Displaying Port Security Settings	47-11
Configuring CDP	48-1
Understanding How CDP Works	48-1
Configuring CDP	48-1
Enabling CDP Globally	48-2
Displaying the CDP Global Configuration	48-2
Enabling CDP on a Port	48-2
Displaying the CDP Interface Configuration	48-3

Monitoring and Maintaining CDP	48-3
Configuring UDLD	49-1
Understanding How UDLD Works	49-1
UDLD Overview	49-1
UDLD Aggressive Mode	49-2
Default UDLD Configuration	49-3
Configuring UDLD	49-3
Enabling UDLD Globally	49-3
Enabling UDLD on Individual LAN Interfaces	49-4
Disabling UDLD on Fiber-Optic LAN Interfaces	49-4
Configuring the UDLD Probe Message Interval	49-5
Resetting Disabled LAN Interfaces	49-5
Configuring NetFlow and NDE	50-1
Understanding How NetFlow and NDE Work	50-2
NetFlow and NDE Overview	50-2
NetFlow and NDE on the MSFC	50-2
NetFlow and NDE on the PFC	50-2
Default NetFlow and NDE Configuration	50-8
NetFlow and NDE Configuration Guidelines and Restrictions	50-9
Configuring NetFlow and NDE	50-9
Configuring NetFlow and NDE on the PFC	50-10
Configuring NetFlow and NDE on the MSFC	50-15
Enabling NetFlow and NDE for Ingress Bridged IP Traffic	50-17
Displaying the NDE Address and Port Configuration	50-18
Configuring NDE Flow Filters	50-19
Displaying the NDE Configuration	50-21
Configuring Local SPAN, RSPAN, and ERSPAN	51-1
Understanding How Local SPAN, RSPAN, and ERSPAN Work	51-1
Local SPAN, RSPAN, and ERSPAN Overview	51-2
Local SPAN, RSPAN, and ERSPAN Sources	51-5
Local SPAN, RSPAN, and ERSPAN Destination Ports	51-6
Local SPAN, RSPAN, and ERSPAN Configuration Guidelines and Restrictions	51-6
Feature Incompatibilities	51-6
Local SPAN, RSPAN, and ERSPAN Session Limits	51-7
Local SPAN, RSPAN, and ERSPAN Guidelines and Restrictions	51-8

VSPAN Guidelines and Restrictions	51-10
RSPAN Guidelines and Restrictions	51-10
ERSPAN Guidelines and Restrictions	51-11
Configuring Local SPAN, RSPAN, and ERSPAN	51-12
Configuring Destination Port Permit Lists (Optional)	51-13
Configuring Local SPAN	51-13
Configuring RSPAN	51-15
Configuring ERSPAN	51-17
Configuring Source VLAN Filtering for Local SPAN and RSPAN	51-22
Configuring a Destination Port as an Unconditional Trunk	51-22
Configuring Destination Trunk Port VLAN Filtering	51-23
Verifying the Configuration	51-24
Configuration Examples	51-25
Configuring SNMP IfIndex Persistence	52-1
Understanding SNMP IfIndex Persistence	52-1
Configuring SNMP IfIndex Persistence	52-2
Enabling SNMP IfIndex Persistence Globally	52-2
Disabling SNMP IfIndex Persistence Globally	52-2
Enabling and Disabling SNMP IfIndex Persistence on Specific Interfaces	52-2
Clearing SNMP IfIndex Persistence Configuration from a Specific Interface	52-3
Power Management and Environmental Monitoring	53-1
Understanding How Power Management Works	53-1
Enabling or Disabling Power Redundancy	53-2
Powering Modules Off and On	53-3
Viewing System Power Status	53-4
Power Cycling Modules	53-5
Power Cycling Power Supplies	53-5
Determining System Power Requirements	53-5
Determining System Hardware Capacity	53-5
Determining Sensor Temperature Threshold	53-9
Understanding How Environmental Monitoring Works	53-10
Monitoring System Environmental Status	53-10
Understanding LED Environmental Indications	53-12

Configuring Online Diagnostics	54-1
Understanding How Online Diagnostics Work	54-1
Configuring Online Diagnostics	54-2
Setting Bootup Online Diagnostics Level	54-2
Configuring On-Demand Online Diagnostics	54-3
Scheduling Online Diagnostics	54-4
Configuring Health-Monitoring Diagnostics	54-5
Running Online Diagnostic Tests	54-6
Starting and Stopping Online Diagnostic Tests	54-6
Displaying Online Diagnostic Tests and Test Results	54-6
Schedule Switchover	54-10
Performing Memory Tests	54-10
Diagnostic Sanity Check	54-11
Configuring Web Cache Services Using WCCP	55-1
Understanding WCCP	55-2
WCCP Overview	55-2
Hardware Acceleration	55-3
Understanding WCCPv1 Configuration	55-4
Understanding WCCPv2 Configuration	55-5
WCCPv2 Features	55-6
Restrictions for WCCPv2	55-7
Configuring WCCP	55-8
Specifying a Version of WCCP	55-8
Configuring a Service Group Using WCCPv2	55-8
Excluding Traffic on a Specific Interface from Redirection	55-10
Registering a Router to a Multicast Address	55-10
Using Access Lists for a WCCP Service Group	55-11
Setting a Password for a Router and Cache Engines	55-11
Verifying and Monitoring WCCP Configuration Settings	55-12
WCCP Configuration Examples	55-12
Changing the Version of WCCP on a Router Example	55-13
Performing a General WCCPv2 Configuration Example	55-13
Running a Web Cache Service Example	55-13
Running a Reverse Proxy Service Example	55-14
Registering a Router to a Multicast Address Example	55-14

Using Access Lists Example	55-14
Setting a Password for a Router and Cache Engines Example	55-15
Verifying WCCP Settings Example	55-15

Using the Top N Utility 56-1

Understanding the Top N Utility	56-1
Top N Utility Overview	56-1
Understanding Top N Utility Operation	56-2
Using the Top N Utility	56-2
Enabling Top N Utility Report Creation	56-3
Displaying the Top N Utility Reports	56-3
Clearing Top N Utility Reports	56-4

Using the Layer 2 Traceroute Utility 57-1

Understanding the Layer 2 Traceroute Utility	57-1
Usage Guidelines	57-1
Using the Layer 2 Traceroute Utility	57-2

APPENDIX A

Acronyms A-1

INDEX



Preface

This preface describes who should read the *Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide*, Release 12.2SX, how it is organized, and its document conventions.

Audience

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

Organization

This guide is organized as follows:

Chapter	Title	Description
Chapter 1	Product Overview	Presents an overview of the Catalyst 6500 series switches.
Chapter 2	Command-Line Interfaces	Describes how to use the command-line interface (CLI).
Chapter 3	Configuring the Switch for the First Time	Describes how to perform a baseline configuration.
Chapter 4	Configuring a Supervisor Engine 720	Describes how to configure a Supervisor Engine 720.
Chapter 5	Configuring a Supervisor Engine 32	Describes how to configure a Supervisor Engine 32.
Chapter 6	Configuring the Supervisor Engine 2 and the Switch Fabric Module	Describes how to configure a Supervisor Engine 2 and the Switch Fabric Module.
Chapter 7	Configuring NSF with SSO Supervisor Engine Redundancy	Describes how to configure NSF with SSO supervisor engine redundancy.
Chapter 8	Configuring SRM with SSO Supervisor Engine Redundancy	Describes how to configure SRM with SSO supervisor engine redundancy.

Chapter	Title	Description
Chapter 9	Configuring RPR and RPR+ Supervisor Engine Redundancy	Describes how to configure RPR and RPR+ supervisor engine redundancy.
Chapter 10	Configuring Interfaces	Describes how to configure non-layer-specific features on LAN interfaces.
Chapter 11	Configuring LAN Ports for Layer 2 Switching	Describes how to configure LAN interfaces to support Layer 2 features, including VLAN trunks.
Chapter 12	Configuring Flex Links	Describes how to configure Flex Links.
Chapter 13	Configuring EtherChannels	Describes how to configure Layer 2 and Layer 3 EtherChannel port bundles.
Chapter 14	Configuring VTP	Describes how to configure the VLAN Trunking Protocol (VTP).
Chapter 15	Configuring VLANs	Describes how to configure VLANs.
Chapter 16	Configuring Private VLANs	Describes how to configure private VLANs.
Chapter 17	Configuring Cisco IP Phone Support	Describes how to configure Cisco IP Phone support.
Chapter 18	Configuring IEEE 802.1Q Tunneling	Describes how to configure IEEE 802.1Q tunneling.
Chapter 19	Configuring Layer 2 Protocol Tunneling	Describes how to configure Layer 2 protocol tunneling.
Chapter 20	Configuring Standard-Compliant IEEE MST	Describes how to configure standard-compliant IEEE MST.
Chapter 21	Configuring STP and Prestandard IEEE 802.1s MST	Describes how to configure the Spanning Tree Protocol (STP) and Prestandard IEEE 802.1s Multiple Spanning Tree (MST).
Chapter 22	Configuring Optional STP Features	Describes how to configure the STP PortFast, UplinkFast, and BackboneFast features.
Chapter 23	Configuring Layer 3 Interfaces	Describes how to configure LAN interfaces to support Layer 3 features.
Chapter 24	Configuring UDE and UDLR	Describes how to configure unidirectional Ethernet (UDE) and unidirectional link routing (UDLR)
Chapter 25	Configuring PFC3BXL and PFC3B Mode Multiprotocol Label Switching	Describes how to configure PFC3BXL or PFC3B Multiprotocol Label Switching (MPLS).
Chapter 26	Configuring IPv4 Multicast VPN Support	Describes how to configure IPv4 Multicast Virtual Private Network (MVPN).
Chapter 27	Configuring IP Unicast Layer 3 Switching	Describes how to configure IP unicast Layer 3 switching.
Chapter 28	Configuring IPv6 Multicast PFC3 and DFC3 Layer 3 Switching	Describes how to configure IPv6 Multicast Multilayer Switching (MMLS).
Chapter 29	Configuring IPv4 Multicast Layer 3 Switching	Describes how to configure IPv4 Multicast Multilayer Switching (MMLS).
Chapter 30	Configuring MLDv2 Snooping for IPv6 Multicast Traffic	Describes how to configure Multicast Listener Discovery version 2 (MLDv2) snooping.

Chapter	Title	Description
Chapter 31	Configuring IGMP Snooping for IPv4 Multicast Traffic	Describes how to configure Internet Group Management Protocol (IGMP) snooping.
Chapter 32	Configuring PIM Snooping	Describes how to configure protocol independent multicast (PIM) snooping.
Chapter 33	Configuring RGMP	Describes how to configure Router-Port Group Management Protocol (RGMP).
Chapter 34	Configuring Network Security	Describes how to configure network security features that are unique to the Catalyst 6500 series switches.
Chapter 35	Understanding Cisco IOS ACL Support	Describes how Catalyst 6500 series switches support Cisco IOS ACLs.
Chapter 36	Configuring VLAN ACLs	Describes how configure VLAN ACLs.
Chapter 37	Configuring Denial of Service Protection	Describes how configure denial of service protection.
Chapter 38	Configuring DHCP Snooping	Describes how configure DHCP snooping.
Chapter 39	Configuring Dynamic ARP Inspection	Describes how configure dynamic ARP inspection.
Chapter 40	Configuring Traffic Storm Control	Describes how to configure traffic storm control.
Chapter 41	Unknown Unicast Flood Blocking	Describes how to configure unknown unicast flood blocking.
Chapter 42	Configuring PFC QoS	Describes how to configure quality of service (QoS).
Chapter 43	Configuring PFC3BXL or PFC3B Mode MPLS QoS	Describes how to configure MPLS QoS.
Chapter 44	Configuring PFC QoS Statistics Data Export	Describes how to configure PFC QoS statistics data export.
Chapter 45	Configuring the Cisco IOS Firewall Feature Set	Describes how to configure the Cisco IOS Firewall feature set.
Chapter 46	Configuring IEEE 802.1X Port-Based Authentication	Describes how to configure IEEE 802.1X port-based authentication.
Chapter 47	Configuring Port Security	Describes how to configure port security.
Chapter 48	Configuring CDP	Describes how to configure Cisco Discovery Protocol (CDP).
Chapter 49	Configuring UDLD	Describes how to configure the UniDirectional Link Detection (UDLD) protocol.
Chapter 50	Configuring NetFlow and NDE	Describes how to configure Neflow Data Export (NDE).
Chapter 51	Configuring Local SPAN, RSPAN, and ERSPAN	Describes how to configure the Switch Port Analyzer (SPAN).
Chapter 52	Configuring SNMP IfIndex Persistence	Describes how to configure SNMP ifIndex persistence.
Chapter 53	Power Management and Environmental Monitoring	Describes how to configure power management and environmental monitoring features.

Chapter	Title	Description
Chapter 54	Configuring Online Diagnostics	Describes how to configure online diagnostics and run diagnostic tests.
Chapter 55	Configuring Web Cache Services Using WCCP	Describes how to configure the Web Cache Communication Protocol (WCCP) .
Chapter 56	Using the Top N Utility	Describes how to use the Top N utility.
Chapter 57	Using the Layer 2 Traceroute Utility	Describes how to use the Layer 2 traceroute utility.

Related Documentation

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

- *Catalyst 6500 Series Switch Installation Guide*
- *Catalyst 6500 Series Switch Module Installation Guide*
- *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX
- *Catalyst 6500 Series Switch Cisco IOS System Message Guide*, Release 12.2SX
- *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 Catalyst 6500 series switch publications:
 - *Configuration Fundamentals Configuration Guide*
 - *Configuration Fundamentals Command Reference*
 - *Bridging and IBM Networking Configuration Guide*
 - *Bridging and IBM Networking Command Reference*
 - *Interface Configuration Guide*
 - *Interface Command Reference*
 - *Network Protocols Configuration Guide*, Part 1, 2, and 3
 - *Network Protocols Command Reference*, Part 1, 2, and 3
 - *Security Configuration Guide*
 - *Security Command Reference*
 - *Switching Services Configuration Guide*
 - *Switching Services Command Reference*
 - *Voice, Video, and Home Applications Configuration Guide*
 - *Voice, Video, and Home Applications Command Reference*
 - *Software Command Summary*
 - *Software System Error Messages*
 - *Debug Command Reference*
 - *Internetwork Design Guide*
 - *Internetwork Troubleshooting Guide*

– *Configuration Builder Getting Started Guide*

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

<http://www.cisco.com/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 .
<i>italic font</i>	Arguments for which you supply values are in <i>italics</i> .
[]	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 <i>screen font</i> .
boldface screen font	Information you must enter is in boldface screen font .
<i>italic screen font</i>	Arguments for which you supply values are in <i>italic screen font</i> .
→	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

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

<http://www.cisco.com/techsupport>

You can access the Cisco website at this URL:

<http://www.cisco.com>

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Product Documentation DVD

Cisco documentation and additional literature are available in the Product Documentation DVD package, which may have shipped with your product. The Product Documentation DVD is updated regularly and may be more current than printed documentation.

The Product Documentation DVD is a comprehensive library of technical product documentation on portable media. The DVD enables you to access multiple versions of hardware and software installation, configuration, and command guides for Cisco products and to view technical documentation in HTML. With the DVD, you have access to the same documentation that is found on the Cisco website without being connected to the Internet. Certain products also have .pdf versions of the documentation available.

The Product Documentation DVD is available as a single unit or as a subscription. Registered Cisco.com users (Cisco direct customers) can order a Product Documentation DVD (product number DOC-DOCDVD=) from Cisco Marketplace at this URL:

<http://www.cisco.com/go/marketplace/>

Ordering Documentation

Beginning June 30, 2005, registered Cisco.com users may order Cisco documentation at the Product Documentation Store in the Cisco Marketplace at this URL:

<http://www.cisco.com/go/marketplace/>

Nonregistered Cisco.com users can order technical documentation from 8:00 a.m. to 5:00 p.m. (0800 to 1700) PDT by calling 1 866 463-3487 in the United States and Canada, or elsewhere by calling 011 408 519-5055. You can also order documentation by e-mail at tech-doc-store-mkpl@external.cisco.com or by fax at 1 408 519-5001 in the United States and Canada, or elsewhere at 011 408 519-5001.

Documentation Feedback

You can rate and provide feedback about Cisco technical documents by completing the online feedback form that appears with the technical documents on Cisco.com.

You can send comments about Cisco documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.
- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

<http://www.cisco.com/go/psirt>

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies—security-alert@cisco.com

An emergency is either a condition in which a system is under active attack or a condition for which a severe and urgent security vulnerability should be reported. All other conditions are considered nonemergencies.

- Nonemergencies—psirt@cisco.com

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

**Tip**

We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.x through 8.x.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one linked in the Contact Summary section of the Security Vulnerability Policy page at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

The link on this page has the current PGP key ID in use.

Obtaining Technical Assistance

Cisco Technical Support provides 24-hour-a-day award-winning technical assistance. The Cisco Technical Support & Documentation website on Cisco.com features extensive online support resources. In addition, if you have a valid Cisco service contract, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not have a valid Cisco service contract, contact your reseller.

Cisco Technical Support & Documentation Website

The Cisco Technical Support & Documentation website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, at this URL:

<http://www.cisco.com/techsupport>

Access to all tools on the Cisco Technical Support & Documentation website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>

**Note**

Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support & Documentation website by clicking the **Tools & Resources** link under Documentation & Tools. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco engineer. The TAC Service Request Tool is located at this URL:

<http://www.cisco.com/techsupport/servicerequest>

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)

EMEA: +32 2 704 55 55

USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

<http://www.cisco.com/techsupport/contacts>

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, documentation, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

<http://www.cisco.com/go/marketplace/>

- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

<http://www.ciscopress.com>

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

<http://www.cisco.com/packet>

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

<http://www.cisco.com/go/iqmagazine>

or view the digital edition at this URL:

<http://ciscoiq.texterity.com/ciscoiq/sample/>

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

<http://www.cisco.com/ipj>

- Networking products offered by Cisco Systems, as well as customer support services, can be obtained at this URL:

<http://www.cisco.com/en/US/products/index.html>

- Networking Professionals Connection is an interactive website for networking professionals to share questions, suggestions, and information about networking products and technologies with Cisco experts and other networking professionals. Join a discussion at this URL:

<http://www.cisco.com/discuss/networking>

- World-class networking training is available from Cisco. You can view current offerings at this URL:

<http://www.cisco.com/en/US/learning/index.html>



Product Overview

This chapter consists of these sections:

- [Supported Hardware and Software, page 1-1](#)
- [User Interfaces, page 1-1](#)
- [Configuring Embedded CiscoView Support, page 1-2](#)
- [Software Features Supported in Hardware by the PFC and DFC, page 1-3](#)

Supported Hardware and Software

For complete information about the chassis, modules, and software features supported by the Catalyst 6500 series switches, refer to the *Release Notes for Cisco IOS Release 12.2SX on the Supervisor Engine 720, Supervisor Engine 32, and Supervisor Engine 2*:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/ol_4164.htm

User Interfaces

Release 12.2SX supports configuration using the following interfaces:

- CLI—See [Chapter 2, “Command-Line Interfaces.”](#)
- SNMP—Refer to the Release 12.2 IOS *Configuration Fundamentals Configuration Guide and Command Reference* at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/index.htm
- Cisco IOS web browser interface—Refer to “Using the Cisco Web Browser” in the IOS *Configuration Fundamentals Configuration Guide* at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/ffcprt1/fcf005.htm
- 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, page 1-2](#)
- [Installing and Configuring Embedded CiscoView, page 1-2](#)
- [Displaying Embedded CiscoView Information, page 1-3](#)

Understanding Embedded CiscoView

The Embedded CiscoView network management system is a web-based interface that uses HTTP and SNMP to provide a graphical representation of the switch and to provide a GUI-based management and configuration interface. You can download the Java Archive (JAR) files for Embedded CiscoView at this URL:

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

Installing and Configuring Embedded CiscoView

To install and configure Embedded CiscoView, perform this task:

	Command	Purpose
Step 1	Router# dir <i>device_name</i>	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 <i>device_name:cv/*</i>	Removes existing files from the CiscoView directory.
Step 3	Router# squeeze <i>device_name:</i>	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 <i>device_name:</i>	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 <i>string</i> ro	Configures the SNMP password for read-only operation.
Step 9	Router(config)# snmp-server community <i>string</i> rw	Configures the SNMP password for read/write operation.



Note

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

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

http://www.cisco.com/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), Policy Feature Card 2 (PFC2), Distributed Forwarding Card 3 (DFC3) and Distributed Forwarding Card (DFC):

- [Software Features Supported in Hardware by the PFC3, PFC2, 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, PFC2, DFC3, and DFC

The PFC3, PFC2, 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 35, “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/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/trafwl/index.htm

- VLAN ACLs (VACLs)—To configure VACLs, see [Chapter 36, “Configuring VLAN ACLs.”](#)

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

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

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcpbr1/qcftpbr.htm



Note

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

- Except on MPLS interfaces, TCP intercept—To configure TCP intercept, see the “[Configuring TCP Intercept](#)” section on page 34-2.
- Firewall feature set images provide these features:
 - Context-Based Access Control (CBAC) —The PFC installs entries in the NetFlow table to direct flows that require CBAC to the MSFC where the CBAC is applied in software on the MSFC.
 - Authentication Proxy—After authentication on the MSFC, the PFC provides TCAM support for the authentication policy.
 - Port-to-Application Mapping (PAM)—PAM is done in software on the MSFC.

To configure firewall features, see [Chapter 45, “Configuring the Cisco IOS Firewall Feature Set.”](#)

- Hardware-assisted NetFlow Aggregation—Refer to this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/nde.htm#1081085>

Software Features Supported in Hardware by the PFC3 and DFC3

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

- With Release 12.2(18)SX and later releases, 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—Refer to the publication at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/mcastmls.htm#1068744>
- Multiple-path Unicast Reverse Path Forwarding (RPF) Check—To configure Unicast RPF Check, see the “[Configuring Unicast Reverse Path Forwarding Check](#)” section on page 34-2.
- 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 supported only in PFC3BXL or PFC3B 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 to be processed in software. (CSCdz51590)

To configure NAT, refer to the *Cisco IOS IP Configuration Guide*, Release 12.2, “IP Addressing and Services,” “Configuring IP Addressing,” “Configuring Network Address Translation,” at this URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcprt1/1cfipadr.htm#1042290

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/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/m1.htm#56404>
(CSCea23296)

- With Release 12.2(18)SXE and later, the PFC3 and DFC3s support IPv4 multicast over point-to-point GRE tunnels in hardware. Releases earlier than Release 12.2(18)SXE support IPv4 multicast over point-to-point GRE tunnels in software on the MSFC.
- GRE Tunneling and IP in IP Tunneling—The PFC3 and DFC3s support the following **tunnel** commands:
 - **tunnel destination**
 - **tunnel mode gre**
 - **tunnel mode ipip**
 - **tunnel source**
 - **tunnel ttl**
 - **tunnel tos**

The MSFC3 supports tunneling configured with any other tunnel commands.

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

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

To configure GRE Tunneling and IP in IP Tunneling, refer to these publications:

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

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

http://www.cisco.com/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. (CSCdy72539)
- Each tunnel interface uses one internal VLAN.
- Each tunnel interface uses one additional router MAC address entry per router MAC address.
- The PFC3A does not support any PFC QoS features on tunnel interfaces.
- The PFC3B and PFC3BXL support PFC QoS features on tunnel interfaces.

- The MSFC3 supports tunnels configured with egress features on the tunnel interface. Examples of egress features are output Cisco IOS ACLs, NAT (for inside to outside translation), TCP intercept, CBAC, and encryption.



Command-Line Interfaces

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



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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:

- [Accessing the CLI, page 2-1](#)
- [Performing Command Line Processing, page 2-3](#)
- [Performing History Substitution, page 2-3](#)
- [Cisco IOS Command Modes, page 2-4](#)
- [Displaying a List of Cisco IOS Commands and Syntax, page 2-5](#)
- [ROM-Monitor Command-Line Interface, page 2-6](#)

Accessing the CLI

These sections describe accessing the CLI:

- [Accessing the CLI through the EIA/TIA-232 Console Interface, page 2-2](#)
- [Accessing the CLI through Telnet, page 2-2](#)

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



Note

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

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

To make a console connection, perform this task:

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

After making a console connection, you see this display:

Press Return for Console prompt

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

Accessing the CLI through Telnet



Note

Before you can make a Telnet connection to the switch, you must configure an IP address (see the [“Configuring IPv4 Routing and Addresses” section on page 23-4](#)).

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

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

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

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

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

User Access Verification

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

Performing Command Line Processing

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

Table 2-1 Keyboard Shortcuts

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

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

Performing History Substitution

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

Table 2-2 History Substitution Commands

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

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

Cisco IOS Command Modes



Note

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

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

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

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

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



Note

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

ROM-monitor mode is a separate mode used when the switch cannot boot properly. For example, the switch might enter ROM-monitor mode if it does not find a valid system image when it is booting, or if its configuration file is corrupted at startup. See the “[ROM-Monitor Command-Line Interface](#)” section on page 2-6.

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

Table 2-3 Frequently Used Cisco IOS Command Modes

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

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

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

Displaying a List of Cisco IOS Commands and Syntax

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

```
Router> ?
```

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

```
Router# co?
configure
```

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

For example:

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

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



Tip

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

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

ROM-Monitor Command-Line Interface

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

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



Note

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

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

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

For more information about the ROM-monitor commands, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX.



Configuring the Switch for the First Time

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

- *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/index.htm
- *Cisco IOS Configuration Fundamentals Configuration Command Reference*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_r/index.htm



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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:

- [Default Configuration, page 3-1](#)
- [Configuring the Switch, page 3-2](#)
- [Protecting Access to Privileged EXEC Commands, page 3-15](#)
- [Recovering a Lost Enable Password, page 3-19](#)
- [Modifying the Supervisor Engine Startup Configuration, page 3-20](#)

Default Configuration

Table 3-1 shows the default configuration.

Table 3-1 Default Configuration

Feature	Default Value
Administrative connection	Normal mode
Global information	No value for the following: <ul style="list-style-type: none"> • System name • System contact • Location
System clock	No value for system clock time
Passwords	No passwords configured for normal mode or enable mode (press the Return key)
Prompt	Router>

Configuring the Switch

These sections describe how to configure the switch:

- [Using the Setup Facility or the setup Command, page 3-2](#)
- [Using Configuration Mode, page 3-10](#)
- [Checking the Running Configuration Before Saving, page 3-10](#)
- [Saving the Running Configuration Settings, page 3-11](#)
- [Reviewing the Configuration, page 3-11](#)
- [Configuring a Default Gateway, page 3-11](#)
- [Configuring a Static Route, page 3-12](#)
- [Configuring a BOOTP Server, page 3-13](#)

Using the Setup Facility or the setup Command

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

- [Setup Overview, page 3-2](#)
- [Configuring the Global Parameters, page 3-3](#)
- [Configuring Interfaces, page 3-8](#)

Setup Overview

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

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

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

```
Configuring interface FastEthernet3/1:
  Is this interface in use?: yes
  Configure IP on this interface?: yes
```

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

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

Configuring the Global Parameters

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

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

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

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

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

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

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

Restricted Rights Legend

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

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

Cisco Internetwork Operating System Software
IOS (tm) c6sup2_sp Software (c6sup2_sp-SPV-M), Version 12.1(5c)EX, EARLY DEPLOYM
ENT RELEASE SOFTWARE (fc1)
Synced to mainline version: 12.1(5c)
TAC:Home:Software:Ios General:CiscoIOSRoadmap:12.1
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Wed 28-Mar-01 18:36 by hqluong
Image text-base: 0x30020980, data-base: 0x306B8000
```

Start as Primary processor

00:00:05: %SYS-3-LOGGER_FLUSHING: System pausing to ensure console debugging output.

00:00:03: Currently running ROMMON from S (Gold) region

00:00:05: %OIR-6-CONSOLE: Changing console ownership to route processor

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

rommon 1 > boot

Self decompressing the image : #####

[OK]

Restricted Rights Legend

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

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

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

cisco Cat6k-MSFC2 (R7000) processor with 114688K/16384K bytes of memory.
Processor board ID SAD04430J9K
R7000 CPU at 300Mhz, Implementation 39, Rev 2.1, 256KB L2, 1024KB L3 Cache
Last reset from power-on
X.25 software, Version 3.0.0.
509K bytes of non-volatile configuration memory.

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

Press RETURN to get started!



Note

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

--- System Configuration Dialog ---

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

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

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



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

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

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

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

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

Current interface summary

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	TFTP	administratively down	down
GigabitEthernet1/1	unassigned	YES	TFTP	administratively down	down
GigabitEthernet1/2	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/1	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/2	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/3	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/4	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/5	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/6	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/7	unassigned	YES	TFTP	administratively down	down
GigabitEthernet3/8	unassigned	YES	TFTP	administratively down	down

(Additional displayed text omitted from this example.)

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

Current interface summary

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	TFTP	administratively down	down
GigabitEthernet1/1	172.20.52.34	YES	NVRAM	up	up
GigabitEthernet1/2	unassigned	YES	TFTP	administratively down	down

```
GigabitEthernet3/1      unassigned      YES TFTP      administratively down down
GigabitEthernet3/2      unassigned      YES TFTP      administratively down down
GigabitEthernet3/3      unassigned      YES TFTP      administratively down down
GigabitEthernet3/4      unassigned      YES TFTP      administratively down down
GigabitEthernet3/5      unassigned      YES TFTP      administratively down down
GigabitEthernet3/6      unassigned      YES TFTP      administratively down down
GigabitEthernet3/7      unassigned      YES TFTP      administratively down down
GigabitEthernet3/8      unassigned      YES TFTP      administratively down down

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

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

A typical minimal configuration using IP follows and continues through [Step 8](#):

Configuring global parameters:

```
Enter host name [Router]: Router
```

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

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

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

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

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

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

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



Note

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

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

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

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

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

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

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

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

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

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

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/fcftp3/fc014.htm

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

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

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

The following configuration command script was created:

```
hostname router
enable secret 5 $1$S3Lx$SuiTYg2UrFK1U0dgWdjvxw.
enable password lab
line vty 0 4
password lab
no snmp-server
!
ip routing eigrp 301

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

```

. !
end

```

```

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

```

```

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

```

This completes the procedure on how to configure global parameters. The setup facility continues with the process to configure interfaces in the next section “[Configuring Interfaces](#).”

Configuring Interfaces

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



Note

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

To configure interfaces, follow these steps:

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

```

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

```

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

```

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

```

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

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

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

```

Use this configuration? [yes/no]:

```

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


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

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

```
Router>
```

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

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

```
Router# show version
Cisco Internetwork Operating System Software
IOS (tm) C6sup2_rp Software (C6sup2_rp-JSV-M), Version 12.1(5c)EX, EARLY DEPLOY)
Synced to mainline version: 12.1(5c)
TAC:Home:Software:Ios General:CiscoIOSRoadmap:12.1
Copyright (c) 1986-2001 by Cisco Systems, Inc.
Compiled Wed 28-Mar-01 17:52 by hqluong
Image text-base: 0x30008980, data-base: 0x315D0000

ROM: System Bootstrap, Version 12.1(3r)E2, RELEASE SOFTWARE (fc1)
BOOTFLASH: C6sup2_rp Software (C6sup2_rp-JSV-M), Version 12.1(5c)EX, EARLY DEPL)

Router uptime is 2 hours, 33 minutes
System returned to ROM by power-on (SP by power-on)
Running default software

cisco Catalyst 6000 (R7000) processor with 114688K/16384K bytes of memory.
Processor board ID SAD04430J9K
R7000 CPU at 300Mhz, Implementation 39, Rev 2.1, 256KB L2, 1024KB L3 Cache
Last reset from power-on
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
1 Virtual Ethernet/IEEE 802.3 interface(s)
48 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)
381K bytes of non-volatile configuration memory.

16384K bytes of Flash internal SIMM (Sector size 512K).
Configuration register is 0x2
Router#
```

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

http://www.cisco.com/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 switch from configuration mode as follows:

-
- Step 1** Connect a console terminal to the console interface of your supervisor engine.
- Step 2** When you are asked if you want to enter the initial dialog, answer **no** to enter the normal operating mode as follows:

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

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

```
Router> enable
```



Note Configuration changes can only be made in enable mode.

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

```
Router#
```

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

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

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

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

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

- Step 5** Save your settings. (See the [“Saving the Running Configuration Settings”](#) section on page 3-11.)
-

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

Checking the Running Configuration Before Saving

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

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

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

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

Saving the Running Configuration Settings

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

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

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

Reviewing the Configuration

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

Configuring a Default Gateway



Note

The switch uses the default gateway only when it is not configured to route.

To send data to another subnet when the switch is not configured with a routing protocol, configure a default gateway. The default gateway must be the IP address of an interface on a router in the same subnet.

To configure a default gateway, perform this task:

	Command	Purpose
Step 1	Router(config)# ip default-gateway A.B.C.D	Configures a default gateway.
Step 2	Router# show ip route	Verifies that the default gateway appears correctly in the IP routing table.

This example shows how to configure a default gateway and how to verify the configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip default-gateway 172.20.52.35
Router(config)# end
3d17h: %SYS-5-CONFIG_I: Configured from console by console
Router# show ip route
Default gateway is 172.20.52.35

Host                Gateway                Last Use    Total Uses  Interface
ICMP redirect cache is empty
Router#
```

Configuring a Static Route

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

To configure a static route, perform this task:

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

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

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

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

```
Router# show running-config
Building configuration...
.
<...output truncated...>
```

```
.
ip 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 switch with subnet mask and connected over VLAN 1:

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

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

```
Router# show running-config
Building configuration...
.
<...output truncated...>
.
ip 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 switch boots, it automatically retrieves the IP address from the BOOTP server.

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

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

-
- Step 1** Install the BOOTP server code on the workstation, if it is not already installed.
 - Step 2** Determine the MAC address from the label on the chassis.
 - Step 3** Add an entry in the BOOTP configuration file (usually /usr/etc/bootptab) for each switch. Press **Return** after each entry to create a blank line between each entry. See the example BOOTP configuration file that follows in Step 4.
 - Step 4** Enter the **reload** command to reboot and automatically request the IP address from the BOOTP server.

This example BOOTP configuration file shows the added entry:

```
# /etc/bootptab: database for bootp server (/etc/bootpd)
#
# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#
#      first field -- hostname
#                      (may be full domain name and probably should be)
#
#      hd -- home directory
#      bf -- bootfile
#      cs -- cookie servers
#      ds -- domain name servers
#      gw -- gateways
#      ha -- hardware address
#      ht -- hardware type
#      im -- impress servers
#      ip -- host IP address
#      lg -- log servers
#      lp -- LPR servers
#      ns -- IEN-116 name servers
#      rl -- resource location protocol servers
#      sm -- subnet mask
#      tc -- template host (points to similar host entry)
#      to -- time offset (seconds)
#      ts -- time servers
#
<information deleted>
#
#####
# Start of individual host entries
#####
Router:      tc=netcisco0:   ha=0000.0ca7.ce00:   ip=172.31.7.97:
dross:      tc=netcisco0:   ha=00000c000139:   ip=172.31.7.26:
<information deleted>
```

Protecting Access to Privileged EXEC Commands

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

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

Setting or Changing a Static Enable Password

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

Command	Purpose
Router(config)# enable password <i>password</i>	Sets a new password or changes an existing password for the privileged EXEC mode.

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

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

To display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration” section on page 3-19](#).

Using the enable password and enable secret Commands

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

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

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

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

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

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

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

**Note**

You cannot recover a lost encrypted password. You must clear NVRAM and set a new password. See the “[Recovering a Lost Enable Password](#)” section on page 3-19 if you lose or forget your password.

To display the password or access level configuration, see the “[Displaying the Password, Access Level, and Privilege Level Configuration](#)” section on page 3-19.

Setting or Changing a Line Password

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

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

To display the password or access level configuration, see the “[Displaying the Password, Access Level, and Privilege Level Configuration](#)” section on page 3-19.

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/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fsaaa/index.htm
- *Cisco IOS Security Command Reference*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_r/index.htm

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

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

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

**Caution**

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

**Note**

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

Encrypting Passwords

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

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

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

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

**Caution**

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

Although you cannot recover a lost encrypted password (that is, you cannot get the original password back), you can regain control of the switch after you lose or forget the encrypted password. See the [“Recovering a Lost Enable Password” section on page 3-19](#) if you lose or forget your password.

To display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration” section on page 3-19](#).

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 3-18](#)
- [Changing the Default Privilege Level for Lines, page 3-18](#)
- [Logging In to a Privilege Level, page 3-18](#)
- [Exiting a Privilege Level, page 3-19](#)
- [Displaying the Password, Access Level, and Privilege Level Configuration, page 3-19](#)

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 <i>command</i>	Sets the privilege level for a command.
Step 2	Router(config)# enable password level level <i>[encryption-type] password</i>	Specifies the enable password for a privilege level.

To display the password or access level configuration, see the “[Displaying the Password, Access Level, and Privilege Level Configuration](#)” section on page 3-19.

Changing the Default Privilege Level for Lines

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

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

To display the password or access level configuration, see the “[Displaying the Password, Access Level, and Privilege Level Configuration](#)” section on page 3-19.

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 <i>level</i>	Exits to a specified privilege level.

Displaying the Password, Access Level, and Privilege Level Configuration

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

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

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

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

This example shows how to display the privilege level configuration:

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

Recovering a Lost Enable Password

To recover a lost enable password, follow these steps:

- Step 1 Connect to the console interface.
- Step 2 Configure the switch to boot up without reading the configuration memory (NVRAM).
- Step 3 Reboot the system.
- Step 4 Access enable mode (which can be done without a password when one is not configured).
- Step 5 View or change the password, or erase the configuration.
- Step 6 Reconfigure the switch to boot up and read the NVRAM as it normally does.
- Step 7 Reboot the system.



Note

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

Modifying the Supervisor Engine Startup Configuration

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

- [Understanding the Supervisor Engine Boot Configuration, page 3-20](#)
- [Configuring the Software Configuration Register, page 3-21](#)
- [Specifying the Startup System Image, page 3-24](#)
- [Understanding Flash Memory, page 3-24](#)
- [CONFIG_FILE Environment Variable, page 3-25](#)
- [Controlling Environment Variables, page 3-26](#)

Understanding the Supervisor Engine Boot Configuration

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

Understanding the Supervisor Engine Boot Process

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

Two user-configurable parameters determine how the switch boots: the configuration register and the BOOT environment variable. The configuration register is described in the [“Modifying the Boot Field and Using the boot Command” section on page 3-22](#). The BOOT environment variable is described in the [“Specifying the Startup System Image” section on page 3-24](#).

Understanding the ROM Monitor

The ROM monitor executes upon power-up, reset, or when a fatal exception occurs. The switch enters ROM-monitor mode if the switch does not find a valid software image, if the NVRAM configuration is corrupted, or if the configuration register is set to enter ROM-monitor mode. From ROM-monitor mode, you can manually load a software image from bootflash or a Flash PC card.



Note

For complete syntax and usage information for the ROM monitor commands, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX publication.

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



Note

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

The ROM monitor has these features:

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

Configuring the Software Configuration Register

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

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

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

[Table 3-2](#) lists the meaning of each of the software configuration memory bits, and [Table 3-3](#) defines the boot field.



Caution

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

Table 3-2 Software Configuration Register Bit Meaning

Bit Number ¹	Hexadecimal	Meaning
00 to 03	0x0000 to 0x000F	Boot field (see Table 3-3)
06	0x0040	Causes system software to ignore NVRAM contents
07	0x0080	OEM ² bit enabled
08	0x0100	Break disabled
09	0x0200	Use secondary bootstrap
10	0x0400	Internet Protocol (IP) broadcast with all zeros

Table 3-2 Software Configuration Register Bit Meaning (continued)

Bit Number ¹	Hexadecimal	Meaning
11 to 12	0x0800 to 0x1000	Console line speed (default is 9600 baud)
13	0x2000	Boot default flash software if network boot fails
14	0x4000	IP broadcasts do not have network numbers
15	0x8000	Enable diagnostic messages and ignore NVRAM contents

1. The factory default value for the configuration register is 0x2102.

2. OEM = original equipment manufacturer.

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

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

Modifying the Boot Field and Using the boot Command

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

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



Note

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

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

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

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

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

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

Modifying the Boot Field

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

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

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

-
- Step 1** Enter the **enable** command and your password to enter privileged level as follows:
- ```
Router> enable
Password:
Router#
```
- Step 2** Enter the **configure terminal** command at the EXEC mode prompt (#) as follows:
- ```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```
- Step 3** Configure the configuration register to 0x2102 as follows:
- ```
Router(config)# config-register 0x2102
```
- Set the contents of the configuration register by entering the **config-register value** configuration command, where *value* is a hexadecimal number preceded by 0x (see [Table 3-2 on page 3-21](#)).
- Step 4** Enter the **end** command to exit configuration mode. The new value settings are saved to memory; however, the new settings do not take effect until the system software is reloaded by rebooting the system.
- Step 5** Enter the **show version EXEC** command to display the configuration register value currently in effect and that will be used at the next reload. The value is displayed on the last line of the screen display, as in this example:
- ```
Configuration register is 0x141 (will be 0x2102 at next reload)
```
- Step 6** Save your settings.
- See the [“Saving the Running Configuration Settings” section on page 3-11](#). However, note that configuration register changes take effect only after the system reloads, such as when you enter a **reload** command from the console.
- Step 7** Reboot the system.
- The new configuration register value takes effect with the next system boot.
-

Verifying the Configuration Register Setting

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

To verify the configuration register setting, perform this task:

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

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

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

Specifying the Startup System Image

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



Note

- Store the system software image in the **sup-bootflash:**, **disk0:**, or **disk1:** device (only Supervisor Engine 720 has **disk1:**).
- A non-ATA Flash PC card in a Supervisor Engine 2 is **slot0:**. Non-ATA Flash PC cards are too small for Release 12.2SX images.
- 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 “Specify the Startup System Image in the Configuration File” section in the “Loading and Maintaining System Images and Microcode” chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Understanding Flash Memory

The following sections describe flash memory:

- [Flash Memory Features, page 3-25](#)
- [Security Features, page 3-25](#)
- [Flash Memory Configuration Process, page 3-25](#)



Note

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

Flash Memory Features

The flash memory components allow you to do the following:

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

Security Features

The flash memory components support the following security features:

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

Flash Memory Configuration Process

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

-
- | | |
|---------------|---|
| Step 1 | Copy a system image to flash memory using TFTP or rcp (refer to the <i>Cisco IOS Configuration Fundamentals Configuration Guide</i> , Release 12.2, “Cisco IOS File Management,” “Loading and Maintaining System Images,” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/ffcprt2/fcf008.htm |
| Step 2 | Configure the system to boot automatically from the file in flash memory. You might need to change the configuration register value. See the “ Modifying the Boot Field and Using the boot Command ” section on page 3-22, 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 **nvr**am:, **disk**0:, 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 switch checks the variable upon startup to determine the location and filename of the configuration file to use for initialization.

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

Controlling Environment Variables

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

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



Note

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

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

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

```
Router# show bootvar
BOOT variable = disk0:c6sup22-jsv-mz.121-5c.EX.bin,1;
CONFIG_FILE variable does not exist
BOOTLDR variable = bootflash:c6msfc2-boot-mz.121-3a.E4
Configuration register is 0x2
Router#
```

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



Configuring a Supervisor Engine 720

This chapter describes how to configure a Supervisor Engine 720 in a Catalyst 6500 series switch. This chapter contains these sections:

- [Using the Slots on a Supervisor Engine 720, page 4-1](#)
- [Configuring Supervisor Engine 720 Ports, page 4-1](#)
- [Configuring and Monitoring the Switch Fabric Functionality, page 4-2](#)



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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 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, including Microdrive cards. 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, page 4-2](#)
- [Configuring the Switch Fabric Functionality, page 4-4](#)
- [Monitoring the Switch Fabric Functionality, page 4-4](#)

Understanding How the Switch Fabric Functionality Works

These sections describe how the switch fabric functionality works:

- [Switch Fabric Functionality Overview, page 4-2](#)
- [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 functionality, fabric-enabled modules also have a direct connection to the 32-Gbps forwarding bus., page 4-2](#)
- [Forwarding Decisions for Layer 3-Switched Traffic, page 4-3](#)
- [Switching Modes, page 4-3](#)

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 functionality, fabric-enabled modules also have a direct connection to the 32-Gbps forwarding bus.

Forwarding Decisions for Layer 3-Switched Traffic

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

- A PFC3 makes all forwarding decisions for each packet that enters the switch through a module without a DFC3.
- A DFC3 makes all forwarding decisions for each packet that enters the switch on a DFC3-enabled module in these situations:
 - If the egress port is on the same module as the ingress port, the DFC3 forwards the packet locally (the packet never leaves the module).
 - If the egress port is on a different fabric-enabled module, the DFC3 sends the packet to the egress module, which sends it out the egress port.
 - If the egress port is on a different nonfabric-enabled module, the DFC3 sends the packet to the Supervisor Engine 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 switch uses this mode for all traffic when only fabric-enabled modules are installed. In this mode, a compact version of the DBus header is forwarded over the switch fabric channel, which provides the best possible performance.
- Truncated mode—The switch uses this mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.
- Bus mode—The switch uses this mode for traffic between nonfabric-enabled modules and for traffic between a nonfabric-enabled module and a fabric-enabled module. In this mode, all traffic passes between the local bus and the supervisor engine bus.

Table 4-1 shows the switching modes used with fabric-enabled and nonfabric-enabled modules installed.

Table 4-1 Switch Fabric Functionality Switching Modes

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

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

2. Displayed as fabric mode in **show** commands.

Configuring the Switch Fabric Functionality

To configure the switching mode, perform this task:

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

When configuring the switching mode, note the following information:

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



Caution

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

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

Monitoring the Switch Fabric Functionality

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

These sections describe how to monitor the switch fabric functionality:

- [Displaying the Switch Fabric Redundancy Status, page 4-5](#)
- [Displaying Fabric Channel Switching Modes, page 4-5](#)
- [Displaying the Fabric Status, page 4-5](#)
- [Displaying the Fabric Utilization, page 4-6](#)
- [Displaying Fabric Errors, page 4-6](#)

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
          channel      speed      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 [<i>slot_number</i> 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 [<i>slot_number</i> 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#
```




Configuring a Supervisor Engine 32

This chapter describes how to configure a Supervisor Engine 32 in a Catalyst 6500 series switch. This chapter contains these sections:

- [Flash Memory on a Supervisor Engine 32, page 5-1](#)
- [Supervisor Engine 32 Ports, page 5-2](#)



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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
- Supervisor Engine 32 has a PFC3B and operates in PFC3B mode.
- With a 3-slot or 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.
- 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/univercd/cc/td/doc/product/lan/cat6000/122sx/ol_4164.htm

Flash Memory on a Supervisor Engine 32

The Supervisor Engine 32 has one external CompactFlash Type II slot and 256 MB internal CompactFlash Flash memory. The CompactFlash Type II slot supports CompactFlash Type II Flash PC cards.

The keyword for the external compact flash memory is **disk0:**. The keyword for the internal compact flash memory is **bootdisk:**.

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.



Configuring the Supervisor Engine 2 and the Switch Fabric Module

This chapter describes how to configure the Supervisor Engine 2 and the Switch Fabric Module (SFM) for the Catalyst 6500 series switches.



Note

- Release 12.2(18)SXE and rebuilds of Release 12.2(18)SXE do not support Supervisor Engine 2.
- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Using the Slots on a Supervisor Engine 2, page 6-1](#)
- [Understanding How the Switch Fabric Module Works, page 6-1](#)
- [Configuring the Switch Fabric Module, page 6-3](#)
- [Monitoring the Switch Fabric Module, page 6-5](#)

Using the Slots on a Supervisor Engine 2

The Supervisor Engine 2 has one Flash PC card (PCMCIA) slot.

With PCMCIA Advanced Technology Attachment (ATA) FlashDisk devices, the keyword for the slot on the active Supervisor Engine 2 is **disk0:** and the keyword for the slot on a redundant Supervisor Engine 2 is **slavedisk0:**.

With non-ATA Flash PC cards, the keyword for the slot on the active Supervisor Engine 2 is **slot0:** and the keyword for the slot on a redundant Supervisor Engine 2 is **slaveslot0:**.

Understanding How the Switch Fabric Module Works

These sections describe how the Switch Fabric Module works:

- [Switch Fabric Module Overview, page 6-2](#)
- [Switch Fabric Module Slots, page 6-2](#)

- [Switch Fabric Redundancy, page 6-2](#)
- [Forwarding Decisions for Layer 3-Switched Traffic, page 6-2](#)
- [Switching Modes, page 6-3](#)

Switch Fabric Module Overview

The Switch Fabric Module 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 Module, fabric-enabled modules also have a direct connection to the 32-Gbps forwarding bus.

The Switch Fabric Module does not have a console. A two-line LCD display on the front panel shows fabric utilization, software revision, and basic system information.

Switch Fabric Module Slots

With a 13-slot chassis, install the Switch Fabric Modules in either slot 7 or 8.



Note

In a 13-slot chassis, only slots 9 through 13 support dual switch fabric interface switching modules (for example, WS-X6816-GBIC).

With all other chassis, install the Switch Fabric Modules in either slot 5 or 6.

Switch Fabric Redundancy

The Switch Fabric Module first installed functions as the primary module. For redundancy, you can install a redundant Switch Fabric Module. When two Switch Fabric Modules are installed at the same time, the module in the upper slot functions as the primary module, and the one in the lower slot functions as the backup. If you reset the module installed in the upper slot, the one in the lower slot becomes active.

No configuration is required for Switch Fabric Module redundancy. The module in the upper slot functions as the primary module and a redundant Switch Fabric Module in the lower slot automatically takes over if the primary module fails.

Forwarding Decisions for Layer 3-Switched Traffic

Either a PFC2 or a Distributed Feature Card (DFC) makes the forwarding decision for Layer 3-switched traffic as follows:

- A PFC2 makes all forwarding decisions for each packet that enter the switch through a module without a DFC.
- A DFC makes all forwarding decisions for each packet that enters the switch on a DFC-enabled module in these situations:
 - If the egress port is on the same module as the ingress port, the DFC forwards the packet locally (the packet never leaves the module).

- If the egress port is on a different fabric-enabled module, the DFC sends the packet across the SFM to the egress module, which sends it out the egress port.
- If the egress port is on a different nonfabric-enabled module, the DFC sends the packet across the SFM to the Supervisor Engine 2. The Supervisor Engine 2 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

When you install a Switch Fabric Module, the traffic is forwarded to and from modules in one of the following modes:

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

Table 6-1 shows the switching modes used with fabric-enabled and nonfabric-enabled modules installed.

Table 6-1 Switching Modes with Switch Fabric Module Installed

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

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

2. Displayed as fabric mode in **show** commands.

Configuring the Switch Fabric Module

These section describe configuring the Switch Fabric Module:

- [Configuring the Switching Mode, page 6-4](#)
- [Configuring Fabric-Required Mode, page 6-4](#)
- [Configuring an LCD Message, page 6-5](#)



Note

When you are in configuration mode you can enter EXEC mode-level commands by entering the **do** keyword before the EXEC mode-level command.

Configuring the Switching Mode

To configure the switching mode, perform this task:

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

When configuring the switching mode, note the following information:

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



Caution

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

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

Configuring Fabric-Required Mode

To configure fabric-required mode, which prevents all switching modules from operating unless there is a Switch Fabric Module installed, perform this task:

Command	Purpose
Router(config)# fabric required	Configures fabric-required mode, which prevents switching modules from operating without a switch fabric module.
Router(config)# no fabric required	Clears fabric-required mode.



Caution

If you enter the **fabric required** command on a switch that does not have a Switch Fabric Module installed, all modules except the supervisor engine turn off.

When configuring fabric-required mode, note the following information:

- If you boot the switch with fabric-required mode configured but without a Switch Fabric Module installed, only the supervisor engine receives power; no switching modules power up.
- When the switch is operating with fabric-required mode configured and a Switch Fabric Module installed, if you remove the switch fabric module or if it fails, the switch removes power from all switching modules; only the supervisor engine remains active.
- When the switch is operating with fabric-required mode configured and with redundant Switch Fabric Modules installed, if you remove both switch fabric modules or if both fail, the switch removes power from all switching modules; only the supervisor engine remains active.

Configuring an LCD Message

To configure a message for display on the LCD, perform this task:

Command	Purpose
Router(config)# fabric lcd-banner <i>d message d</i>	Configures a message for display on the LCD.
Router(config)# no fabric lcd-banner	Clears the message displayed on the LCD.

When configuring a message for display on the LCD, note the following information:

- The *d* parameter is a delimiting character. You cannot use the delimiting character in the message. The delimiter is a character of your choice—a pound sign (#), for example.
- You can use the following tokens, in the form \$(token), in the message text:
 - \$(hostname)—Displays the switch's host name.
 - \$(domain)—Displays the switch's domain name.

Monitoring the Switch Fabric Module

The Switch Fabric Module 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 Module:

- [Displaying the Module Information, page 6-6](#)
- [Displaying the Switch Fabric Module Redundancy Status, page 6-6](#)
- [Displaying Fabric Channel Switching Modes, page 6-6](#)
- [Displaying the Fabric Status, page 6-7](#)
- [Displaying the Fabric Utilization, page 6-7](#)
- [Displaying Fabric Errors, page 6-7](#)



Note

The Switch Fabric Module does not require any user configuration.

Displaying the Module Information

To display the module information, perform this task:

Command	Purpose
Router# show module { 5 6 7 8 }	Displays module information.

This example shows how to display module information:

```
Router# show module 5
Mod Ports Card Type                               Model                Serial No.
-----
  5      0 Switching Fabric Module                WS-C6500-SFM         SAD04420JR5

Mod MAC addresses                               Hw   Fw           Sw           Status
-----
  5  0001.0002.0003 to 0001.0002.0003   1.0  6.1(3)       6.2(0.97)    Ok
```

Displaying the Switch Fabric Module Redundancy Status

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

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

This example shows how to display the switch fabric module 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
bus-only mode is allowed
Module Slot      Switching Mode
1                Bus
2                Bus
3                DCEF
4                DCEF
5                No Interfaces
6                DCEF
```


Displaying the Fabric Status

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

Command	Purpose
Router# show fabric status [<i>slot_number</i> all]	Displays fabric status.

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

```
Router# show fabric status all
  slot    channel    module    fabric
  status      status
  1         0        OK        OK
  3         0        OK        OK
  3         1        OK        OK
  4         0        OK        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 [<i>slot_number</i> all]	Displays fabric utilization.

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

```
Router# show fabric utilization all
  slot    channel  Ingress %  Egress %
  1         0         0         0
  3         0         0         0
  3         1         0         0
  4         0         0         0
  4         1         0         0
  6         0         0         0
  6         1         0         0
  7         0         0         0
  7         1         0         0
Router#
```

Displaying Fabric Errors

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

Command	Purpose
Router# show fabric errors [<i>slot_number</i> all]	Displays fabric errors.

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

```
Router# show fabric errors
      slot      channel  module  module  module  fabric
                        crc      hbeat    sync    sync
      1          0        0         0         0         0
      3          0        0         0         0         0
      3          1        0         0         0         0
      4          0        0         0         0         0
      4          1        0         0         0         0
      6          0        0         0         0         0
      6          1        0         0         0         0
      7          0        0         0         0         0
      7          1        0         0         0         0
Router#
```



Configuring NSF with SSO Supervisor Engine Redundancy

This chapter describes how to configure supervisor engine redundancy using Cisco nonstop forwarding (NSF) with stateful switchover (SSO).



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- Release 12.2(18)SXD and later releases support nonstop forwarding (NSF) with stateful switchover (SSO) on Supervisor Engine 720 and Supervisor Engine 2.
- Release 12.2(17b)SXA, rebuilds of Release 12.2(17b)SXA, Release 12.2(17d)SXB, and rebuilds of Release 12.2(17d)SXB support SRM with SSO on Supervisor Engine 720 (see [Chapter 8](#), “Configuring SRM with SSO Supervisor Engine Redundancy”).
- Release 12.2(18)SXD and later releases do not support SRM with SSO.
- All releases support RPR and RPR+ (see [Chapter 9](#), “Configuring RPR and RPR+ Supervisor Engine Redundancy”).
- NSF with SSO does not support IPv6 multicast traffic. If you configure support for IPv6 multicast traffic, configure RPR or RPR+ redundancy.

This chapter consists of these sections:

- [Understanding NSF with SSO Supervisor Engine Redundancy, page 7-1](#)
- [Supervisor Engine Configuration Synchronization, page 7-9](#)
- [NSF Configuration Tasks, page 7-11](#)
- [Copying Files to the Redundant Supervisor Engine, page 7-20](#)

Understanding NSF with SSO Supervisor Engine Redundancy

These sections describe supervisor engine redundancy using NSF with SSO:

- [NSF with SSO Supervisor Engine Redundancy Overview, page 7-2](#)
- [SSO Operation, page 7-2](#)

- [NSF Operation, page 7-3](#)
- [Cisco Express Forwarding, page 7-3](#)
- [Multicast MLS NSF with SSO, page 7-4](#)
- [Routing Protocols, page 7-4](#)
- [NSF Benefits and Restrictions, page 7-8](#)

NSF with SSO Supervisor Engine Redundancy Overview



Note

- With Release 12.2(18)SXD and earlier releases, when a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.
- With a Supervisor Engine 720 and Release 12.2(18)SXE and later releases, 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)
- With a Supervisor Engine 2 and Release 12.2(18)SXD1 and later releases, if all the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on the redundant supervisor engine, which ensures that all modules are operating in dCEF mode. (CSCec05612)

Catalyst 6500 series switches support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. Cisco NSF works with SSO to minimize the amount of time a network is unavailable to its users following a switchover while continuing to forward IP packets. Catalyst 6500 series switches also support route processor redundancy (RPR), route processor redundancy plus (RPR+), and single router mode with stateful switchover (SRM with SSO) for redundancy. For information about these redundancy modes, see [Chapter 9, “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 switch, or is manually shut down for maintenance. This type of switchover ensures that Layer 2 traffic is not interrupted.

In networking devices running SSO, both supervisor engines must be running the same configuration so that the redundant supervisor engine is always ready to assume control following a fault on the active supervisor engine. SSO switchover also preserves FIB and adjacency entries and can forward Layer 3 traffic after a switchover. Configuration information and data structures are synchronized from the active

to the redundant supervisor engine at startup and whenever changes to the active supervisor engine configuration occur. Following an initial synchronization between the two supervisor engines, SSO maintains state information between them, including forwarding information.

During switchover, system control and routing protocol execution is transferred from the active supervisor engine to the redundant supervisor engine. The switch requires between 0 and 3 seconds to switchover from the active to the redundant supervisor engine.

NSF Operation

Cisco NSF always runs with SSO and provides redundancy for Layer 3 traffic. NSF works with SSO to minimize the amount of time that a network is unavailable to its users following a switchover. The main purpose of NSF is to continue forwarding IP packets following a supervisor engine switchover.

Cisco NSF is supported by the BGP, OSPF, 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 switch will continue to leak packets using the old cache for these protocols. The router builds the mroute cache and installs the shortcuts in hardware. After the new routes are learned, a timer is triggered to go through the database and purge the old flows.

**Note**

Multicast MLS NSF with SSO requires NSF support in the unicast protocols.

Routing Protocols

The routing protocols run only on the 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 its 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.
- The Hot Standby Routing Protocol (HSRP) is not SSO-aware, meaning state information is not maintained between the active and standby supervisor engine during normal operation. HSRP and SSO can coexist but both features work independently. Traffic that relies on HSRP may switch to the HSRP standby in the event of a supervisor switchover.
- The Gateway Load Balancing Protocol (GLBP) is not SSO-aware, meaning state information is not maintained between the active and standby supervisor engine during normal operation. GLBP and SSO can coexist but both features work independently. Traffic that relies on GLBP may switch to the GLBP standby in the event of a Supervisor switchover.

- The Virtual Redundancy Routing Protocols (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.
- Multiprotocol Label Switching (MPLS) is not supported with Cisco NSF with SSO; however, MPLS and NSF with SSO can coexist. If NSF with SSO is configured in the same chassis with MPLS, the failover performance of MPLS protocols will be at least equivalent to RPR+ while the supported NSF with SSO protocols still retain the additional benefits of NSF with SSO.
- All neighboring devices participating in BGP NSF must be NSF-capable and configured for BGP graceful restart.
- OSPF NSF for virtual links is not supported.
- All OSPF networking devices on the same network segment must be NSF-aware (running an NSF software image).
- For IETF IS-IS, all neighboring devices must be running an NSF-aware software image.
- Multicast NSF with SSO is supported by the Supervisor Engine 720 only.
- The underlying unicast protocols must be NSF-aware in order to use multicast NSF with SSO.

Supervisor Engine Configuration Synchronization

These sections describe supervisor engine configuration synchronization:

- [Supervisor Engine Redundancy Guidelines and Restrictions, page 7-9](#)
- [Redundancy Configuration Guidelines and Restrictions, page 7-10](#)



Note

Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine.

Supervisor Engine Redundancy Guidelines and Restrictions

These sections describe supervisor engine redundancy guidelines and restrictions:

- [Redundancy Configuration Guidelines and Restrictions, page 7-10](#)
- [Hardware Configuration Guidelines and Restrictions, page 7-10](#)
- [Configuration Mode Restrictions, page 7-11](#)

Redundancy Configuration Guidelines and Restrictions

These guidelines and restrictions apply to all redundancy modes:

- With Release 12.2(18)SXD and earlier releases, when a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.
- With a Supervisor Engine 720 and Release 12.2(18)SXE and later releases, 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)
- With a Supervisor Engine 2 and Release 12.2(18)SXD1 and later releases, if all the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on the redundant supervisor engine, which ensures that all modules are operating in dCEF mode. (CSCec05612)
- Supervisor engine redundancy does not provide supervisor engine mirroring or supervisor engine load balancing. Only one supervisor engine is active.
- Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine.
- Supervisor engine switchover takes place after the failed supervisor engine completes a core dump. A core dump can take up to 15 minutes. To get faster switchover time, disable core dump on the supervisor engines.

Hardware Configuration Guidelines and Restrictions

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

- Cisco IOS running on the supervisor engine and the 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 switch on its own, which means all supervisor engine resources are duplicated, including all Flash devices.
- Make separate console connections to each supervisor engine. Do not connect a Y cable to the console ports.
- Both supervisor engines must have the same system image (see the [“Copying Files to the Redundant Supervisor Engine”](#) section on page 7-20).

**Note**

If a newly installed redundant supervisor engine has the Catalyst operating system installed, remove the active supervisor engine and boot the switch with only the redundant supervisor engine installed. Follow the procedures in the current release notes to convert the redundant supervisor engine from the Catalyst operating system.

- The configuration register in the startup-config must be set to autoboot.

**Note**

There is no support for booting from the network.

Configuration Mode Restrictions

The following configuration restrictions apply during the startup synchronization process:

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

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

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

NSF Configuration Tasks

The following sections describe the configuration tasks for the NSF feature:

- [Configuring SSO, page 7-11](#)
- [Configuring Multicast MLS NSF with SSO, page 7-12](#)
- [Verifying Multicast NSF with SSO, page 7-13](#)
- [Configuring CEF NSF, page 7-13](#)
- [Verifying CEF NSF, page 7-13](#)
- [Configuring BGP NSF, page 7-14](#)
- [Verifying BGP NSF, page 7-14](#)
- [Configuring OSPF NSF, page 7-15](#)
- [Verifying OSPF NSF, page 7-15](#)
- [Configuring IS-IS NSF, page 7-16](#)
- [Verifying IS-IS NSF, page 7-17](#)

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.



Note

The **sso** keyword is supported in Release 12.2(17b)SXA and later releases.

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 <i>time</i>	Specifies the maximum time to wait for protocol convergence; valid values are from 0 to 3600 seconds.

	Command	Purpose
Step 3	Router(config)# mls ip multicast sso leak interval	Specifies the packet leak interval; valid values are from 0 to 3600 seconds. For PIM sparse mode and PIM dense mode this is the period of time after which packet leaking for existing PIM sparse mode and PIM dense mode multicast forwarding entries should be completed.
Step 4	Router(config)# mls ip multicast sso leak percentage	Specifies the percentage of multicast flows; valid values are from 1 to 100 percent. The value represents the percentage of the total number of existing PIM sparse mode and PIM dense mode multicast flows that should be flagged for packet leaking.

Verifying Multicast NSF with SSO

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

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

Configuring CEF NSF

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

Verifying CEF NSF

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

```
router# show cef state

CEF Status [RP]
CEF enabled/running
dCEF enabled/running
CEF switching enabled/running
CEF default capabilities:
Always FIB switching:      yes
Default CEF switching:    yes
Default dCEF switching:    yes
Update HWIDB counters:    no
```

```

Drop multicast packets:    no
.
.
.
CEF NSF capable:          yes
IPC delayed func on SSO:  no
RRP state:
I am standby RRP:         no
My logical slot:          0
RF PeerComm:              no

```

Configuring BGP NSF



Note

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

To configure BGP for NSF, perform this task (repeat this procedure on each of the BGP NSF peer devices):

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# router bgp <i>as-number</i>	Enables a BGP routing process, which places the router in router configuration mode.
Step 3	Router(config-router)# bgp graceful-restart	<p>Enables the BGP graceful restart capability, starting BGP NSF.</p> <p>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.</p> <p>Use this command on the restarting router and all of its peers.</p>

Verifying BGP NSF

To verify BGP NSF, you must check that the graceful restart function is configured on the SSO-enabled networking device and on the neighbor devices. To verify, follow these steps:

- Step 1** Verify that “bgp graceful-restart” appears in the BGP configuration of the SSO-enabled router by entering the **show running-config** command:

```

Router# show running-config
.
.
.
router bgp 120
.
.
.
bgp graceful-restart
neighbor 10.2.2.2 remote-as 300

```


·
·
·

Step 2 Repeat step 1 on each of the BGP neighbors.

Step 3 On the SSO device and the neighbor device, verify that the graceful restart function is shown as both advertised and received, and confirm the address families that have the graceful restart capability. If no address families are listed, then BGP NSF also will not occur:

```
router#show ip bgp neighbors x.x.x.x

BGP neighbor is 192.168.2.2, remote AS YY, external link
  BGP version 4, remote router ID 192.168.2.2
  BGP state = Established, up for 00:01:18
  Last read 00:00:17, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh:advertised and received(new)
    Address family IPv4 Unicast:advertised and received
    Address famiiy IPv4 Multicast:advertised and received
    Graceful Restart Capabilty:advertised and received
    Remote Restart timer is 120 seconds
    Address families preserved by peer:
      IPv4 Unicast, IPv4 Multicast
  Received 1539 messages, 0 notifications, 0 in queue
  Sent 1544 messages, 0 notifications, 0 in queue
  Default minimum time between advertisement runs is 30 seconds
```

Configuring OSPF NSF



Note

All peer devices participating in OSPF NSF must be made OSPF NSF-aware, which happens automatically once you install an NSF software image on the device.

To configure OSPF NSF, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# router ospf <i>processID</i>	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 [<i>tag</i>]	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 [<i>minutes</i>]	(Optional) Specifies the minimum time between NSF restart attempts. The default time between <i>consecutive</i> NSF restart attempts is 5 minutes.

	Command	Purpose
Step 5	Router(config-router)# nsf t3 { manual [<i>seconds</i>] 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 <i>seconds</i>	(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 <i>as-number</i>	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
```



Configuring SRM with SSO Supervisor Engine Redundancy

This chapter describes how to configure Supervisor Engine 720 redundancy using single router mode (SRM) with stateful switchover (SSO).



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- Release 12.2(18)SXD and later releases support nonstop forwarding (NSF) with stateful switchover (SSO) on Supervisor Engine 720 and Supervisor Engine 2 (see [Chapter 7, “Configuring NSF with SSO Supervisor Engine Redundancy”](#)).
- Release 12.2(17b)SXA, rebuilds of Release 12.2(17b)SXA, Release 12.2(17d)SXB, and rebuilds of Release 12.2(17d)SXB support SRM with SSO on Supervisor Engine 720.
- Supervisor Engine 2 does not support SRM with SSO.
- Release 12.2(18)SXD and later releases do not support SRM with SSO.
- All releases support RPR and RPR+ (see [Chapter 9, “Configuring RPR and RPR+ Supervisor Engine Redundancy”](#)).

This chapter consists of these sections:

- [Understanding SRM with SSO, page 8-1](#)
- [Supervisor Engine 720 Redundancy Guidelines and Restrictions, page 8-4](#)
- [Configuring Supervisor Engine 720 Redundancy, page 8-6](#)
- [Copying Files to the Redundant Supervisor Engine, page 8-8](#)

Understanding SRM with SSO

These sections describe Supervisor Engine 720 redundancy using SRM with SSO:

- [Supervisor Engine Redundancy Overview, page 8-2](#)
- [SRM with SSO Operation, page 8-2](#)
- [Supervisor Engine 720 Configuration Synchronization, page 8-4](#)

Supervisor Engine Redundancy Overview

**Note**

With Release 12.2(18)SXD and earlier releases, the two Gigabit Ethernet interfaces on a redundant Supervisor Engine 720 are always active.

Catalyst 6500 series switches support fault resistance by allowing a redundant Supervisor Engine 720 to take over if the primary Supervisor Engine 720 fails. Catalyst 6500 series switches support these redundancy modes:

- Route processor redundancy (RPR):
 - Supports a switchover time of 2 or more minutes (see [Chapter 7, “Configuring NSF with SSO Supervisor Engine Redundancy”](#)).
 - Supported on Supervisor Engine 720 and Supervisor Engine 2.
- Route processor redundancy plus (RPR+):
 - Supports a switchover time of 30 or more seconds (see [Chapter 7, “Configuring NSF with SSO Supervisor Engine Redundancy”](#)).
 - Supported on Supervisor Engine 720 and Supervisor Engine 2.
- Single router mode with stateful switchover (SRM with SSO):
 - Supports a switchover time of 1 or more seconds.
 - Supported on Supervisor Engine 720.
 - Not supported on Supervisor Engine 2.
- Nonstop Forwarding (NSF) with SSO:
 - Supported on Supervisor Engine 720 and Supervisor Engine 2.
 - See [Chapter 7, “Configuring NSF with SSO Supervisor Engine Redundancy.”](#)

The following events cause a switchover:

- A hardware failure on the active Supervisor Engine 720
- Clock synchronization failure between Supervisor Engine 720s
- A manual switchover

SRM with SSO Operation

**Caution**

To avoid reloads with software releases where caveat CSCed17605 is not resolved, do not configure the SSO with SRM redundancy mode with a WS-SVC-IPSEC-1 module installed. Caveat CSCed17605 is resolved in Release 12.2(17d)SXB and later releases.

**Note**

SRM with SSO redundancy mode does not support MPLS. If you configure MPLS, use the RPR+ redundancy mode.

When the switch is powered on, SRM with SSO runs between the two Supervisor Engine 720s. The Supervisor Engine 720 that boots first becomes the active Supervisor Engine 720. The Multilayer Switch Feature Card 3 MSFC3 and Policy Feature Card 3 PFC3 become fully operational. The configuration of the redundant Supervisor Engine 720 and MSFC3 is exactly the same as the active Supervisor Engine 720 and MSFC3. Processes such as routing protocols are created on both the active MSFC3 and the redundant MSFC3.

The redundant Supervisor Engine 720 is fully initialized and configured, which shortens the switchover time. The active Supervisor Engine 720 checks the image version of the redundant Supervisor Engine 720 when the redundant Supervisor Engine 720 comes online. If the image on the redundant Supervisor Engine 720 does not match the image on the active Supervisor Engine 720, RPR redundancy mode is used.

If the active Supervisor Engine 720 or MSFC3 fails, the redundant Supervisor Engine 720 and MSFC3 become active. The newly active Supervisor Engine 720 uses the existing PFC3 Layer 3 switching information to forward traffic while the newly active MSFC3 builds its routing table.

SRM with SSO supports the following features:

- Auto-startup and bootvar synchronization between active and redundant Supervisor Engine 720s.
- Hardware signals that detect and decide the active or redundant status of Supervisor Engine 720s.
- Clock synchronization every 60 seconds from the active to the redundant Supervisor Engine 720.
- An operational Supervisor Engine 720 present in place of the failed unit becomes the redundant Supervisor Engine 720.
- 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 720, installed modules are not reloaded during a switchover.
- Online insertion and removal (OIR) of the redundant Supervisor Engine 720—SRM with SSO allows OIR of the redundant Supervisor Engine 720 for maintenance. When the redundant Supervisor Engine 720 is inserted, the active Supervisor Engine 720 detects its presence and begins to transition the redundant Supervisor Engine 720 to a fully initialized state.
- Synchronization of OIR events
- Manual user-initiated switchover using the **redundancy force-switchover** command

SRM with SSO supports stateful switchover of these Layer 2 features:

- Link negotiation
- Flow control
- VLANs
- VTP
- VLAN trunks
- DTP
- STP
- PAgP/LACP
- CDP
- UDLD
- SPAN/RSPAN
- Voice VLAN and inline power
- 802.1x

- 802.1Q
- Port security
- MAC move notification
- Traffic storm control
- IEEE 802.1Q tunneling
- Layer 2 protocol tunneling
- ARP
- Bridge groups
- IGMP snooping

Supervisor Engine 720 Configuration Synchronization

**Note**

Configuration changes made through SNMP are not synchronized to the redundant Supervisor Engine 720. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active Supervisor Engine 720 to trigger synchronization of the startup-config file on the redundant Supervisor Engine 720 and with SRM with SSO or RPR+, reload the redundant Supervisor Engine 720 and MSFC3.

With SRM with SSO mode, the following operations trigger configuration synchronization:

- When a redundant Supervisor Engine 720 first comes online, the startup-config file is copied from the active Supervisor Engine 720 to the redundant Supervisor Engine 720. This synchronization overwrites any existing startup configuration file on the redundant Supervisor Engine 720.
- When configuration changes occur during normal operation, redundancy performs an incremental synchronization from the active Supervisor Engine 720 to the redundant Supervisor Engine 720. Redundancy synchronizes user-entered CLI commands incrementally line-by-line from the active Supervisor Engine 720 to the redundant Supervisor Engine 720.

Even though the redundant Supervisor Engine 720 is fully initialized, it only interacts with the active Supervisor Engine 720 to receive incremental changes to the configuration files as they occur. You cannot enter CLI commands on the redundant Supervisor Engine 720.

Supervisor Engine 720 Redundancy Guidelines and Restrictions

These sections describe Supervisor Engine 720 redundancy guidelines and restrictions:

- [Redundancy Guidelines and Restrictions, page 8-5](#)
- [Hardware Configuration Guidelines and Restrictions, page 8-5](#)
- [Configuration Mode Restrictions, page 8-6](#)

Redundancy Guidelines and Restrictions

When configuring SRM with SSO, follow these guidelines and restrictions:

- With Release 12.2(18)SXD and earlier releases, the two Gigabit Ethernet interfaces on a redundant Supervisor Engine 720 are always active.
- Supervisor engine redundancy does not provide Supervisor Engine 720 mirroring or Supervisor Engine 720 load balancing. Only one Supervisor Engine 720 is active.
- Configuration changes made through SNMP are not synchronized to the redundant Supervisor Engine 720. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active Supervisor Engine 720 to trigger synchronization of the startup-config file on the redundant Supervisor Engine 720 and with SRM with SSO, reload the redundant Supervisor Engine 720 and MSFC3.
- Supervisor Engine 720 switchover takes place after the failed Supervisor Engine 720 completes a core dump. A core dump can take up to 15 minutes. To get faster switchover time, disable core dump on the Supervisor Engine 720s.
- Both Supervisor Engine 720s must run the same version of Cisco IOS software. If the Supervisor Engine 720s are not running the same version of Cisco IOS software, the redundant Supervisor Engine 720 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 switch that has a redundant Supervisor Engine 720.
- Before installing a redundant Supervisor Engine 720, 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 15, “Configuring VLANs”](#)).

Hardware Configuration Guidelines and Restrictions

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

- Cisco IOS running on the Supervisor Engine 720 and the MSFC3 supports redundant configurations where the Supervisor Engine 720s and MSFC3 routers are identical. If they are not identical, one will boot first and become active and hold the other Supervisor Engine 720 and MSFC3 in a reset condition.
- Each Supervisor Engine 720 must have the resources to run the switch on its own, which means all Supervisor Engine 720 resources are duplicated, including all flash devices.
- Make separate console connections to each Supervisor Engine 720. Do not connect a Y cable to the console ports.
- Both Supervisor Engine 720s must have the same system image (see the [“Copying Files to the Redundant Supervisor Engine”](#) section on page 8-8).

**Note**

If a newly installed redundant Supervisor Engine 720 has the Catalyst operating system installed, remove the active Supervisor Engine 720 and boot the switch with only the redundant Supervisor Engine 720 installed. Follow the procedures in the current release notes to convert the redundant Supervisor Engine 720 from the Catalyst operating system.

- The configuration register in the startup-config must be set to autoboot (see the [“Modifying the Boot Field”](#) section on page 3-23).

**Note**

There is no support for booting from the network.

With Release 12.2(17b)SXA, rebuilds of Release 12.2(17b)SXA, Release 12.2(17d)SXB, and rebuilds of Release 12.2(17d)SXB, if these requirements are met, the Supervisor Engine 720 functions in SRM with SSO mode by default.

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 720 switchover, these configuration changes are lost.

Configuring Supervisor Engine 720 Redundancy

These sections describe how to configure Supervisor Engine 720 redundancy:

- [Configuring SRM with SSO Redundancy](#), page 8-6
- [Configuring the SRM with SSO Route Convergence Interval](#), page 8-7
- [Synchronizing the Supervisor Engine Configurations](#), page 8-8
- [Displaying the Redundancy States](#), page 8-8

Configuring SRM with SSO Redundancy

To configure SRM with SSO redundancy, perform this task:

	Command	Purpose
Step 1	Router(config)# redundancy	Enters redundancy configuration mode.
Step 2	Router(config-red)# mode sso	Configures SRM with SSO. When this command is entered, the redundant Supervisor Engine 720 is reloaded and begins to work in SRM with SSO mode.
Step 3	Router# show running-config	Verifies that SRM with SSO is enabled.
Step 4	Router# show redundancy states	Displays the operating redundancy mode.

When configuring redundancy, note the following:

- The **sso** keyword is supported on Supervisor Engine 720 in Release 12.2(17b)SXA, rebuilds of Release 12.2(17b)SXA, Release 12.2(17d)SXB, and rebuilds of Release 12.2(17d)SXB.
- The **sso** keyword is not supported on Supervisor Engine 2 in Release 12.2(17b)SXA, rebuilds of Release 12.2(17b)SXA, Release 12.2(17d)SXB, and rebuilds of Release 12.2(17d)SXB.

This example shows how to configure the system for SRM with 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) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
    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 the SRM with SSO Route Convergence Interval

After a switchover, SRM with SSO uses the existing PFC3 Layer 3 switching information to forward traffic while the newly active MSFC3 builds its routing table. You can configure how long the newly active PFC3 waits before purging the existing PFC3 Layer 3 switching information.

To configure the SRM with SSO route convergence interval, perform this task:

	Command	Purpose
Step 1	Router(config)# redundancy	Enters redundancy configuration mode.
Step 2	Router(config-red)# route-converge-interval <i>interval_seconds</i>	Configures the SRM with SSO route convergence interval. Valid values for <i>interval_seconds</i> are from 60 to 3600 seconds.
	Router(config-red)# no route-converge-interval	Returns to the default route convergence interval (120 seconds).
Step 3	Router# show running-config	Verifies the configuration.

Synchronizing the Supervisor Engine Configurations

During normal operation, the startup-config and config-registers configuration are synchronized by default between the two Supervisor Engine 720s. In a switchover, the new active Supervisor Engine 720 uses the current 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#
```

Copying Files to the Redundant Supervisor Engine

Use the following command to copy a file to the **disk0:** device on a redundant Supervisor Engine 720:

```
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 720:

```
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 MSFC3:

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



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 *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL: <http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- All releases support RPR and RPR+.
- With Release 12.2(18)SXE and later releases, RPR and RPR+ support IPv6 multicast traffic.
- Release 12.2(18)SXD and later releases support nonstop forwarding (NSF) with stateful switchover (SSO) on Supervisor Engine 720 and Supervisor Engine 2 (see [Chapter 7, “Configuring NSF with SSO Supervisor Engine Redundancy”](#)).
- Release 12.2(17b)SXA, rebuilds of Release 12.2(17b)SXA, Release 12.2(17d)SXB, and rebuilds of Release 12.2(17d)SXB support SRM with SSO on Supervisor Engine 720 (see [Chapter 8, “Configuring SRM with SSO Supervisor Engine Redundancy”](#)).
- Release 12.2(18)SXD and later releases do not support SRM with SSO.

This chapter consists of these sections:

- [Understanding RPR and RPR+, page 9-1](#)
- [Supervisor Engine Redundancy Guidelines and Restrictions, page 9-4](#)
- [Configuring Supervisor Engine Redundancy, page 9-7](#)
- [Performing a Fast Software Upgrade, page 9-9](#)
- [Copying Files to the Redundant Supervisor Engine, page 9-11](#)

Understanding RPR and RPR+

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

- [Supervisor Engine Redundancy Overview, page 9-2](#)
- [RPR Operation, page 9-2](#)

- [RPR+ Operation, page 9-3](#)
- [Supervisor Engine Configuration Synchronization, page 9-4](#)

Supervisor Engine Redundancy Overview



Note

- With Release 12.2(18)SXD and earlier releases, when a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.
- With a Supervisor Engine 720 and Release 12.2(18)SXE and later releases, 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)
- With a Supervisor Engine 2 and Release 12.2(18)SXD1 and later releases, if all the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on the redundant supervisor engine, which ensures that all modules are operating in dCEF mode. (CSCec05612)

Catalyst 6500 series switches support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. Catalyst 6500 series switches 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 9-9.](#))

When the switch is powered on, RPR runs between the two supervisor engines. The supervisor engine that boots first becomes the RPR active supervisor engine. The Multilayer Switch Feature Card and Policy Feature Card become fully operational. The 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 switch 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 9-4](#)
- [RPR+ Supervisor Engine Configuration Synchronization, page 9-4](#)

**Note**

Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine 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, page 9-5](#)
- [RPR+ Guidelines and Restrictions, page 9-5](#)
- [Hardware Configuration Guidelines and Restrictions, page 9-6](#)
- [Configuration Mode Restrictions, page 9-6](#)

Redundancy Guidelines and Restrictions

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

- With Release 12.2(18)SXD and earlier releases, when a redundant supervisor engine is in standby mode, the two Gigabit Ethernet interfaces on the redundant supervisor engine are always active.
- With a Supervisor Engine 720 and Release 12.2(18)SXE and later releases, 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)
- With a Supervisor Engine 2 and Release 12.2(18)SXD1 and later releases, if all the installed switching modules have DFCs, enter the **fabric switching-mode allow dcef-only** command to disable the Ethernet ports on the redundant supervisor engine, which ensures that all modules are operating in dCEF mode. (CSCec05612)
- Supervisor engine redundancy does not provide supervisor engine mirroring or supervisor engine load balancing. Only one supervisor engine is active.
- Configuration changes made through SNMP are not synchronized to the redundant supervisor engine. After you configure the switch through SNMP, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine 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 switch 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 switch that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- Supervisor engine redundancy does not support configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy (see [Chapter 15, “Configuring VLANs”](#)).

Hardware Configuration Guidelines and Restrictions

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

- Cisco IOS running on the supervisor engine and the 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 switch on its own, which means all supervisor engine resources are duplicated, including all Flash devices.
- Make separate console connections to each supervisor engine. Do not connect a Y cable to the console ports.
- Both supervisor engines must have the same system image (see the [“Copying Files to the Redundant Supervisor Engine”](#) section on page 9-11).



Note

If a newly installed redundant supervisor engine has the Catalyst operating system installed, remove the active supervisor engine and boot the switch with only the redundant supervisor engine installed. Follow the procedures in the current release notes to convert the redundant supervisor engine from the Catalyst operating system.

- The configuration register in the startup-config must be set to autoboot (see the [“Modifying the Boot Field”](#) section on page 3-23).



Note

There is no support for booting from the network.

With releases earlier than Release 12.2(17b)SXA, if these requirements are met, the switch functions in RPR+ mode by default.

Configuration Mode Restrictions

The following configuration restrictions apply during the startup synchronization process:

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

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

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

Configuring Supervisor Engine Redundancy

These sections describe how to configure supervisor engine redundancy:

- [Configuring Redundancy, page 9-7](#)
- [Synchronizing the Supervisor Engine Configurations, page 9-8](#)
- [Displaying the Redundancy States, page 9-9](#)

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.

To manually synchronize the configurations used by the two supervisor engines, perform this task on the active supervisor engine:


	Command	Purpose
Step 1	Router(config)# redundancy	Enters redundancy configuration mode.
Step 2	Router(config-red)# main-cpu	Enters main-cpu configuration submode.
Step 3	Router(config-r-mc)# auto-sync { startup-config config-register bootvar standard }	Synchronizes the configuration elements.
Step 4	Router(config-r-mc)# end	Returns to privileged EXEC mode.
Step 5	Router# copy running-config startup-config	Forces a manual synchronization of the configuration files in NVRAM. Note This step is not required to synchronize the running configuration file in DRAM.



Note The **auto-sync standard** command does not synchronize the boot variables.

This example shows how to reenable the default automatic synchronization feature using the **auto-sync standard** command to synchronize the startup-config and config-register configuration of the active supervisor engine with the redundant supervisor engine:

```
Router(config)# redundancy
Router(config-red)# main-cpu
Router(config-r-mc)# auto-sync standard
Router(config-r-mc)# auto-sync bootvar
Router(config-r-mc)# end
Router# copy running-config startup-config
```



Note To manually synchronize only individual elements of the standard auto-sync configuration, disable the default automatic synchronization feature.

This example shows how to disable default automatic synchronization and only allow automatic synchronization of the config-registers of the active supervisor engine to the redundant supervisor engine while disallowing synchronization of the startup configuration:

```
Router(config)# redundancy
Router(config-red)# main-cpu
Router(config-r-mc)# no auto-sync standard
Router(config-r-mc)# auto-sync config-register
Router(config-r-mc)# end
Router# copy running-config startup-config
```

Displaying the Redundancy States

To display the redundancy states, perform this task:

Command	Purpose
Router# show redundancy states	Displays the redundancy states.

This example shows how to display the redundancy states:

```
Router# show redundancy states
my state = 13 -ACTIVE
    peer state = 8  -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Route Processor Redundancy Plus
Redundancy Mode (Configured)  = Route Processor Redundancy Plus
    Split Mode = Disabled
    Manual Swact = Enabled
    Communications = Up

    client count = 11
    client_notification_TMR = 30000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
        keep_alive threshold = 18
        RF debug mask = 0x0

Router#
```

Performing a Fast Software Upgrade

The fast software upgrade (FSU) procedure supported by RPR allows you to upgrade the Cisco IOS image on the supervisor engines without reloading the system.



Note

If you are performing a first-time upgrade to RPR from EHSA, you must reload both supervisor engines. FSU from EHSA is not supported.

To perform an FSU, perform this task:

	Command	Purpose
Step 1	<pre>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</pre>	Copies the new Cisco IOS image to bootflash on both supervisor engines.
Step 2	<pre>Router# config terminal Router(config)# config-register 0x2102 Router(config)# boot system flash device:file_name</pre>	Configures the supervisor engines to boot the new image.
Step 3	<pre>Router# copy running-config start-config</pre>	Saves the configuration.
Step 4	<pre>Router# hw-module {module num} reset</pre>	<p>Reloads the redundant supervisor engine and brings it back online (running the new version of the Cisco IOS software).</p> <p>Note Before reloading the redundant supervisor engine, make sure you wait long enough to ensure that all configuration synchronization changes have completed.</p>
Step 5	<pre>Router# redundancy force-switchover</pre>	<p>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.</p> <p>The old active supervisor engine reboots with the new image and becomes the redundant supervisor engine.</p> <p>Note To perform an EHSA to RPR FSU, use the reload command in Step 5.</p>

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
```




Configuring Interfaces

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

- [Understanding Interface Configuration, page 10-1](#)
- [Using the Interface Command, page 10-2](#)
- [Configuring a Range of Interfaces, page 10-4](#)
- [Defining and Using Interface-Range Macros, page 10-5](#)
- [Configuring Optional Interface Features, page 10-6](#)
- [Understanding Online Insertion and Removal, page 10-16](#)
- [Monitoring and Maintaining Interfaces, page 10-16](#)
- [Checking the Cable Status Using the TDR, page 10-19](#)



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- The Release 12.2 publications at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/index.htm>

Understanding Interface Configuration

Many features in the software are enabled on a per-interface basis. When you enter the **interface** command, you must specify the following information:

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

**Note**

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

- Slot number—The slot in which the module is installed. On the Catalyst 6500 series switch, slots are numbered starting with 1, from top to bottom.
- Port number—The physical port number on the module. On the Catalyst 6500 series switch, the port numbers always begin with 1. When facing the rear of the switch, ports are numbered from the left to the right.

You can identify ports from the physical location. You also can use **show** commands to display information about a specific port, or all the ports.

Using the Interface Command

**Note**

You use the commands described in this section to configure both physical ports and logical interfaces.

These procedures apply to all interface configuration processes. Begin the interface configuration process in global configuration mode. To use the interface command, follow these steps:

- Step 1** Enter the **configure terminal** command at the privileged EXEC prompt to enter global configuration mode:

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

- Step 2** In the global configuration mode, enter the **interfaces** command. Identify the interface type and the number of the connector or interface card.

The following example shows how to select Fast Ethernet, slot 5, interface 1:

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

- Step 3** Enter the **show interfaces** EXEC command to see a list of all interfaces that are installed. A report is provided for each interface that the device supports, as shown in this display:

```
Router# show interfaces fastethernet 5/48
FastEthernet5/48 is up, line protocol is up
  Hardware is C6k 100Mb 802.3, address is 0050.f0ac.3083 (bia 0050.f0ac.3083)
  Internet address is 172.20.52.18/27
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
    4834677 packets input, 329545368 bytes, 0 no buffer
    Received 4796465 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
```

```

51926 packets output, 15070051 bytes, 0 underruns
0 output errors, 2 collisions, 2 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
Router#

```

Step 4 Enter the **show hardware EXEC** command to see a list of the system software and hardware:

```

Router# show hardware
Cisco Internetwork Operating System Software
IOS (tm) C6sup2_rp Software (C6sup2_rp-JSV-M), Version 12.1(5c)EX, EARLY DEPLOY
Synced to mainline version: 12.1(5c)
TAC:Home:Software:Ios General:CiscoIOSRoadmap:12.1
Copyright (c) 1986-2001 by Cisco Systems, Inc.
Compiled Wed 28-Mar-01 17:52 by hqluong
Image text-base: 0x30008980, data-base: 0x315D0000

ROM: System Bootstrap, Version 12.1(3r)E2, RELEASE SOFTWARE (fc1)
BOOTFLASH: C6sup2_rp Software (C6sup2_rp-JSV-M), Version 12.1(5c)EX, EARLY DEPL

Router uptime is 2 hours, 55 minutes
System returned to ROM by power-on (SP by power-on)
Running default software

cisco Catalyst 6000 (R7000) processor with 114688K/16384K bytes of memory.
Processor board ID SAD04430J9K
R7000 CPU at 300Mhz, Implementation 39, Rev 2.1, 256KB L2, 1024KB L3 Cache
Last reset from power-on
Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
1 Virtual Ethernet/IEEE 802.3 interface(s)
48 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)
381K bytes of non-volatile configuration memory.

16384K bytes of Flash internal SIMM (Sector size 512K).
Configuration register is 0x2

Router#

```

Step 5 To begin configuring Fast Ethernet port 5/5, enter the **interface** keyword, interface type, and slot number/port number at the privileged EXEC prompt, as shown in the following example:

```

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/5
Router(config-if)#

```



Note You do not need to add a space between the interface type and interface number. For example, in the preceding line you can specify either *fastethernet 5/5* or *fastethernet5/5*.

Step 6 After each **interface** command, enter the interface configuration commands your particular interface requires.

The commands you enter define the protocols and applications that will run on the interface. The commands are collected and applied to the **interface** command until you enter another **interface** command or press **Ctrl-Z** to get out of interface configuration mode and return to privileged EXEC mode.

- Step 7** After you configure an interface, check its status by using the EXEC **show** commands listed in [“Monitoring and Maintaining Interfaces” section on page 10-16](#).

Configuring a Range of Interfaces

The interface-range configuration mode allows you to configure multiple interfaces with the same configuration parameters. After you enter the interface-range configuration mode, all command parameters you enter are attributed to all interfaces within that range until you exit out of the interface-range configuration mode.

To configure a range of interfaces with the same configuration, perform this task:

Command	Purpose
Router(config)# [no] interface range { {vlan <i>vlan_ID</i> - <i>vlan_ID</i> [, vlan <i>vlan_ID</i> - <i>vlan_ID</i>]} { <i>type</i> ¹ <i>slot/port</i> - <i>port</i> [, <i>type</i> ¹ <i>slot/port</i> - <i>port</i>]} { <i>macro_name</i> [, <i>macro_name</i>]}}	Selects the range of interfaces to be configured.

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

When configuring a range of interfaces, note the following information:

- For information about macros, see the [“Defining and Using Interface-Range Macros” section on page 10-5](#).
- You can enter up to five comma-separated ranges.
- You are not required to enter spaces before or after the comma.
- With releases earlier than Release 12.2(18)SXE, you must add a space between the interface numbers and the dash when using the **interface range** command. For example, **interface range fastethernet 1 - 5** is valid syntax; **interface range fastethernet 1-5** is invalid.
- With Release 12.2(18)SXE and later releases, you do not need to add a space between the interface numbers and the dash when using the **interface range** command.
- With releases earlier than Release 12.2(18)SXD, the **no interface range** command does not support VLAN interfaces.
- With Release 12.2(18)SXD and later releases, the **no interface range** command supports VLAN interfaces.
- With releases earlier than Release 12.2(18)SXD, for VLAN interfaces, the **interface range** command supports only those VLAN interfaces for which Layer 2 VLANs have been created with the **interface vlan** command (the **show running-configuration** command displays the configured VLAN interfaces). The **interface range** command does not support VLAN interfaces that are not displayed by the **show running-configuration** command.
- With Release 12.2(18)SXD and later releases, the **interface range** command supports VLAN interfaces for which Layer 2 VLANs have not been created with the **interface vlan** command.



Note

The link state messages (LINK-3-UPDOWN and LINEPROTO-5-UPDOWN) are disabled by default. Enter the **logging event link status** command on each interface where you want the messages enabled.

This example shows how to reenoble 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 reenoble 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 <i>macro_name</i> { vlan <i>vlan_ID</i> - <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> - <i>port</i> } [, { <i>type</i> ¹ <i>slot/port</i> - <i>port</i> }]	Defines the interface-range macro and save it in NVRAM.
Router(config)# no define interface-range <i>macro_name</i>	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 <i>macro_name</i>	Selects the interface range to be configured using the values saved in a named interface-range macro.

This example shows how to change to the interface-range configuration mode using the interface-range macro `enet_list`:

```
Router(config)# interface range macro enet_list
Router(config-if)#
```

Configuring Optional Interface Features

These sections describe optional interface features:

- [Configuring Ethernet Interface Speed and Duplex Mode, page 10-7](#)
- [Configuring Jumbo Frame Support, page 10-10](#)
- [Configuring IEEE 802.3Z Flow Control, page 10-13](#)
- [Configuring the Port Debounce Timer, page 10-14](#)
- [Adding a Description for an Interface, page 10-15](#)

Configuring Ethernet Interface Speed and Duplex Mode

These sections describe how to configure Ethernet port speed and duplex mode:

- [Speed and Duplex Mode Configuration Guidelines, page 10-7](#)
- [Configuring the Ethernet Interface Speed, page 10-7](#)
- [Setting the Interface Duplex Mode, page 10-8](#)
- [Configuring Link Negotiation on Gigabit Ethernet Ports, page 10-8](#)
- [Displaying the Speed and Duplex Mode Configuration, page 10-9](#)

Speed and Duplex Mode Configuration Guidelines

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

- If you set the Ethernet port speed to auto, the switch automatically sets the duplex mode to auto.
- If you enter the **no speed** command, the switch automatically configures both speed and duplex to auto.
- If you configure an Ethernet port speed to a value other than auto (for example, 10, 100, or 1000 Mbps), configure the connecting port to match. Do not configure the connecting port to negotiate the speed.
- If you manually configure the Ethernet port speed to either 10 Mbps or 100 Mbps, the switch prompts you to also configure the duplex mode on the port.



Note

Catalyst 6500 series switches cannot automatically negotiate Ethernet port speed and duplex mode if the connecting port is configured to a value other than auto.



Caution

Changing the Ethernet port speed and duplex mode configuration might shut down and reenables 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 <i>slot/port</i>	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).

Release 12.2(17a)SX and later releases support the **10 100 1000** keywords after the **auto** keyword. When configuring the port speed for a 10/100/1000-Mbps Ethernet port with Release 12.2(17a)SX and later releases, 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-Mbps 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 <i>slot/port</i>	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 10-1](#) shows the four possible link negotiation configurations and the resulting link status for each configuration.

Table 10-1 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 Router(config-if)# no speed nonegotiate	Disables link negotiation. 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 type¹ 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 10-10](#)
- [Configuring MTU Sizes, page 10-12](#)



Caution

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

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

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



Note

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

Understanding Jumbo Frame Support

These sections describe jumbo frame support:

- [Jumbo Frame Support Overview, page 10-10](#)
- [Ethernet Ports, page 10-11](#)
- [VLAN Interfaces, page 10-12](#)

Jumbo Frame Support Overview

A jumbo frame is a frame larger than the default Ethernet size. You enable jumbo frame support by configuring a larger-than-default maximum transmission unit (MTU) size on a port or VLAN interface and configuring the global LAN port MTU size.



Note

- Jumbo frame support fragments routed traffic in software on the 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 10-13).

Bridged and Routed Traffic Size Check at Ingress Gigabit Ethernet Ports

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

Routed Traffic Size Check on the PFC

For traffic that needs to be routed, Jumbo frame support on the PFC compares traffic sizes to the configured MTU sizes and provides Layer 3 switching for jumbo traffic between interfaces configured with MTU sizes large enough to accommodate the traffic. Between interfaces that are not configured with large enough MTU sizes, if the “do not fragment bit” is not set, the PFC sends the traffic to the 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 10-13).

Ethernet Ports

These sections describe configuring nondefault MTU sizes on Ethernet ports:

- [Ethernet Port Overview, page 10-11](#)
- [Layer 3 Ethernet Ports, page 10-12](#)
- [Layer 2 Ethernet Ports, page 10-12](#)

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 10-13](#)).

Layer 2 Ethernet Ports

On a Layer 2 port, you can only configure an MTU size that matches the global LAN port MTU size (see the [“Configuring the Global Egress LAN Port MTU Size” section on page 10-13](#)).

VLAN Interfaces

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

Configuring MTU Sizes

These sections describe how to configure MTU sizes:

- [Configuring MTU Sizes, page 10-12](#)
- [Configuring the Global Egress LAN Port MTU Size, page 10-13](#)

Configuring the MTU Size

To configure the MTU size, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{ vlan <i>vlan_ID</i> {{ <i>type</i> ¹ <i>slot/port</i> } port-channel <i>port_channel_number</i> <i>slot/port</i> }}	Selects the interface to configure.
Step 2	Router(config-if)# mtu <i>mtu_size</i> Router(config-if)# no mtu	Configures the MTU size. Reverts to the default MTU size (1500 bytes).
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show running-config interface [[gigabitethernet tengigabitethernet] <i>slot/port</i>]	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 10-13](#)).

This example shows how to configure the MTU size on Gigabit Ethernet port 1/2:

```
Router# configure terminal
Router(config)# interface gigabitethernet 1/2
Router(config-if)# mtu 9216
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show interface gigabitethernet 1/2
GigabitEthernet1/2 is administratively down, line protocol is down
  Hardware is C6k 1000Mb 802.3, address is 0030.9629.9f88 (bia 0030.9629.9f88)
  MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,
  <...Output Truncated...>
Router#
```

Configuring the Global Egress LAN Port MTU Size

To configure the global egress LAN port MTU size, perform this task:

	Command	Purpose
Step 1	Router(config)# system jumbomtu <i>mtu_size</i>	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 802.3Z Flow Control

Gigabit Ethernet and 10-Gigabit Ethernet ports on the Catalyst 6500 series switches use flow control to stop the transmission of frames to the port for a specified time; other Ethernet ports use flow control to respond to flow-control requests.

If a Gigabit Ethernet or 10-Gigabit Ethernet port receive buffer becomes full, the port transmits an IEEE 802.3Z 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 802.3Z pause frames from other devices.

To configure flow control on an Ethernet port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	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 [<i>type</i> ¹ <i>slot/port</i>] 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 <i>type</i> ¹ <i>slot/port</i>	Selects the port to configure.
Step 2	Router(config-if)# link debounce [time <i>debounce_time</i>]	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 Mbps over copper media.
- Only releases earlier than Release 12.2(17a)SX support WS-X6502-10GE.
- Releases earlier than Release 12.2(18)SXD assume that 10 Gbps ports use fiber media.
- Release 12.2(18)SXD and later releases recognize 10 Gbps copper media.
- Releases earlier than Release 12.2(18)SXD do not detect media-only changes.
- Release 12.2(18)SXD and later releases detect media-only changes.

Table 10-2 lists the time delay that occurs before notification of a link change.

Table 10-2 Port Debounce Timer Delay Time

Port Type	Debounce Timer Disabled	Debounce Timer Enabled
Ports operating at 10 Mbps or 100 Mbps	300 milliseconds	3100 milliseconds
Ports operating at 1000 Mbps or 10 Gbps over copper media	300 milliseconds	3100 milliseconds
Ports operating at 1000 Mbps 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 <i>string</i>	Adds a description for an interface.
Router(config-if) # no description	Deletes a description from an interface.

This example shows how to add a description on Fast Ethernet port 5/5:

```
Router(config)# interface fastethernet 5/5  
Router(config-if)# description Channel-group to "Marketing"
```

Understanding Online Insertion and Removal

The online insertion and removal (OIR) feature supported on the Catalyst 6500 series switches allows you to remove and replace modules while the system is online. You can shut down the modules before removal and restart it after insertion without causing other software or interfaces to shut down.



Note

Do not remove or install more than one module at a time. After you remove or install a module, check the LEDs before continuing. For module LED descriptions, refer to the *Catalyst 6500 Series Switch Installation Guide*.

When a module has been removed or installed, the Catalyst 6500 series switch stops processing traffic for the module and scans the system for a configuration change. Each interface type is verified against the system configuration, and then the system runs diagnostics on the new module. There is no disruption to normal operation during module insertion or removal.

The switch can bring only an identical replacement module online. To support OIR of an identical module, the module configuration is not removed from the running-config file when you remove a module.

If the replacement module is different from the removed module, you must configure it before the switch can bring it online.

Layer 2 MAC addresses are stored in an EEPROM, which allows modules to be replaced online without requiring the system to update switching tables and data structures. Regardless of the types of modules installed, the Layer 2 MAC addresses do not change unless you replace the supervisor engine. If you do replace the supervisor engine, the Layer 2 MAC addresses of *all* ports change to those specified in the address allocator on the new supervisor engine.

Monitoring and Maintaining Interfaces

You can perform the tasks in the following sections to monitor and maintain interfaces:

- [Monitoring Interface Status, page 10-17](#)
- [Clearing Counters on an Interface, page 10-17](#)
- [Resetting an Interface, page 10-18](#)
- [Shutting Down and Restarting an Interface, page 10-18](#)

Monitoring Interface Status

The software contains commands that you can enter at the EXEC prompt to display information about the interface including the version of the software and the hardware and statistics about interfaces. The following table lists some of the interface monitoring commands. (You can display the complete list of **show** commands by using the **show ?** command at the EXEC prompt.) These commands are described in the *Cisco IOS Interface Command Reference* publication.

To display information about the interface, perform these tasks:

Command	Purpose
Router# show ibc	Displays current internal status information.
Router# show eobc	Displays current internal out-of-band information.
Router# show interfaces [<i>type slot/port</i>]	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 [<i>type slot/port</i>]	Displays the global (system-wide) and interface-specific status of any configured protocol.
Router# show version	Displays the hardware configuration, software version, the names and sources of configuration files, and the boot images.

This example shows how to display the status of Fast Ethernet port 5/5:

```
Router# show protocols fastethernet 5/5
FastEthernet5/5 is up, line protocol is up
Router#
```

Clearing Counters on an Interface

To clear the interface counters shown with the **show interfaces** command, perform this task:

Command	Purpose
Router# clear counters {{ <i>vlan vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>channel_ID</i> }}	Clears interface counters.
1. <i>type</i> = ethernet, fastethernet, gigabitethernet, or tengigabitethernet	

This example shows how to clear and reset the counters on Fast Ethernet port 5/5:

```
Router# clear counters fastethernet 5/5
Clear "show interface" counters on this interface [confirm] y
Router#
*Sep 30 08:42:55: %CLEAR-5-COUNTERS: Clear counter on interface FastEthernet5/5
```

The **clear counters** command clears all the current counters from the interface unless the optional arguments specify a specific interface.

**Note**

The **clear counters** command clears counters displayed with the EXEC **show interfaces** command, not counters retrieved using SNMP.

Resetting an Interface

To reset an interface, perform this task:

Command	Purpose
Router# clear interface <i>type</i> ¹ <i>slot/port</i>	Resets an interface.

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

This example shows how to reset Fast Ethernet port 5/5:

```
Router# clear interface fastethernet 5/5
Router#
```

Shutting Down and Restarting an Interface

You can shut down an interface, which disables all functions on the specified interface and shows the interface as unavailable on all monitoring command displays. This information is communicated to other network servers through all dynamic routing protocols. The interface is not included in any routing updates.

To shut down an interface and then restart it, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>channel_ID</i> }}	Selects the interface to be configured.
Step 2	Router(config-if)# shutdown	Shuts down the interface.
Step 3	Router(config-if)# no shutdown	Reenables the interface.

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

This example shows how to shut down Fast Ethernet port 5/5:

```
Router(config)# interface fastethernet 5/5
Router(config-if)# shutdown
Router(config-if)#
*Sep 30 08:33:47: %LINK-5-CHANGED: Interface FastEthernet5/5, changed state to
administratively down
```

This example shows how to reenabale Fast Ethernet port 5/5:

```
Router(config-if)# no shutdown
Router(config-if)#
*Sep 30 08:36:00: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
```

To check if an interface is disabled, enter the EXEC **show interfaces** command. An interface that has been shut down is shown as administratively down in the **show interfaces** command display.

Checking the Cable Status Using the TDR

You can check the status of copper cables using the time domain reflectometer (TDR). The TDR detects a cable fault by sending a signal through the cable and reading the signal that is reflected back to it. All or part of the signal can be reflected back by any number of cable defects or by the end of the cable itself.

Use the TDR to determine if the cabling is at fault if you cannot establish a link. This test is especially important when replacing an existing switch, upgrading to Gigabit Ethernet, or installing new cables.

The port must be up before running the TDR test. If the port is down, you cannot enter the **test cable-diagnostics tdr** command successfully, and the following message is displayed:

```
Router# test cable-diagnostics tdr interface gigabitethernet2/12
% Interface Gi2/12 is administratively down
% Use 'no shutdown' to enable interface before TDR test start.
```

**Note**

- TDR can test cables up to a maximum length of 115 meters.
- See the [Release Notes for Cisco IOS Release 12.2SX 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
<code>test cable-diagnostics tdr interface {interface interface-number}</code>	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 #
```




Configuring LAN Ports for Layer 2 Switching

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



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- To configure Layer 3 interfaces, see [Chapter 23, “Configuring Layer 3 Interfaces.”](#)

This chapter consists of these sections:

- [Understanding How Layer 2 Switching Works, page 11-1](#)
- [Default Layer 2 LAN Interface Configuration, page 11-5](#)
- [Layer 2 LAN Interface Configuration Guidelines and Restrictions, page 11-5](#)
- [Configuring LAN Interfaces for Layer 2 Switching, page 11-6](#)

Understanding How Layer 2 Switching Works

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

- [Understanding Layer 2 Ethernet Switching, page 11-1](#)
- [Understanding VLAN Trunks, page 11-2](#)
- [Layer 2 LAN Port Modes, page 11-4](#)

Understanding Layer 2 Ethernet Switching

These sections describe Layer 2 Ethernet switching:

- [Layer 2 Ethernet Switching Overview, page 11-2](#)
- [Switching Frames Between Segments, page 11-2](#)
- [Building the Address Table, page 11-2](#)

Layer 2 Ethernet Switching Overview

Catalyst 6500 series switches support simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

Catalyst 6500 series switches solve congestion problems caused by high-bandwidth devices and by a large number of users by assigning each device (for example, a server) to its own 10-, 100-, or 1000-Mbps collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a properly configured switched environment achieve full access to the bandwidth.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. Normally, Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles.

Switching Frames Between Segments

Each LAN port on a Catalyst 6500 series switch can connect to a single workstation or server, or to a hub through which workstations or servers connect to the network.

On a typical Ethernet hub, all ports connect to a common backplane within the hub, and the bandwidth of the network is shared by all devices attached to the hub. If two stations establish a session that uses a significant level of bandwidth, the network performance of all other stations attached to the hub is degraded.

To reduce degradation, the switch considers each LAN port to be an individual segment. When stations connected to different LAN ports need to communicate, the switch forwards frames from one LAN port to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between LAN ports efficiently, the switch maintains an address table. When a frame enters the switch, it associates the MAC address of the sending network device with the LAN port on which it was received.

Building the Address Table

Catalyst 6500 series switches build the address table by using the source address of the frames received. When the switch receives a frame for a destination address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the switch adds its relevant source address and port ID to the address table. The switch then forwards subsequent frames to a single LAN port without flooding to all LAN ports.

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

Understanding VLAN Trunks

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

- [Trunking Overview, page 11-3](#)
- [Encapsulation Types, page 11-3](#)

Trunking Overview



Note

For information about VLANs, see [Chapter 15, “Configuring VLANs.”](#)

A trunk is a point-to-point link between the switch and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

Two trunking encapsulations are available on all Ethernet ports:

- Inter-Switch Link (ISL)—ISL is a Cisco-proprietary trunking encapsulation.



Note

The following switching modules do not support ISL encapsulation:

- WS-X6502-10GE
- WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6548-GE-45AF
- WS-X6148-GE-TX, WS-X6148V-GE-TX, WS-X6148-GE-45AF

- 802.1Q—802.1Q is an industry-standard trunking encapsulation.

You can configure a trunk on a single Ethernet port or on an EtherChannel. For more information about EtherChannel, see [Chapter 13, “Configuring EtherChannels.”](#)

Ethernet trunk ports support several trunking modes (see [Table 11-2 on page 11-4](#)). 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 14, “Configuring VTP.”](#)

Encapsulation Types

[Table 11-1](#) lists the Ethernet trunk encapsulation types.

Table 11-1 Ethernet Trunk Encapsulation Types

Encapsulation	Function
<code>switchport trunk encapsulation isl</code>	Specifies ISL encapsulation on the trunk link. Note Some modules do not support ISL encapsulation (see the “Trunking Overview” section on page 11-3).

Table 11-1 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 11-2 lists the Layer 2 LAN port modes and describes how they function on LAN ports.

Table 11-2 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 11-3 shows the Layer 2 LAN port default configuration.

Table 11-3 Layer 2 LAN Interface Default Configuration

Feature	Default
Interface mode:	
<ul style="list-style-type: none"> Before entering the switchport command After entering the switchport command 	Layer 3 (unconfigured) switchport mode dynamic desirable
Trunk encapsulation	switchport trunk encapsulation negotiate
Allowed VLAN range	VLANs 1 to 4094, except reserved VLANs (see Table 15-1 on page 15-2)
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	<ul style="list-style-type: none"> 100 for 10-Mbps Ethernet LAN ports 19 for 10/100-Mbps Fast Ethernet LAN ports 19 for 100-Mbps Fast Ethernet LAN ports 4 for 1,000-Mbps Gigabit Ethernet LAN ports 2 for 10,000-Mbps 10-Gigabit Ethernet LAN ports

Layer 2 LAN Interface Configuration Guidelines and Restrictions

When configuring Layer 2 LAN ports, follow these guidelines and restrictions:

- The following switching modules do not support ISL encapsulation:
 - WS-X6502-10GE
 - WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6548-GE-45AF
 - WS-X6148-GE-TX, WS-X6148V-GE-TX, WS-X6148-GE-45AF

- The following configuration guidelines and restrictions apply when using 802.1Q trunks and impose some limitations on the trunking strategy for a network. Note these restrictions when using 802.1Q trunks:
 - When connecting Cisco switches through an 802.1q trunk, make sure the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.
 - Disabling spanning tree on the native VLAN of an 802.1Q trunk without disabling spanning tree on every VLAN in the network can cause spanning tree loops. We recommend that you leave spanning tree enabled on the native VLAN of an 802.1Q trunk. If this is not possible, disable spanning tree on every VLAN in the network. Make sure your network is free of physical loops before disabling spanning tree.
 - When you connect two Cisco switches through 802.1Q trunks, the switches exchange spanning tree BPDUs on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved IEEE 802.1d spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all other VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).
 - Non-Cisco 802.1Q switches maintain only a single instance of spanning tree (the Mono Spanning Tree, or MST) that defines the spanning tree topology for all VLANs. When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the MST of the non-Cisco switch and the native VLAN spanning tree of the Cisco switch combine to form a single spanning tree topology known as the Common Spanning Tree (CST).
 - Because Cisco switches transmit BPDUs to the SSTP multicast MAC address on VLANs other than the native VLAN of the trunk, non-Cisco switches do not recognize these frames as BPDUs and flood them on all ports in the corresponding VLAN. Other Cisco switches connected to the non-Cisco 802.1q cloud receive these flooded BPDUs. This allows Cisco switches to maintain a per-VLAN spanning tree topology across a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud separating the Cisco switches is treated as a single broadcast segment between all switches connected to the non-Cisco 802.1q cloud through 802.1q trunks.
 - Make certain that the native VLAN is the same on all of the 802.1q trunks connecting the Cisco switches to the non-Cisco 802.1q cloud.
 - If you are connecting multiple Cisco switches to a non-Cisco 802.1q cloud, all of the connections must be through 802.1q trunks. You cannot connect Cisco switches to a non-Cisco 802.1q cloud through ISL trunks or through access ports. Doing so causes the switch to place the ISL trunk port or access port into the spanning tree “port inconsistent” state and no traffic will pass through the port.

Configuring LAN Interfaces for Layer 2 Switching

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

- [Configuring a LAN Port for Layer 2 Switching, page 11-7](#)
- [Configuring a Layer 2 Switching Port as a Trunk, page 11-7](#)
- [Configuring a LAN Interface as a Layer 2 Access Port, page 11-14](#)
- [Configuring a Custom IEEE 802.1Q EtherType Field Value, page 11-15](#)

**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 <i>type</i> ¹ <i>slot/port</i>	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.
	Router(config-if)# no switchport	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. 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 [<i>type</i> ¹ <i>slot/port</i>]	Displays the running configuration of the interface.
Step 7	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] switchport	Displays the switch port configuration of the interface.
Step 8	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] 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 11-8](#)
- [Configuring the Layer 2 Trunk to Use DTP, page 11-9](#)
- [Configuring the Layer 2 Trunk Not to Use DTP, page 11-9](#)
- [Configuring the Access VLAN, page 11-10](#)
- [Configuring the 802.1Q Native VLAN, page 11-10](#)

- [Configuring the List of VLANs Allowed on a Trunk, page 11-11](#)
- [Configuring the List of Prune-Eligible VLANs, page 11-11](#)
- [Completing Trunk Configuration, page 11-12](#)
- [Verifying Layer 2 Trunk Configuration, page 11-12](#)
- [Configuration and Verification Examples, page 11-13](#)

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 11-7](#) 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 11-9](#)) 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 11-12](#) 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 11-7 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 11-2 on page 11-4](#) for information about trunking modes.



Note

Complete the steps in the [“Completing Trunk Configuration”](#) section on page 11-12 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 11-7 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 11-8).
- To support the **switchport nonegotiate** command, you must enter the **switchport mode trunk** command.
- Enter the **switchport mode dynamic trunk** command. See [Table 11-2 on page 11-4](#) 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 11-8) 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 11-9).

**Note**

Complete the steps in the “[Completing Trunk Configuration](#)” section on page 11-12 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 11-7 before performing the tasks in this section.

To configure the access VLAN, perform this task:

Command	Purpose
Router(config-if) # switchport access vlan <i>vlan_ID</i>	(Optional) Configures the access VLAN, which is used if the interface stops trunking. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
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 11-12 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 11-7 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 <i>vlan_ID</i>	(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 15-1 on page 15-2](#)).
- The access VLAN is not automatically used as the native VLAN.

**Note**

Complete the steps in the [“Completing Trunk Configuration”](#) section on page 11-12 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 11-7 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 } <i>vlan</i> [, <i>vlan</i> [, <i>vlan</i> [, ...]]	(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 11-12 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 11-7 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 } <i>vlan</i> [, <i>vlan</i> [, <i>vlan</i> [, ...]]}}	(Optional) Configures the list of prune-eligible VLANs on the trunk (see the “Understanding VTP Pruning” section on page 14-3).
Router(config-if)# no switchport trunk pruning vlan	Reverts to the default value (all VLANs prune-eligible).

When configuring the list of prune-eligible VLANs on a trunk, note the following information:

- The *vlan* parameter is either a single VLAN number from 1 through 4094, except reserved VLANs (see [Table 15-1 on page 15-2](#)), or a range of VLANs described by two VLAN numbers, the lesser one first, separated by a dash. Do not enter any spaces between comma-separated *vlan* parameters or in dash-specified ranges.
- The default list of VLANs allowed to be pruned contains all VLANs.
- Network devices in VTP transparent mode do not send VTP Join messages. On Catalyst 6500 series switches with trunk connections to network devices in VTP transparent mode, configure the VLANs used by the transparent-mode network devices or that need to be carried across the transparent-mode network devices as pruning ineligible.



Note

Complete the steps in the [“Completing Trunk Configuration” section on page 11-12](#) 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 <i>type</i> ¹ <i>slot/port</i>	Displays the running configuration of the interface.
Step 2	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] switchport	Displays the switch port configuration of the interface.
Step 3	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] 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 1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-802,850,917,999,1002-1005
```

```
Port      Vlans in spanning tree forwarding state and not pruned
Fa5/8 1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-802,850,917,999,1002-1005
```

```
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 15-10).

To configure a LAN port as a Layer 2 access port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# shutdown	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching. Note You must enter the switchport command once without any keywords to configure the LAN port as a Layer 2 port before you can enter additional switchport commands with keywords.
Step 4	Router(config-if)# no switchport	Clears Layer 2 LAN port configuration.
Step 5	Router(config-if)# switchport mode access Router(config-if)# no switchport mode	Configures the LAN port as a Layer 2 access port. Reverts to the default switchport mode (switchport mode dynamic desirable).
Step 6	Router(config-if)# switchport access vlan <i>vlan_ID</i> Router(config-if)# no switchport access vlan	Places the LAN port in a VLAN. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2). Reverts to the default access VLAN (VLAN 1).
Step 7	Router(config-if)# no shutdown	Activates the interface. (Required only if you shut down the interface.)
Step 8	Router(config-if)# end	Exits configuration mode.
Step 9	Router# show running-config interface [<i>type</i> ¹ <i>slot/port</i>]	Displays the running configuration of the interface.
Step 10	Router# show interfaces [<i>type</i> ¹ <i>slot/port</i>] 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

With Release 12.2(17a)SX and later releases, 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 <i>value</i>	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.
- Each port supports only one EtherType field value. A port that is configured with a custom EtherType field value does not recognize frames that have any other EtherType field value as tagged frames. For example, a trunk port that is configured with a custom EtherType field value does not recognize the standard 0x8100 EtherType field value on 802.1Q-tagged frames and cannot put the frames into the VLAN to which they belong.

**Caution**

A port that is configured with a custom EtherType field value considers frames that have any other EtherType field value to be untagged frames. A trunk port with a custom EtherType field value places frames with any other EtherType field value into the native VLAN. An access port or tunnel port with a custom EtherType field value places frames that are tagged with any other EtherType field value into the access VLAN. If you misconfigure a custom EtherType field value, frames might be placed into the wrong VLAN.

- See the [Release Notes for Cisco IOS Release 12.2SX 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.
- You cannot configure a custom EtherType field value on the ports in an EtherChannel.
- You cannot form an EtherChannel from ports that are configured with custom EtherType field values.

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

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



Configuring Flex Links

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



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX, at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

The chapter consists of these sections:

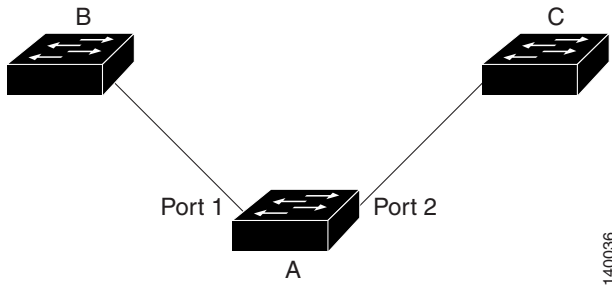
- [Understanding Flex Links, page 12-1](#)
- [Configuring Flex Links, page 12-2](#)
- [Monitoring Flex Links, page 12-3](#)

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 12-1](#), ports 1 and 2 on switch A are connected to uplink switches B and C. Because they are configured as Flex Links, only one of the interfaces is forwarding traffic and the other one is in standby mode. If port 1 is the active link, it begins forwarding traffic between port 1 and switch B; the link between port 2 (the backup link) and switch C is not forwarding traffic. If port 1 goes down, port 2 comes up and starts forwarding traffic to switch C. When port 1 comes back up, it goes into standby mode and does not forward traffic; port 2 continues to forward traffic.

Figure 12-1 Flex Links Configuration Example

If a primary (forwarding) link goes down, a trap notifies the network management stations. If the standby link goes down, a trap notifies the users.

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

Configuring Flex Links

These sections contain this configuration information:

- [Flex Links Default Configuration, page 12-2](#)
- [Flex Links Configuration Guidelines and Restrictions, page 12-2](#)
- [Configuring Flex Links, page 12-3](#)

Flex Links Default Configuration

There is no default Flex Links configuration.

Flex Links Configuration Guidelines and Restrictions

When configuring Flex Links, follow these guidelines and restrictions:

- You can configure only one Flex Links backup link for any active link, and it must be a different interface from the active interface.
- An interface can belong to only one Flex Links pair. An interface can be a backup link for only one active link. An active link cannot belong to another Flex Links pair.
- Neither of the links can be a port that belongs to an EtherChannel. However, you can configure two port channels (EtherChannel logical interfaces) as Flex Links, and you can configure a port channel and a physical interface as Flex Links, with either the port channel or the physical interface as the active link.
- A backup link does not have to be the same type as the active link (Fast Ethernet, Gigabit Ethernet, or port channel). However, you should configure both Flex Links with similar characteristics so that there are no loops or changes in operation if the standby link becomes active.
- STP is disabled on Flex Links ports. If STP is disabled on the switch, be sure that there are no Layer 2 loops in the network topology.
- Do not configure 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 switch startup configuration file.

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

This example shows how to configure an interface with a backup interface and how to verify the configuration:

```
Router# configure terminal
Router(conf)# interface fastethernet1/1
Router(conf-if)# switchport backup interface fastethernet1/2
Router(conf-if)# exit
Router# show interface switchport backup
Router Backup Interface Pairs:
```

Active Interface	Backup Interface	State
FastEthernet1/1	FastEthernet1/2	Active Up/Backup Standby
FastEthernet1/3	FastEthernet2/4	Active Up/Backup Standby
Port-channel1	GigabitEthernet7/1	Active Up/Backup Standby

Monitoring Flex Links

Table 12-1 shows the privileged EXEC command for monitoring the Flex Links configuration.

Table 12-1 Flex Links Monitoring Command

Command	Purpose
show interface [{type ¹ slot/port} {port-channel number}] switchport backup	Displays the Flex Links backup interface configured for an interface, or displays all Flex Links configured on the switch and the state of each active and backup interface (up or standby mode).

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



Configuring EtherChannels

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



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How EtherChannels Work, page 13-1](#)
- [EtherChannel Feature Configuration Guidelines and Restrictions, page 13-5](#)
- [Configuring EtherChannels, page 13-6](#)

Understanding How EtherChannels Work

These sections describe how EtherChannels work:

- [EtherChannel Feature Overview, page 13-1](#)
- [Understanding How EtherChannels Are Configured, page 13-2](#)
- [Understanding Port Channel Interfaces, page 13-4](#)
- [Understanding Load Balancing, page 13-5](#)

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.

With Release 12.2(18)SX and later releases, a Catalyst 6500 series switch supports a maximum of 128 EtherChannels. With releases earlier than Release 12.2(18)SX, a Catalyst 6500 series switch supports a maximum of 64 EtherChannels.

You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6500 series switch. All LAN ports in each EtherChannel must be the same speed and must all be configured as either Layer 2 or Layer 3 LAN ports.

**Note**

The network device to which a Catalyst 6500 series switch is connected may impose its own limits on the number of ports in an EtherChannel.

If a segment within an EtherChannel fails, traffic previously carried over the failed link switches to the remaining segments within the EtherChannel. When a failure occurs, the EtherChannel feature sends a trap that identifies the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one segment in an EtherChannel are blocked from returning on any other segment of the EtherChannel.

Understanding How EtherChannels Are Configured

These sections describe how EtherChannels are configured:

- [EtherChannel Configuration Overview, page 13-2](#)
- [Understanding Manual EtherChannel Configuration, page 13-3](#)
- [Understanding PAGP EtherChannel Configuration, page 13-3](#)
- [Understanding IEEE 802.3ad LACP EtherChannel Configuration, page 13-3](#)

EtherChannel Configuration Overview

You can configure EtherChannels manually or you can use the Port Aggregation Control Protocol (PAgP) or the Link Aggregation Control Protocol (LACP) to form EtherChannels. The EtherChannel protocols allow ports with similar characteristics to form an EtherChannel through dynamic negotiation with connected network devices. PAgP is a Cisco-proprietary protocol and LACP is defined in IEEE 802.3ad.

PAgP and LACP do not interoperate with each other. Ports configured to use PAgP cannot form EtherChannels with ports configured to use LACP. Ports configured to use LACP cannot form EtherChannels with ports configured to use PAgP.

[Table 13-1](#) lists the user-configurable EtherChannel modes.

Table 13-1 EtherChannel Modes

Mode	Description
on	Mode that forces the LAN port to channel unconditionally. In the on mode, a usable EtherChannel exists only when a LAN port group in the on mode is connected to another LAN port group in the on mode. Because ports configured in the on mode do not negotiate, there is no negotiation traffic between the ports. You cannot configure the on mode with an EtherChannel protocol.
auto	PAgP mode that places a LAN port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not initiate PAgP negotiation. (Default)
desirable	PAgP mode that places a LAN port into an active negotiating state, in which the port initiates negotiations with other LAN ports by sending PAgP packets.
passive	LACP mode that places a port into a passive negotiating state, in which the port responds to LACP packets it receives but does not initiate LACP negotiation. (Default)
active	LACP mode that places a port into an active negotiating state, in which the port initiates negotiations with other ports by sending LACP packets.

Understanding Manual EtherChannel Configuration

Manually configured EtherChannel ports do not exchange EtherChannel protocol packets. A manually configured EtherChannel forms only when you enter configure all ports in the EtherChannel compatibly.

Understanding PAgP EtherChannel Configuration

PAgP supports the automatic creation of EtherChannels by exchanging PAgP packets between LAN ports. PAgP packets are exchanged only between ports in **auto** and **desirable** modes.

The protocol learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once PAgP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

Both the **auto** and **desirable** modes allow PAgP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. Layer 2 EtherChannels also use VLAN numbers.

LAN ports can form an EtherChannel when they are in different PAgP modes if the modes are compatible. For example:

- A LAN port in **desirable** mode can form an EtherChannel successfully with another LAN port that is in **desirable** mode.
- A LAN port in **desirable** mode can form an EtherChannel with another LAN port in **auto** mode.
- A LAN port in **auto** mode cannot form an EtherChannel with another LAN port that is also in **auto** mode, because neither port will initiate negotiation.

Understanding IEEE 802.3ad LACP EtherChannel Configuration

LACP supports the automatic creation of EtherChannels by exchanging LACP packets between LAN ports. LACP packets are exchanged only between ports in **passive** and **active** modes.

The protocol learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

Both the **passive** and **active** modes allow LACP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. Layer 2 EtherChannels also use VLAN numbers.

LAN ports can form an EtherChannel when they are in different LACP modes as long as the modes are compatible. For example:

- A LAN port in **active** mode can form an EtherChannel successfully with another LAN port that is in **active** mode.
- A LAN port in **active** mode can form an EtherChannel with another LAN port in **passive** mode.
- A LAN port in **passive** mode cannot form an EtherChannel with another LAN port that is also in **passive** mode, because neither port will initiate negotiation.

LACP uses the following parameters:

- LACP system priority—You must configure an LACP system priority on each switch running LACP. The system priority can be configured automatically or through the CLI (see the [“Configuring the LACP System Priority and System ID” section on page 13-10](#)). LACP uses the system priority with the switch MAC address to form the system ID and also during negotiation with other systems.



Note The LACP system ID is the combination of the LACP system priority value and the MAC address of the switch.

- LACP port priority—You must configure an LACP port priority on each port configured to use LACP. The port priority can be configured automatically or through the CLI (see the [“Configuring Channel Groups” section on page 13-8](#)). 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.
- LACP administrative key—LACP automatically configures an administrative key value equal to the channel group identification number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port’s ability to aggregate with other ports is determined by these factors:
 - Port physical characteristics, such as data rate, duplex capability, and point-to-point or shared medium
 - Configuration restrictions that you establish

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

Understanding Port Channel Interfaces

Each EtherChannel has a numbered port channel interface. With Release 12.2(18)SXE and later releases, you can configure a maximum of 128 port-channel interfaces, numbered from 1 to 256. With releases earlier than Release 12.2(18)SXE, you can configure a maximum of 64 port-channel interfaces, numbered from 1 to 256.

The configuration that you apply to the port channel interface affects all LAN ports assigned to the port channel interface.

After you configure an EtherChannel, the configuration that you apply to the port channel interface affects the EtherChannel; the configuration that you apply to the LAN ports affects only the LAN port where you apply the configuration. To change the parameters of all ports in an EtherChannel, apply the configuration commands to the port channel interface, for example, Spanning Tree Protocol (STP) commands or commands to configure a Layer 2 EtherChannel as a trunk.

Understanding Load Balancing

An EtherChannel balances the traffic load across the links in an EtherChannel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel.

EtherChannel load balancing can use MAC addresses or IP addresses. EtherChannel load balancing can also use Layer 4 port numbers. EtherChannel load balancing can use either source or destination or both source and destination addresses or ports. The selected mode applies to all EtherChannels configured on the switch. EtherChannel load balancing can use MPLS Layer 2 information.

Use the option that provides the balance criteria with the greatest variety in your configuration. For example, if the traffic on an EtherChannel is going only to a single MAC address and you use the destination MAC address as the basis of EtherChannel load balancing, the EtherChannel always chooses the same link in the EtherChannel; using source addresses or IP addresses might result in better load balancing.

EtherChannel Feature Configuration Guidelines and Restrictions

When EtherChannel interfaces are configured improperly, they are disabled automatically to avoid network loops and other problems. To avoid configuration problems, observe these guidelines and restrictions:

- The commands in this chapter can be used on all LAN ports in Catalyst 6500 series switches, including the ports on the supervisor engine and a redundant supervisor engine.
- Release 12.2(17b)SXA and later releases provide support for more than 1 Gbps of traffic per EtherChannel on the WS-X6548-GE-TX and WS-X6548V-GE-TX switching modules.
- With Release 12.2(17a)SX and Release 12.2(17a)SX1, the WS-X6548-GE-TX and WS-X6548V-GE-TX fabric-enabled switching modules do not 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 ISL trunking to an EtherChannel, Cisco IOS software automatically adds a **switchport trunk encapsulation dot1q** command to the port-channel interface to prevent configuration of the EtherChannel as an ISL trunk. The **switchport trunk encapsulation dot1q** command is inactive when the EtherChannel is not a trunk.
- All Ethernet LAN ports on all modules, including those on a redundant supervisor engine, support EtherChannels (maximum of eight LAN ports) with no requirement that the LAN ports be physically contiguous or on the same module.
- Configure all LAN ports in an EtherChannel to use the same EtherChannel protocol; you cannot run two EtherChannel protocols in one EtherChannel.
- Configure all LAN ports in an EtherChannel to operate at the same speed and in the same duplex mode.
- LACP does not support half-duplex. Half-duplex ports in an LACP EtherChannel are put in the suspended state.
- Enable all LAN ports in an EtherChannel. If you shut down a LAN port in an EtherChannel, it is treated as a link failure and its traffic is transferred to one of the remaining ports in the EtherChannel.

- An EtherChannel will not form if one of the LAN ports is a Switched Port Analyzer (SPAN) destination port.
- For Layer 3 EtherChannels, assign Layer 3 addresses to the port channel logical interface, not to the LAN ports in the channel.
- For Layer 2 EtherChannels:
 - Assign all LAN ports in the EtherChannel to the same VLAN or configure them as trunks.
 - If you configure an EtherChannel from trunking LAN ports, verify that the trunking mode is the same on all the trunks. LAN ports in an EtherChannel with different trunk modes can operate unpredictably.
 - An EtherChannel supports the same allowed range of VLANs on all the LAN ports in a trunking Layer 2 EtherChannel. If the allowed range of VLANs is not the same, the LAN ports do not form an EtherChannel.
 - LAN ports with different STP port path costs can form an EtherChannel as long they are compatibly configured with each other. If you set different STP port path costs, the LAN ports are not incompatible for the formation of an EtherChannel.
 - An EtherChannel will not form if protocol filtering is set differently on the LAN ports.
- After you configure an EtherChannel, the configuration that you apply to the port channel interface affects the EtherChannel. The configuration that you apply to the LAN ports affects only the LAN port where you apply the configuration.
- When QoS is enabled, enter the **no mls qos channel-consistency** port-channel interface command to support EtherChannels that have ports with and without strict-priority queues.

Configuring EtherChannels

These sections describe how to configure EtherChannels:

- [Configuring Port Channel Logical Interfaces for Layer 3 EtherChannels, page 13-7](#)
- [Configuring Channel Groups, page 13-8](#)
- [Configuring EtherChannel Load Balancing, page 13-11](#)
- [Configuring the EtherChannel Min-Links Feature, page 13-12](#)



Note

Make sure that the LAN ports are configured correctly (see the “[EtherChannel Feature Configuration Guidelines and Restrictions](#)” section on page 13-5).

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 13-8](#)).
- 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 13-8](#)).
- 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:

	Command	Purpose
Step 1	Router(config)# interface port-channel <i>number</i>	Creates the port channel interface.
	Router(config)# no interface port-channel <i>number</i>	Deletes the port channel interface.
Step 2	Router(config-if)# ip address <i>ip_address mask</i>	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 <i>group_number</i>	Verifies the configuration.

With Release 12.2(18)SXE and later releases, the *group_number* can be 1 through 256, up to a maximum of 128 port-channel interfaces. With releases earlier than Release 12.2(18)SXE, the *group_number* can be 1 through 256, up to a maximum of 64 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 13-7), 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:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects a LAN port to configure.
Step 2	Router(config-if)# no ip address	Ensures that there is no IP address assigned to the LAN port.
Step 3	Router(config-if)# channel-protocol (lACP pagp)	(Optional) On the selected LAN port, restricts the channel-group command to the EtherChannel protocol configured with the channel-protocol command.
	Router(config-if)# no channel-protocol	Removes the restriction.
Step 4	Router(config-if)# channel-group <i>number</i> mode { active auto desirable on passive }	Configures the LAN port in a port channel and specifies the mode (see Table 13-1 on page 13-2). PAGP supports only the auto and desirable modes. LACP supports only the active and passive modes.
	Router(config-if)# no channel-group	Removes the LAN port from the channel group.
Step 5	Router(config-if)# lacp port-priority <i>priority_value</i>	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.
	Router(config-if)# no lacp port-priority	Reverts to the default.
Step 6	Router(config-if)# end	Exits configuration mode.
Step 7	Router# show running-config interface <i>type</i> ¹ <i>slot/port</i> Router# show interfaces <i>type</i> ¹ <i>slot/port</i> etherchannel	Verifies the configuration.

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

This example shows how to configure Fast Ethernet ports 5/6 and 5/7 into port channel 2 with PAGP mode **desirable**:

```
Router# configure terminal
Router(config)# interface range fastethernet 5/6 -7
Router(config-if)# channel-group 2 mode desirable
Router(config-if)# end
```

**Note**

See the [“Configuring a Range of Interfaces”](#) section on page 10-4 for information about the **range** keyword.

This example shows how to verify the configuration of port channel interface 2:

```
Router# show running-config interface port-channel 2
Building configuration...
```

```
Current configuration:
!
interface Port-channel2
 no ip address
 switchport
 switchport access vlan 10
 switchport mode access
end
Router#
```

This example shows how to verify the configuration of Fast Ethernet port 5/6:

```
Router# show running-config interface fastethernet 5/6
Building configuration...
```

```
Current configuration:
!
interface FastEthernet5/6
 no ip address
 switchport
 switchport access vlan 10
 switchport mode access
 channel-group 2 mode desirable
end
Router# show interfaces fastethernet 5/6 etherchannel
Port state      = Down Not-in-Bndl
Channel group = 12          Mode = Desirable-S1      Gcchange = 0
Port-channel   = null      GC    = 0x00000000        Pseudo port-channel = Po1
2
Port index     = 0          Load = 0x00          Protocol =   PAgP

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        P - Device learns on physical port.
        d - PAgP is down.

Timers: H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.    I - Interface timer is running.

Local information:

          Hello  Partner  PAgP    Learning  Group
Port    Flags State  Timers Interval Count  Priority  Method  Ifindex
Fa5/2   d      U1/S1          1s        0      128     Any    0

Age of the port in the current state: 04d:18h:57m:19s
```

This example shows how to verify the configuration of port channel interface 2 after the LAN ports have been configured:

```
Router# show etherchannel 12 port-channel
Port-channels in the group:
-----

Port-channel: Po12
-----

Age of the Port-channel   = 04d:18h:58m:50s
Logical slot/port        = 14/1           Number of ports = 0
GC                        = 0x00000000    HotStandBy port = null
Port state                = Port-channel  Ag-Not-Inuse
Protocol                  =   PAgP

Router#
```

Configuring the LACP System Priority and System ID

The LACP system ID is the combination of the LACP system priority value and the MAC address of the switch.

To configure the LACP system priority and system ID, perform this task:

	Command	Purpose
Step 1	Router(config)# lacp system-priority <i>priority_value</i>	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.
	Router(config)# no lacp system-priority	Reverts to the default.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show lacp sys-id	Verifies the configuration.

This example shows how to configure the LACP system priority:

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

This example shows how to verify the configuration:

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

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

Configuring EtherChannel Load Balancing

To configure EtherChannel load balancing, perform this task:

	Command	Purpose
Step 1	Router(config)# port-channel load-balance { src-mac dst-mac src-dst-mac src-ip dst-ip src-dst-ip src-port dst-port src-dst-port }	Configures EtherChannel load balancing.
	Router(config)# no port-channel load-balance	Reverts to default EtherChannel load balancing.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show etherchannel load-balance	Verifies the configuration.

The load-balancing keywords indicate the following information:

- **dst-ip**—Destination IP addresses
- **dst-mac**—Destination MAC addresses
- **dst-port**—Destination Layer 4 port
- **mpls**—Load balancing for MPLS packets
- **src-dst-ip**—Source and destination IP addresses
- **src-dst-mac**—Source and destination MAC addresses
- **src-dst-port**—Source and destination Layer 4 port
- **src-ip**—Source IP addresses
- **src-mac**—Source MAC addresses
- **src-port**—Source Layer 4 port

This example shows how to configure EtherChannel to use source and destination IP addresses:

```
Router# configure terminal
Router(config)# port-channel load-balance src-dst-ip
Router(config)# end
Router(config)#
```

This example shows how to verify the configuration:

```
Router# show etherchannel load-balance
Source XOR Destination IP address
Router#
```

Configuring the EtherChannel Min-Links Feature



Note

Release 12.2(18)SXF and later releases support the EtherChannel Min-Links feature.

The EtherChannel Min-Links feature is supported on [LACP](#) EtherChannels. This feature allows you to configure the minimum number of member ports that must be in the link-up state and bundled in the EtherChannel for the port channel interface to transition to the link-up state. You can use the EtherChannel Min-Links feature to prevent low-bandwidth LACP EtherChannels from becoming active. This feature also causes LACP EtherChannels to become inactive if they have too few active member ports to supply your required minimum bandwidth.

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

	Command	Purpose
Step 1	Router(config)# interface port-channel <i>number</i>	Selects an LACP port channel interface.
Step 2	Router(config-if)# port-channel min-links <i>number</i>	Configures the minimum number of member ports that must be in the link-up state and bundled in the EtherChannel for the port channel interface to transition to the link-up state.
	Router(config-if)# no port-channel min-links	Reverts to the default number of active member ports (one).
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show running-config interface <i>type</i> ¹ <i>slot/port</i> Router# show interfaces <i>type</i> ¹ <i>slot/port</i> etherchannel	Verifies the configuration.



Note

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

This example shows how to configure port channel interface 1 to be inactive if fewer than 2 member ports are active in the EtherChannel:

```
Router# configure terminal
Router(config)# interface port-channel 1
Router(config-if)# port-channel min-links 2
Router(config-if)# end
```



Configuring VTP

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



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How VTP Works, page 14-1](#)
- [VTP Default Configuration, page 14-5](#)
- [VTP Configuration Guidelines and Restrictions, page 14-5](#)
- [Configuring VTP, page 14-6](#)

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 15, “Configuring VLANs.”](#)

These sections describe how VTP works:

- [Understanding the VTP Domain, page 14-2](#)
- [Understanding VTP Modes, page 14-2](#)
- [Understanding VTP Advertisements, page 14-3](#)

- [Understanding VTP Version 2, page 14-3](#)
- [Understanding VTP Pruning, page 14-3](#)

Understanding the VTP Domain

A VTP domain (also called a VLAN management domain) is made up of one or more interconnected network devices that share the same VTP domain name. A network device can be configured to be in one and only one VTP domain. You make global VLAN configuration changes for the domain using either the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

By default, the Catalyst 6500 series switch is in VTP server mode and is in the no-management domain state until the switch receives an advertisement for a domain over a trunk link or you configure a management domain.

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

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

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

VTP maps VLANs dynamically across multiple LAN types with unique names and internal index associations. Mapping eliminates excessive device administration required from network administrators.

Understanding VTP Modes

You can configure a Catalyst 6500 series switch to operate in any one of these VTP modes:

- **Server**—In VTP server mode, you can create, modify, and delete VLANs and specify other configuration parameters (such as VTP version and VTP pruning) for the entire VTP domain. VTP servers advertise their VLAN configuration to other network devices in the same VTP domain and synchronize their VLAN configuration with other network devices based on advertisements received over trunk links. VTP server is the default mode.
- **Client**—VTP clients behave the same way as VTP servers, but you cannot create, change, or delete VLANs on a VTP client.
- **Transparent**—VTP transparent network devices do not participate in VTP. A VTP transparent network device does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements. However, in VTP version 2, transparent network devices do forward VTP advertisements that they receive out their trunking LAN ports.



Note

Catalyst 6500 series switches automatically change from VTP server mode to VTP client mode if the switch detects a failure while writing configuration to NVRAM. If this happens, the switch cannot be returned to VTP server mode until the NVRAM is functioning.

Understanding VTP Advertisements

Each network device in the VTP domain sends periodic advertisements out each trunking LAN port to a reserved multicast address. VTP advertisements are received by neighboring network devices, which update their VTP and VLAN configurations as necessary.

The following global configuration information is distributed in VTP advertisements:

- VLAN IDs (ISL and 802.1Q)
- Emulated LAN names (for ATM LANE)
- 802.10 SAID values (FDDI)
- VTP domain name
- VTP configuration revision number
- VLAN configuration, including maximum transmission unit (MTU) size for each VLAN
- Frame format

Understanding VTP Version 2

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

**Note**

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

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

- Token Ring support—VTP version 2 supports Token Ring LAN switching and VLANs (Token Ring Bridge Relay Function [TrBRF] and Token Ring Concentrator Relay Function [TrCRF]). For more information about Token Ring VLANs, see the [“Understanding How VLANs Work” section on page 15-1](#).
- Unrecognized Type-Length-Value (TLV) Support—A VTP server or client propagates configuration changes to its other trunks, even for TLVs it is not able to parse. The unrecognized TLV is saved in NVRAM.
- Version-Dependent Transparent Mode—In VTP version 1, a VTP transparent network device inspects VTP messages for the domain name and version, and forwards a message only if the version and domain name match. Because only one domain is supported in the supervisor engine software, VTP version 2 forwards VTP messages in transparent mode without checking the version.
- Consistency Checks—In VTP version 2, VLAN consistency checks (such as VLAN names and values) are performed only when you enter new information through the CLI or SNMP. Consistency checks are not performed when new information is obtained from a VTP message, or when information is read from NVRAM. If the digest on a received VTP message is correct, its information is accepted without consistency checks.

Understanding VTP Pruning

VTP pruning enhances network bandwidth use by reducing unnecessary flooded traffic, such as broadcast, multicast, unknown, and flooded unicast packets. VTP pruning increases available bandwidth by restricting flooded traffic to those trunk links that the traffic must use to access the appropriate network devices. By default, VTP pruning is disabled.

For VTP pruning to be effective, all devices in the management domain must support VTP pruning. On devices that do not support VTP pruning, you must manually configure the VLANs allowed on trunks.

Figure 14-1 shows a switched network without VTP pruning enabled. Interface 1 on network Switch 1 and port 2 on Switch 4 are assigned to the Red VLAN. A broadcast is sent from the host connected to Switch 1. Switch 1 floods the broadcast, and every network device in the network receives it, even though Switches 3, 5, and 6 have no ports in the Red VLAN.

You enable pruning globally on the Catalyst 6500 series switch (see the “Enabling VTP Pruning” section on page 14-7). You configure pruning on Layer 2 trunking LAN ports (see the “Configuring a Layer 2 Switching Port as a Trunk” section on page 11-7).

Figure 14-1 Flooding Traffic without VTP Pruning

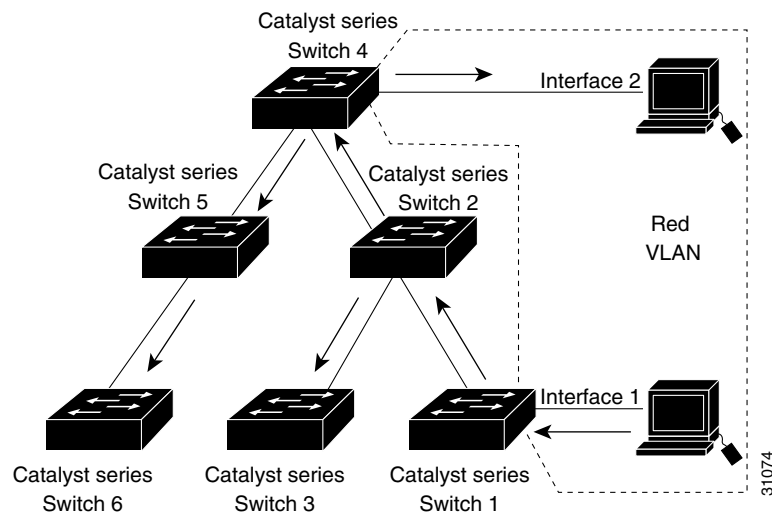
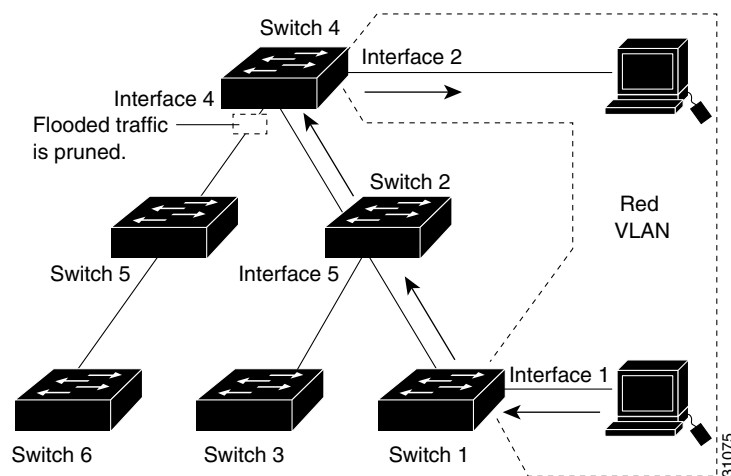


Figure 14-2 shows the same switched network with VTP pruning enabled. The broadcast traffic from Switch 1 is not forwarded to Switches 3, 5, and 6 because traffic for the Red VLAN has been pruned on the links indicated (port 5 on Switch 2 and port 4 on Switch 4).

Figure 14-2 Flooding Traffic with VTP Pruning



Enabling VTP pruning on a VTP server enables pruning for the entire management domain. VTP pruning takes effect several seconds after you enable it. By default, VLANs 2 through 1000 are pruning eligible. VTP pruning does not prune traffic from pruning-ineligible VLANs. VLAN 1 is always pruning ineligible; traffic from VLAN 1 cannot be pruned.

To configure VTP pruning on a trunking LAN port, use the **switchport trunk pruning vlan** command (see the “Configuring a Layer 2 Switching Port as a Trunk” section on page 11-7). VTP pruning operates when a LAN port is trunking. You can set VLAN pruning eligibility when VTP pruning is enabled or disabled for the VTP domain, when any given VLAN exists or not, and when the LAN port is currently trunking or not.

VTP Default Configuration

Table 14-1 shows the default VTP configuration.

Table 14-1 VTP Default Configuration

Feature	Default Value
VTP domain name	Null
VTP mode	Server
VTP version 2 enable state	Version 2 is disabled
VTP password	None
VTP pruning	Disabled

VTP Configuration Guidelines and Restrictions

When implementing VTP in your network, follow these guidelines and restrictions:

- Supervisor engine redundancy does not support nondefault VLAN data file names or locations. Do not enter the **vtp file file_name** command on a switch that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- All network devices in a VTP domain must run the same VTP version.
- You must configure a password on each network device in the management domain when in secure mode.



Caution

If you configure VTP in secure mode, the management domain will not function properly if you do not assign a management domain password to each network device in the domain.

- A VTP version 2-capable network device can operate in the same VTP domain as a network device running VTP version 1 provided VTP version 2 is disabled on the VTP version 2-capable network device (VTP version 2 is disabled by default).
- Do not enable VTP version 2 on a network device unless all of the network devices in the same VTP domain are version 2-capable. When you enable VTP version 2 on a network device, all of the version 2-capable network devices in the domain enable VTP version 2.
- In a Token Ring environment, you must enable VTP version 2 for Token Ring VLAN switching to function properly.

- When you enable or disable VTP pruning on a VTP server, VTP pruning for the entire management domain is enabled or disabled.
- The pruning-eligibility configuration applies globally to all trunks on the switch. You cannot configure pruning-eligibility separately for each trunk.
- When you configure VLANs as pruning eligible or pruning ineligible, pruning eligibility for those VLANs is affected on that switch only, not on all network devices in the VTP domain.
- If there is insufficient DRAM available for use by VTP, the VTP mode changes to transparent.
- Network devices in VTP transparent mode do not send VTP Join messages. On Catalyst 6500 series switches with trunk connections to network devices in VTP transparent mode, configure the VLANs that are used by the transparent-mode network devices or that need to be carried across trunks as pruning ineligible. For information about configuring prune eligibility, see the [“Configuring the List of Prune-Eligible VLANs” section on page 11-11](#).

Configuring VTP

These sections describe how to configure VTP:

- [Configuring VTP Global Parameters, page 14-6](#)
- [Configuring the VTP Mode, page 14-8](#)
- [Displaying VTP Statistics, page 14-10](#)

Configuring VTP Global Parameters

These sections describe configuring the VTP global parameters:

- [Configuring a VTP Password, page 14-6](#)
- [Enabling VTP Pruning, page 14-7](#)
- [Enabling VTP Version 2, page 14-7](#)

**Note**

You can enter the VTP global parameters in either global configuration mode or in EXEC mode.

Configuring a VTP Password

To configure the VTP global parameters, perform this task:

Command	Purpose
Router(config)# vtp password <i>password_string</i>	Sets a password, which can be from 8 to 64 characters long, for the VTP domain.
Router(config)# no vtp password	Clears the password.

This example shows one way to configure a VTP password in global configuration mode:

```
Router# configure terminal
Router(config)# vtp password WATER
Setting device VLAN database password to WATER.
Router#
```

This example shows how to configure a VTP password in EXEC mode:

```
Router# vtp password WATER
Setting device VLAN database password to WATER.
Router#
```


Note

The password is not stored in the running-config file.

Enabling VTP Pruning

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

	Command	Purpose
Step 1	Router(config)# vtp pruning	Enables VTP pruning in the management domain.
	Router(config)# no vtp pruning	Disables VTP pruning in the management domain.
Step 2	Router# show vtp status	Verifies the configuration.

This example shows one way to enable VTP pruning in the management domain:

```
Router# configure terminal
Router(config)# vtp pruning
Pruning switched ON
```

This example shows how to enable VTP pruning in the management domain with any release:

```
Router# vtp pruning
Pruning switched ON
```

This example shows how to verify the configuration:

```
Router# show vtp status | include Pruning
VTP Pruning Mode: Enabled
Router#
```

For information about configuring prune eligibility, see the [“Configuring the List of Prune-Eligible VLANs” section on page 11-11](#).

Enabling VTP Version 2

VTP version 2 is disabled by default on VTP version 2-capable network devices. When you enable VTP version 2 on a network device, every VTP version 2-capable network device in the VTP domain enables version 2.


Caution

VTP version 1 and VTP version 2 are not interoperable on network devices in the same VTP domain. Every network device in the VTP domain must use the same VTP version. Do not enable VTP version 2 unless every network device in the VTP domain supports version 2.


Note

In a Token Ring environment, you must enable VTP version 2 for Token Ring VLAN switching to function properly on devices that support Token Ring interfaces.

To enable VTP version 2, perform this task:

	Command	Purpose
Step 1	Router(config)# vtp version {1 2}	Enables VTP version 2.
	Router(config)# no vtp version	Reverts to the default (VTP version 1).
Step 2	Router# show vtp status	Verifies the configuration.

This example shows one way to enable VTP version 2:

```
Router# configure terminal
Router(config)# vtp version 2
V2 mode enabled.
Router(config)#
```

This example shows how to enable VTP version 2 with any release:

```
Router# vtp version 2
V2 mode enabled.
Router#
```

This example shows how to verify the configuration:

```
Router# show vtp status | include V2
VTP V2 Mode: Enabled
Router#
```

Configuring the VTP Mode

To configure the VTP mode, perform this task:

	Command	Purpose
Step 1	Router(config)# vtp mode {client server transparent}	Configures the VTP mode.
	Router(config)# no vtp mode	Reverts to the default VTP mode (server).
Step 2	Router(config)# vtp domain domain_name	(Optional for server mode) Defines the VTP domain name, which can be up to 32 characters long. VTP server mode requires a domain name. If the switch has a trunk connection to a VTP domain, the switch learns the domain name from the VTP server in the domain. Note You cannot clear the domain name.
Step 3	Router(config)# end	Exits VLAN configuration mode.
Step 4	Router# show vtp status	Verifies the configuration.



Note

When VTP is disabled, you can enter VLAN configuration commands in configuration mode instead of the VLAN database mode and the VLAN configuration is stored in the startup configuration file.

This example shows how to configure the switch as a VTP server:

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

This example shows how to verify the configuration:

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

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

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

This example shows how to verify the configuration:

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

This example shows how to disable VTP on the switch:

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

This example shows how to verify the configuration:

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

Displaying VTP Statistics

To display VTP statistics, including VTP advertisements sent and received and VTP errors, perform this task:

Command	Purpose
Router# show vtp counters	Displays VTP statistics.

This example shows how to display VTP statistics:

```
Router# show vtp counters
VTP statistics:
Summary advertisements received : 7
Subset advertisements received  : 5
Request advertisements received  : 0
Summary advertisements transmitted : 997
Subset advertisements transmitted : 13
Request advertisements transmitted : 3
Number of config revision errors : 0
Number of config digest errors   : 0
Number of V1 summary errors      : 0

VTP pruning statistics:

Trunk          Join Transmitted Join Received  Summary advts received from
-----          -----
Fa5/8          43071             42766             5
non-pruning-capable device
```




Configuring VLANs

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



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How VLANs Work, page 15-1](#)
- [VLAN Default Configuration, page 15-6](#)
- [VLAN Configuration Guidelines and Restrictions, page 15-8](#)
- [Configuring VLANs, page 15-9](#)

Understanding How VLANs Work

The following sections describe how VLANs work:

- [VLAN Overview, page 15-1](#)
- [VLAN Ranges, page 15-2](#)
- [Configurable VLAN Parameters, page 15-3](#)
- [Understanding Token Ring VLANs, page 15-3](#)

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 a 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 21-2).

Catalyst 6500 series switches support 4096 VLANs in accordance with the IEEE 802.1Q standard. These VLANs are organized into several ranges; you use each range slightly differently. Some of these VLANs are propagated to other switches in the network when you use the VLAN Trunking Protocol (VTP). The extended-range VLANs are not propagated, so you must configure extended-range VLANs manually on each network device.

Table 15-1 describes the VLAN ranges.

Table 15-1 VLAN Ranges

VLANs	Range	Usage	Propagated by VTP
0, 4095	Reserved	For system use only. You cannot see or use these VLANs.	—
1	Normal	Cisco default. You can use this VLAN but you cannot delete it.	Yes
2–1001	Normal	For Ethernet VLANs; you can create, use, and delete these VLANs.	Yes
1002–1005	Normal	Cisco defaults for FDDI and Token Ring. You cannot delete VLANs 1002–1005.	Yes
1006–4094	Extended	For Ethernet VLANs only.	No

The following information applies to VLAN ranges:

- Layer 3 LAN ports, WAN interfaces and subinterfaces, and some software features use internal VLANs in the extended range. You cannot use an extended range VLAN that has been allocated for internal use.
- To display the VLANs used internally, enter the **show vlan internal usage** command. With earlier releases, enter the **show vlan internal usage** and **show cwan vlans** commands.
- You can configure ascending internal VLAN allocation (from 1006 and up) or descending internal VLAN allocation (from 4094 and down).
- Switches running the Catalyst operating system do not support configuration of VLANs 1006–1024. If you configure VLANs 1006–1024, ensure that the VLANs do not extend to any switches running Catalyst software.
- You must enable the extended system ID to use extended range VLANs (see the [“Understanding the Bridge ID”](#) section on page 21-2).

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:

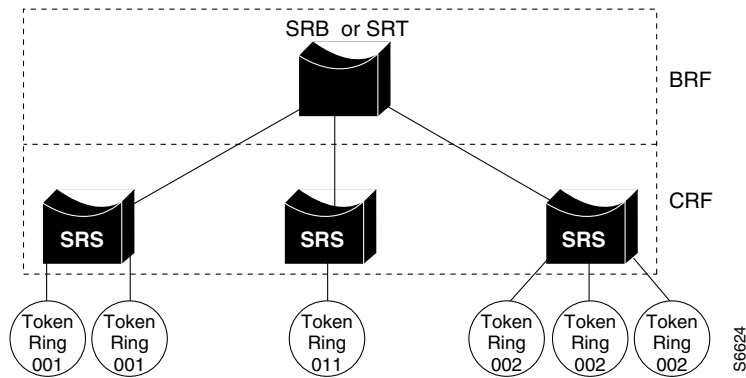
- [Token Ring TrBRF VLANs, page 15-3](#)
- [Token Ring TrCRF VLANs, page 15-4](#)

**Note**

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

Token Ring TrBRF VLANs

Token Ring Bridge Relay Function (TrBRF) VLANs interconnect multiple Token Ring Concentrator Relay Function (TrCRF) VLANs in a switched Token Ring network (see [Figure 15-1](#)). 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 15-1 Interconnected Token Ring TrBRF and TrCRF VLANs

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

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

**Caution**

Certain parent TrBRF STP and TrCRF bridge mode configurations can place the logical ports (the connection between the TrBRF and the TrCRF) of the TrBRF in a blocked state. For more information, see the [“VLAN Configuration Guidelines and Restrictions”](#) section on page 15-8.

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.

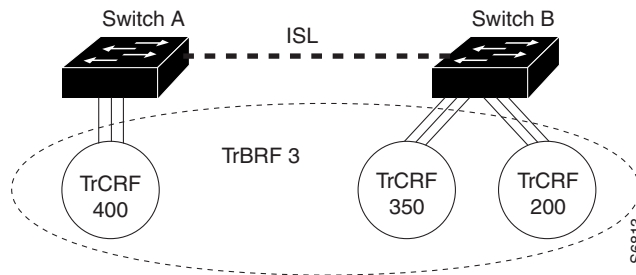
Token Ring TrCRF VLANs

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.

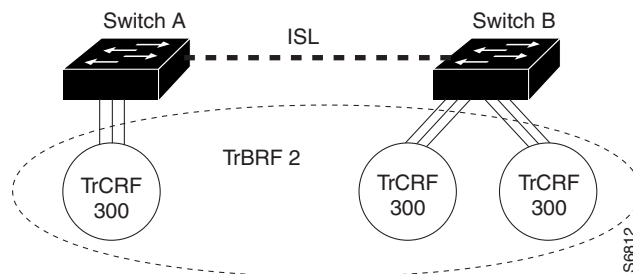
TrCRFs typically are undistributed, which means each TrCRF is limited to the ports on a single network device. Multiple undistributed TrCRFs on the same or separate network devices can be associated with a single parent TrBRF (see [Figure 15-2](#)). The parent TrBRF acts as a multiport bridge, forwarding 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 15-2 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 15-3](#)), and traffic is passed between the default TrCRFs located on separate network devices if the network devices are connected through an ISL trunk.

Figure 15-3 Distributed TrCRF

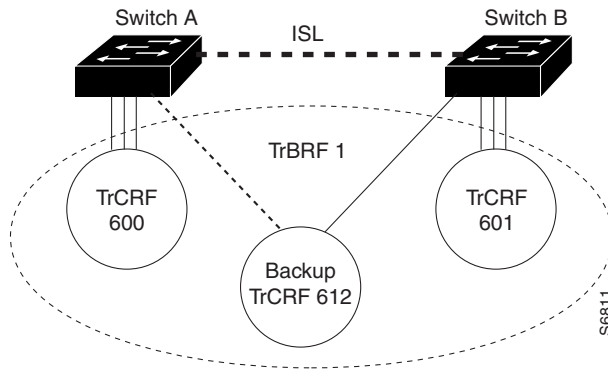
Within a TrCRF, source-route switching forwards frames based on either MAC addresses or route descriptors. The entire VLAN can operate as a single ring, with frames switched between ports within a single TrCRF.

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 15-4](#) illustrates the backup TrCRF.

Figure 15-4 Backup TrCRF



VLAN Default Configuration

Tables 15-2 through 15-6 show the default configurations for the different VLAN media types.

Table 15-2 Ethernet VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1	1–4094
VLAN name	“default” for VLAN 1 “VLANvlan_ID” for other Ethernet VLANs	—
802.10 SAID	10vlan_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 15-3 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 15-3 FDDI VLAN Defaults and Ranges (continued)

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

Table 15-4 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 15-5 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 15-6 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 15-6 Token Ring (TrBRF) VLAN Defaults and Ranges (continued)

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

VLAN Configuration Guidelines and Restrictions

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

- Supervisor engine redundancy does not support nondefault VLAN data file names or locations. Do not enter the **vtp file file_name** command on a switch that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- RPR+ redundancy (see [Chapter 9, “Configuring RPR and RPR+ Supervisor Engine Redundancy”](#)) does not support a configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy.
- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode. See the [“VLAN Configuration Options” section on page 15-9](#).
- Before you can create a VLAN, the Catalyst 6500 series switch must be in VTP server mode or VTP transparent mode. For information on configuring VTP, see [Chapter 14, “Configuring VTP.”](#)
- The VLAN configuration is stored in the vlan.dat file, which is stored in nonvolatile memory. You can cause inconsistency in the VLAN database if you manually delete the vlan.dat file. If you want to modify the VLAN configuration or VTP, use the commands described in this guide and in the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX publication.
- To do a complete backup of your configuration, include the vlan.dat file in the backup.
- The Cisco IOS **end** command is not supported in VLAN database mode.
- You cannot enter **Ctrl-Z** to exit VLAN database mode.
- Catalyst 6500 series switches do not support Token Ring or FDDI media. The switch does not forward FDDI, FDDI-Net, TrCRF, or TrBRF traffic, but it can propagate the VLAN configuration through VTP.
- When a Catalyst 6500 series switch is configured as a VTP server, you can configure FDDI and Token Ring VLANs from the switch.
- You must configure a TrBRF before you configure the TrCRF (the parent TrBRF VLAN you specify must exist).
- In a Token Ring environment, the logical interfaces (the connection between the TrBRF and the TrCRF) of the TrBRF are placed in a blocked state if either of these conditions exists:
 - The TrBRF is running the IBM STP, and the TrCRF is in SRT mode.
 - The TrBRF is running the IEEE STP, and the TrCRF is in SRB mode.

Configuring VLANs

These sections describe how to configure VLANs:

- [VLAN Configuration Options, page 15-9](#)
- [Creating or Modifying an Ethernet VLAN, page 15-10](#)
- [Assigning a Layer 2 LAN Interface to a VLAN, page 15-12](#)
- [Configuring the Internal VLAN Allocation Policy, page 15-12](#)
- [Configuring VLAN Translation, page 15-13](#)
- [Mapping 802.1Q VLANs to ISL VLANs, page 15-16](#)

**Note**

VLANs support a number of parameters that are not discussed in detail in this section. For complete information, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX publication.

VLAN Configuration Options

These sections describe the VLAN configuration options:

- [VLAN Configuration in Global Configuration Mode, page 15-9](#)
- [VLAN Configuration in VLAN Database Mode, page 15-10](#)

VLAN Configuration in Global Configuration Mode

If the switch is in VTP server or transparent mode (see the “[Configuring VTP](#)” section on page 14-6), you can configure VLANs in global and config-vlan configuration modes. When you configure VLANs in global and config-vlan configuration modes, the VLAN configuration is saved in the vlan.dat files. To display the VLAN configuration, enter the **show vlan** command.

If the switch is in VLAN transparent mode, use the copy **running-config startup-config** command to save the VLAN configuration to the startup-config file. After you save the running configuration as the startup configuration, use the **show running-config** and **show startup-config** commands to display the VLAN configuration.

**Note**

- When the switch boots, if the VTP domain name and VTP mode in the startup-config and vlan.dat files do not match, the switch uses the configuration in the vlan.dat file.
- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode.

VLAN Configuration in VLAN Database Mode



Note

- VLAN database mode is supported in releases earlier than Release 12.2(18)SXD.
- You cannot configure extended-range VLANs in VLAN database mode. You can configure extended-range VLANs only in global configuration mode. RPR+ redundancy does not support configuration entered in VLAN database mode. Use global configuration mode with RPR+ redundancy.

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

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

Creating or Modifying an Ethernet VLAN

User-configured VLANs have unique IDs from 1 to 4094, except for reserved VLANs (see [Table 15-1 on page 15-2](#)). 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 15-6](#) 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 <i>vlan_ID</i> { [- <i>vlan_ID</i>] [, <i>vlan_ID</i>] } Router(config-vlan)# or Router(vlan)# vlan <i>vlan_ID</i> Router(config)# no vlan <i>vlan_ID</i> Router(config-vlan)# or Router(vlan)# no vlan <i>vlan_ID</i>	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 [<i>id</i> <i>name</i>] <i>vlan</i>	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.
- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode.
- Layer 3 ports and some software features use extended-range VLANs. If the VLAN you are trying to create or modify is being used by a Layer 3 port or a software feature, the switch displays a message and does not modify the VLAN configuration.

When deleting VLANs, note the following information:

- You cannot delete the default VLANs for the different media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005.
- When you delete a VLAN, any LAN ports configured as access ports assigned to that VLAN become inactive. The ports remain associated with the VLAN (and inactive) until you assign them to a new VLAN.

This example shows how to create an Ethernet VLAN in global configuration mode and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 3
Router(config-vlan)# end
Router# show vlan id 3
```

VLAN	Name	Status	Ports
3	VLAN0003	active	

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
3	enet	100003	1500	-	-	-	-	-	0	0

Primary	Secondary	Type	Interfaces

This example shows how to create an Ethernet VLAN in VLAN database mode:

```
Router# vlan database
Router(vlan)# vlan 3
VLAN 3 added:
    Name: VLAN0003
Router(vlan)# exit
APPLY completed.
Exiting....
```

This example shows how to verify the configuration:

```
Router# show vlan name VLAN0003
```

VLAN	Name	Status	Ports
3	VLAN0003	active	

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	Trans1	Trans2
3	enet	100003	1500	-	-	-	-	0	0

Router#

Assigning a Layer 2 LAN Interface to a VLAN

A VLAN created in a management domain remains unused until you assign one or more LAN ports to the VLAN.



Note

Make sure you assign LAN ports to a VLAN of the appropriate type. Assign Ethernet ports to Ethernet-type VLANs.

To assign one or more LAN ports to a VLAN, complete the procedures in the [“Configuring LAN Interfaces for Layer 2 Switching”](#) section on page 11-6.

Configuring the Internal VLAN Allocation Policy


For more information about VLAN allocation, see the [“VLAN Ranges”](#) section on page 15-2.



Note

The internal VLAN allocation policy is applied only following a reload.

To configure the internal VLAN allocation policy, perform this task:

	Command	Purpose
Step 1	Router(config)# vlan internal allocation policy {ascending descending}	Configures the internal VLAN allocation policy.
	Router(config)# no vlan internal allocation policy	Returns to the default (ascending).
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# reload	Applies the new internal VLAN allocation policy.
		 Caution You do not need to enter the reload command immediately. Enter the reload command during a planned maintenance window.

When you configure the internal VLAN allocation policy, note the following information:

- Enter the **ascending** keyword to allocate internal VLANs from 1006 and up.
- Enter the **descending** keyword to allocate internal VLAN from 4094 and down.

This example shows how to configure descending as the internal VLAN allocation policy:

```
Router# configure terminal
Router(config)# vlan internal allocation policy descending
```

Configuring VLAN Translation

On trunk ports, you can translate one VLAN number to another VLAN number, which transfers all traffic received in one VLAN to the other VLAN.

These sections describe VLAN translation:

- [VLAN Translation Guidelines and Restrictions, page 15-13](#)
- [Configuring VLAN Translation on a Trunk Port, page 15-15](#)
- [Enabling VLAN Translation on Other Ports in a Port Group, page 15-15](#)



Note

- Release 12.2(17b)SXA and later releases support VLAN translation.
- To avoid spanning tree loops, be careful not to misconfigure the VLAN translation feature.

VLAN Translation Guidelines and Restrictions

When translating VLANs, follow these guidelines and restrictions:

- A VLAN translation configuration is inactive if it is applied to ports that are not Layer 2 trunks.
- Do not configure translation of ingress native VLAN traffic on an 802.1Q trunk. Because 802.1Q native VLAN traffic is untagged, it cannot be recognized for translation. You can translate traffic from other VLANs to the native VLAN of an 802.1Q trunk.
- Do not remove the VLAN to which you are translating from the trunk.
- The VLAN translation configuration applies to all ports in a port group. VLAN translation is disabled by default on all ports in a port group. Enable VLAN translation on ports as needed.
- The following table lists:
 - The modules that support VLAN translation
 - The port groups to which VLAN translation configuration applies
 - The number of VLAN translations supported by the port groups
 - The trunk types supported by the modules



Note

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

Product Number	Number of Ports	Number of Port Groups	Port Ranges per Port Group	Translations per Port Group	VLAN Translation Trunk-Type Support
WS-SUP720	2	1	1–2	32	802.1Q

Product Number	Number of Ports	Number of Port Groups	Port Ranges per Port Group	Translations per Port Group	VLAN Translation Trunk-Type Support
WS-X6704-10GE	4	4	1 port in each group	128	ISL 802.1Q
WS-X6502-10GE	1	1	1 port in 1 group	32	802.1Q
WS-X6724-SFP	24	2	1–12 13–24	128	ISL 802.1Q
WS-X6816-GBIC	16	2	1–8 9–16	32	802.1Q
WS-X6516A-GBIC	16	2	1–8 9–16	32	802.1Q
WS-X6516-GBIC	16	2	1–8 9–16	32	802.1Q
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 11-7](#).

Configuring VLAN Translation on a Trunk Port

To translate VLANs on a trunk port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	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 <i>original_vlan_ID translated_vlan_ID</i>	Translates a VLAN to another VLAN. The valid range is 1 to 4094.
	Router(config-if)# no switchport vlan mapping { all <i>original_vlan_ID translated_vlan_ID</i> }	Deletes the mapping.
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show interface <i>type</i> ¹ <i>slot/port</i> 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 <i>type</i> ¹ <i>slot/port</i>	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 <i>type</i> ¹ <i>slot/port</i> vlan mapping	Verifies the VLAN mapping.

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

This example shows how to enable VLAN translation on a port:

```
Router# configure terminal
Router(config)# interface gigabitethernet 5/2
Router(config-if)# switchport vlan mapping enable
Router(config-if)# end
Router#
```

Mapping 802.1Q VLANs to ISL VLANs

The valid range of user-configurable ISL VLANs is 1 through 1001 and 1006 through 4094. The valid range of VLANs specified in the IEEE 802.1Q standard is 1 to 4094. You can map 802.1Q VLAN numbers to ISL VLAN numbers.

802.1Q VLANs in the range 1 through 1001 and 1006 through 4094 are automatically mapped to the corresponding ISL VLAN. 802.1Q VLAN numbers corresponding to reserved VLAN numbers must be mapped to an ISL VLAN in order to be recognized and forwarded by Cisco network devices.

These restrictions apply when mapping 802.1Q VLANs to ISL VLANs:

- You can configure up to eight 802.1Q-to-ISL VLAN mappings on the Catalyst 6500 series switch.
- You can only map 802.1Q VLANs to Ethernet-type ISL VLANs.
- Do not enter the native VLAN of any 802.1Q trunk in the mapping table.
- When you map an 802.1Q VLAN to an ISL VLAN, traffic on the 802.1Q VLAN corresponding to the mapped ISL VLAN is blocked. For example, if you map 802.1Q VLAN 1007 to ISL VLAN 200, traffic on 802.1Q VLAN 200 is blocked.
- VLAN mappings are local to each Catalyst 6500 series switch. Make sure you configure the same VLAN mappings on all appropriate network devices.

To map an 802.1Q VLAN to an ISL VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# vlan mapping dot1q <i>dot1q_vlan_ID</i> isl <i>isl_vlan_ID</i>	Maps an 802.1Q VLAN to an ISL Ethernet VLAN. The valid range for <i>dot1q_vlan_ID</i> is 1001 to 4094. The valid range for <i>isl_vlan_ID</i> is the same.
	Router(config)# no vlan mapping dot1q {all <i>dot1q_vlan_ID</i> }	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
```




Configuring Private VLANs

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



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How Private VLANs Work, page 16-1](#)
- [Private VLAN Configuration Guidelines and Restrictions, page 16-2](#)
- [Configuring Private VLANs, page 16-5](#)

Understanding How Private VLANs Work



Note

To configure private VLANs, the switch must be in VTP transparent mode.

Private VLANs provide Layer 2 isolation between ports within the same private VLAN. There are three types of private VLAN ports:

- **Promiscuous**—A promiscuous port can communicate with all interfaces, including the community and isolated ports within a private VLAN.
- **Isolated**—An isolated port has complete Layer 2 separation from other ports within the same private VLAN except for the promiscuous port. 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**—Community ports communicate among themselves and with their promiscuous ports. These interfaces are isolated at Layer 2 from all other interfaces in other communities or isolated ports within their private VLAN.



Note

Because trunks can support the VLANs carrying traffic between isolated, community, and promiscuous ports, isolated and community port traffic might enter or leave the switch through a trunk interface.

Private VLAN ports are associated with a set of supporting VLANs that are used to create the private VLAN structure. A private VLAN uses VLANs three ways:

- Primary VLAN—Carries traffic from promiscuous ports to isolated, community, and other promiscuous ports.
- Isolated VLAN—Carries traffic from isolated ports to promiscuous ports.
- Community VLAN—Carries traffic between community ports and to promiscuous ports. You can configure multiple community VLANs in a private VLAN.

**Note**

Isolated and community VLANs are both called secondary VLANs.

You can extend private VLANs across multiple devices by trunking the primary, isolated, and community VLANs to other devices that support private VLANs.

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 only need to communicate with a default gateway to gain access outside the private VLAN. With end stations in a private VLAN, you can do the following:

- Designate selected ports connected to end stations (for example, interfaces connected to servers) as isolated to prevent any communication at Layer 2. (For example, if the end stations were servers, this configuration would prevent Layer 2 communication between the servers.)
- Designate the interfaces to which the default gateway(s) and selected end stations (for example, backup servers or LocalDirector) are attached as promiscuous to allow all end stations access.
- Reduce VLAN and IP subnet consumption by preventing traffic between end stations even though they are in the same VLAN and IP subnet.

A promiscuous port can serve only one primary VLAN.

A promiscuous port can serve as many isolated or community VLANs as desired.

With a promiscuous port, you can connect a wide range of devices as “access points” to a private VLAN. For example, you can connect a promiscuous port to the “server port” of LocalDirector to connect an isolated VLAN or a number of community VLANs to the server so that LocalDirector can load balance the servers present in the isolated or community VLANs, or you can use a promiscuous port to monitor or back up all the private VLAN servers from an administration workstation.

Private VLAN Configuration Guidelines and Restrictions

When configuring private VLANs, follow these guidelines and restrictions:

- Set VTP to transparent mode. After you configure a private VLAN, you cannot change the VTP mode to client or server. See [Chapter 14, “Configuring VTP.”](#)
- You cannot include VLAN 1 or VLANs 1002 to 1005 in the private VLAN configuration.
- 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.
- Configure Layer 3 VLAN interfaces only for primary VLANs. Layer 3 VLAN interfaces for isolated and community VLANs are inactive while the VLAN is configured as an isolated or community VLAN.

- Do not configure private VLAN ports as EtherChannels. While a port is part of the private VLAN configuration, any EtherChannel configuration for it is inactive.
- With releases earlier than Release 12.2(18)SX, you cannot configure port security on ports that are in a private VLAN.
- Destination SPAN configuration supersedes private VLAN configuration. While a port is a destination SPAN port, any private VLAN configuration for it is inactive.
- Private VLANs support the following 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 51, “Configuring Local SPAN, RSPAN, and ERSPAN.”](#)

- 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.
- Enable PortFast and BPDU guard on isolated and community ports to prevent STP loops due to misconfigurations and to speed up STP convergence (see [Chapter 22, “Configuring Optional STP Features”](#)). When enabled, STP applies the BPDU guard feature to all PortFast-configured Layer 2 LAN ports.
- If you delete a VLAN used in the private VLAN configuration, the private VLAN ports associated with the VLAN become inactive.
- 12-Port Restriction:
 - 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.
 - In releases earlier than Release 12.2(17a)SX, the “12-port restriction” applies to these Ethernet switching modules: WS-X6548-RJ-45, WS-X6548-RJ-21, WS-X6524-100FX-MM.
 - In Release 12.2(17a)SX and later releases, 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
- In releases where CSCsb44185 is resolved, a port configured with the **switchport mode dynamic auto** or **switchport mode dynamic desirable** command

While one port within the group of 12 ports is one of these, 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 **shutdown** and **no shutdown** commands.

- 24-Port Restriction:

In all releases, this “24-port restriction” applies to the WS-X6548-GE-TX and WS-X6148-GE-TX 10/100/1000 Mb Ethernet switching modules. Within groups of 24 ports (1–24, 25–48), do not configure ports as isolated ports or community VLAN ports when one port within the group of 24 ports is any of these:

- A trunk port
- A SPAN destination port
- A promiscuous private VLAN port
- In releases where CSCsb44185 is resolved, a port configured with the **switchport mode dynamic auto** or **switchport mode dynamic desirable** command

While one port within the group of 24 ports is one of these, 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 **shutdown** and **no shutdown** commands.

- Private VLAN ports can be on different network devices as long as the devices are trunk connected and the primary and secondary VLANs have not been removed from the trunk.
- VTP does not support private VLANs. You must configure private VLANs on each device where you want private VLAN ports.
- To maintain the security of your private VLAN configuration and 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.
- We recommend that you prune the private VLANs from the trunks on devices that carry no traffic in the private VLANs.
- In networks with some devices using MAC address reduction, and others not using MAC address reduction, STP parameters do not necessarily propagate to ensure that the spanning tree topologies match. You should manually check the STP configuration to ensure that the primary, isolated, and community VLANs’ spanning tree topologies match.
- If you enable MAC address reduction on the switch, we recommend that you enable MAC address reduction on all the devices in your network to ensure that the STP topologies of the private VLANs match.
- In a network where private VLANs are configured, if you enable MAC address reduction on some devices and disable it on others (mixed environment), use the default bridge priorities to make sure that the root bridge is common to the primary VLAN and to all its associated isolated and community VLANs. Be consistent with the ranges employed by the MAC address reduction feature regardless of whether it is enabled on the system. MAC address reduction allows only discrete levels and uses all intermediate values internally as a range. You should disable a root bridge with private VLANs and MAC address reduction, and configure the root bridge with any priority higher than the highest priority range used by any nonroot bridge.
- You can apply different quality of service (QoS) configuration to primary, isolated, and community VLANs (see [Chapter 42, “Configuring PFC QoS”](#)).
- You cannot apply VACLs to secondary VLANs (see the [Chapter 36, “Configuring VLAN ACLs”](#)).
- 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 34, “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.
- Do not apply dynamic access control entries (ACEs) to primary VLANs. Cisco IOS dynamic ACL configuration applied to a primary VLAN is inactive while the VLAN are part of the private VLAN configuration.
- Sticky ARP prevents MAC address spoofing by ensuring that ARP entries (IP address, MAC address, and source VLAN) do not age out. With Release 12.2(18)SXF and later releases, you can configure sticky ARP on a per-interface basis. For information about configuring sticky ARP, see the [“Configuring Sticky ARP” section on page 37-36](#). The following guidelines and restrictions apply to private VLAN sticky ARP:
 - ARP entries learned on Layer 3 private VLAN interfaces are sticky ARP entries (we recommend that you display and verify private VLAN interface ARP entries).
 - For security reasons, private VLAN port sticky ARP entries do not age out. 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
```

Configuring Private VLANs

These sections describe how to configure private VLANs:

- [Configuring a VLAN as a Private VLAN, page 16-6](#)
- [Associating Secondary VLANs with a Primary VLAN, page 16-7](#)
- [Mapping Secondary VLANs to the Layer 3 VLAN Interface of a Primary VLAN, page 16-8](#)
- [Configuring a Layer 2 Interface as a Private VLAN Host Port, page 16-9](#)
- [Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port, page 16-10](#)



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 <i>vlan_ID</i>	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 [<i>type</i>]	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 <i>primary_vlan_ID</i>	Enters VLAN configuration submode for the primary VLAN.
Step 2	Router(config-vlan)# private-vlan association { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	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 [<i>type</i>]	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 <i>primary_vlan_ID</i>	Enters interface configuration mode for the primary VLAN.
Step 2	Router(config-if)# private-vlan mapping { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	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 <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching: <ul style="list-style-type: none"> 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.
Step 4	Router(config-if)# switchport private-vlan host-association <i>primary_vlan_ID</i> <i>secondary_vlan_ID</i>	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 [<i>type</i> ¹ <i>slot/port</i>] 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 <i>type</i> ¹ <i>slot/port</i>	Selects the LAN interface to configure.
Step 2	Router(config-if)# switchport	Configures the LAN interface for Layer 2 switching: <ul style="list-style-type: none"> You must enter the switchport command once without any keywords to configure the LAN interface as a Layer 2 interface before you can enter additional switchport commands with keywords. Required only if you have not entered the switchport command already for the interface.
Step 3	Router(config-if)# switchport mode private-vlan { host promiscuous }	Configures the Layer 2 port as a private VLAN promiscuous port.
	Router(config-if)# no switchport mode private-vlan	Clears the private VLAN port configuration.
Step 4	Router(config-if)# switchport private-vlan mapping <i>primary_vlan_ID</i> { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	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 [<i>type</i> ¹ <i>slot/port</i>] 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
```




Configuring Cisco IP Phone Support

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



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding Cisco IP Phone Support, page 17-1](#)
- [Default Cisco IP Phone Support Configuration, page 17-4](#)
- [Cisco IP Phone Support Configuration Guidelines and Restrictions, page 17-4](#)
- [Configuring Cisco IP Phone Support, page 17-5](#)

Understanding Cisco IP Phone Support

These sections describe Cisco IP phone support:

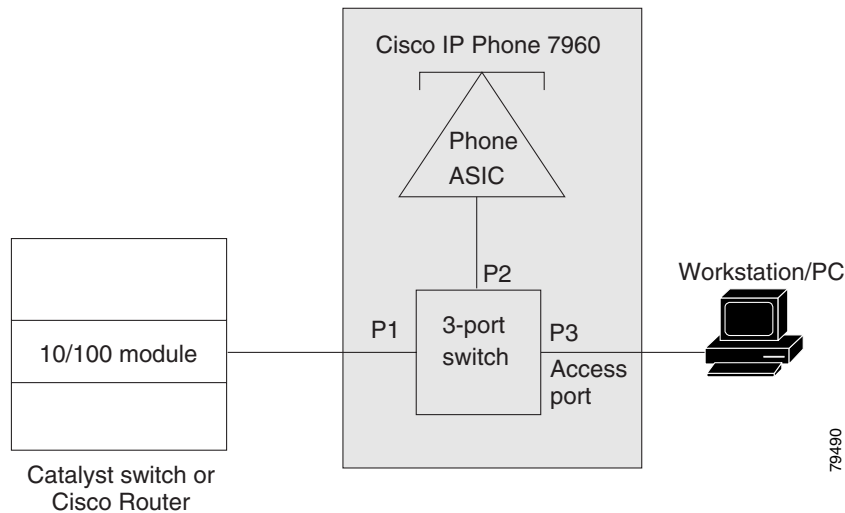
- [Cisco IP Phone Connections, page 17-1](#)
- [Cisco IP Phone Voice Traffic, page 17-2](#)
- [Cisco IP Phone Data Traffic, page 17-3](#)
- [Cisco IP Phone Power Configurations, page 17-3](#)

Cisco IP Phone Connections

The Cisco IP phone contains an integrated 3-port 10/100 switch. The ports are dedicated connections to these devices:

- Port 1 connects to the switch.
- Port 2 is an internal 10/100 interface that carries the Cisco IP phone traffic.
- Port 3 connects to a PC or other device.

[Figure 17-1](#) shows a Cisco IP phone connected between a switch and a PC.

Figure 17-1 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 42, “Configuring PFC QoS”](#)).



Note

You can configure the ports on WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules to trust received Layer 2 CoS values (QoS port architecture 1p1q0t/1p3q1t). The WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules cannot supply power to Cisco IP phones. Configure QoS policies that use the Layer 3 IP precedence value on other switching modules.

You can configure a Layer 2 access port with an attached Cisco IP phone to use one VLAN for voice traffic and another VLAN for data traffic from a device attached to the Cisco IP phone.

You can configure Layer 2 access ports on the switch to send Cisco Discovery Protocol (CDP) packets that instruct an attached Cisco IP phone to transmit voice traffic to the switch in any of the following ways:

- In the voice VLAN, tagged with a Layer 2 CoS priority value
- In the access VLAN, tagged with a Layer 2 CoS priority value
- In the access VLAN, untagged (no Layer 2 CoS priority value)



Note

In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5 for voice traffic and 3 for voice control traffic).

You cannot use Cisco IOS software commands to configure the frame type used by data traffic sent from a device attached to the access port on the Cisco IP phone.

Cisco IP Phone Data Traffic

**Note**

Untagged traffic from the device attached to the Cisco IP phone passes through the Cisco IP phone unchanged, regardless of the trust state of the access port on the Cisco IP phone.

To process tagged data traffic (traffic in 802.1Q or 802.1p frame types) from the device attached to the access port on the Cisco IP phone (see [Figure 17-1](#)), you can configure Layer 2 access ports on the switch to send CDP packets that instruct an attached Cisco IP phone to configure the access port on the Cisco IP phone to either of these two modes:

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

Cisco IP Phone Power Configurations

These sections describe Cisco IP phone power configurations:

- [Locally Powered Cisco IP Phones, page 17-3](#)
- [Inline-Powered Cisco IP Phones, page 17-3](#)

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 at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/ol_4164.htm

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 Catalyst 6500 series switch port connected to the Cisco IP phone to send configuration information to the Cisco IP phone.
- You can configure a voice VLAN only on a Layer 2 LAN port.
- You can configure the ports on WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules to trust received Layer 2 CoS values (QoS port architecture 1p1q0t/1p3q1t). The WS-X6548-RJ-45 and WS-X6548-RJ-21 switching modules cannot supply power to Cisco IP phones.

- You cannot configure 10/100 Mbps ports with QoS port architecture 1p4t/2q2t to trust received Layer 2 CoS values. Configure policies to trust the Layer 3 IP precedence value on switching modules with QoS port architecture 1p4t/2q2t.
- The following conditions indicate that the Cisco IP phone and a device attached to the Cisco IP phone are in the same VLAN and must be in the same IP subnet:
 - If they both use 802.1p or untagged frames
 - If the Cisco IP phone uses 802.1p frames and the device uses untagged frames
 - If the Cisco IP phone uses untagged frames and the device uses 802.1p frames
 - If the Cisco IP phone uses 802.1Q frames and the voice VLAN is the same as the access VLAN
- The Cisco IP phone and a device attached to the Cisco IP phone cannot communicate if they are in the same VLAN and subnet but use different frame types, because traffic between devices in the same subnet is not routed (routing would eliminate the frame type difference).
- You cannot use Cisco IOS software commands to configure the frame type used by traffic sent from a device attached to the access port on the Cisco IP phone.
- If you enable port security on a port configured with a voice VLAN and if there is a PC connected to the Cisco IP phone, set the maximum allowed secure addresses on the port to at least 3.
- You cannot configure static secure MAC addresses in the voice VLAN.
- Ports configured with a voice VLAN can be secure ports (refer to [Chapter 47, “Configuring Port Security”](#)).
- In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5 for voice traffic and 3 for voice control traffic).

Configuring Cisco IP Phone Support

These sections describe how to configure Cisco IP phone support:

- [Configuring Voice Traffic Support, page 17-5](#)
- [Configuring Data Traffic Support, page 17-7](#)
- [Configuring Inline Power Support, page 17-8](#)



Note

Voice VLANs are referred to as *auxiliary VLANs* in the Catalyst software publications.

Configuring Voice Traffic Support

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

	Command	Purpose
Step 1	Router(config)# interface fastethernet <i>slot/port</i>	Selects the port to configure.
Step 2	Router(config-if)# switchport voice vlan { <i>voice_vlan_ID</i> 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 switch puts the 802.1Q voice traffic into the voice VLAN.
- Enter the **dot1p** keyword to send CDP packets that configure the Cisco IP phone to transmit voice traffic in 802.1p frames, tagged with VLAN ID 0 and a Layer 2 CoS value (the default is 5 for voice traffic and 3 for voice control traffic). The switch puts the 802.1p voice traffic into the access VLAN.
- Enter the **untagged** keyword to send CDP packets that configure the Cisco IP phone to transmit untagged voice traffic. The switch puts the untagged voice traffic into the access VLAN.
- Enter the **none** keyword to allow the Cisco IP phone to use its own configuration and transmit untagged voice traffic. The switch puts the untagged voice traffic into the access VLAN.
- In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5).
- Refer to [Chapter 42, “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 11-14 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 <i>slot/port</i>	Selects the port to configure.
Step 2	Router(config-if)# mls qos trust extend [cos <i>cos_value</i>]	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 <i>slot/port</i> switchport Router# show running-config interface fastethernet <i>slot/port</i>	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 <i>slot/port</i>	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 <i>slot/port</i>]	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
```



Configuring IEEE 802.1Q Tunneling

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



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX switching modules do not support IEEE 802.1Q tunneling.

This chapter consists of these sections:

- [Understanding How 802.1Q Tunneling Works, page 18-1](#)
- [802.1Q Tunneling Configuration Guidelines and Restrictions, page 18-3](#)
- [Configuring 802.1Q Tunneling, page 18-6](#)

Understanding How 802.1Q Tunneling Works

802.1Q tunneling enables service providers to use a single VLAN to support customers who have multiple VLANs, while preserving customer VLAN IDs and keeping traffic in different customer VLANs segregated.

A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that you dedicate to tunneling, which then becomes a tunnel VLAN. To keep customer traffic segregated, each customer requires a separate tunnel VLAN, but that one tunnel VLAN supports all of the customer's VLANs.

802.1Q tunneling is not restricted to point-to-point tunnel configurations. Any tunnel port in a tunnel VLAN is a tunnel entry and exit point. An 802.1Q tunnel can have as many tunnel ports as are needed to connect customer switches.

The customer switches are trunk connected, but with 802.1Q tunneling, the service provider switches only use one service provider VLAN to carry all the customer VLANs, instead of directly carrying all the customer VLANs.

With 802.1Q tunneling, tagged customer traffic comes from an 802.1Q trunk port on a customer device and enters the service-provider edge switch through a tunnel port. The link between the 802.1Q trunk port on a customer device and the tunnel port is called an asymmetrical link because one end is

configured as an 802.1Q trunk port and the other end is configured as a tunnel port. You assign the tunnel port to an access VLAN ID unique to each customer. See [Figure 18-1 on page 18-2](#) and [Figure 18-2 on page 18-3](#).

Figure 18-1 IEEE 802.1Q Tunnel Ports in a Service-Provider Network

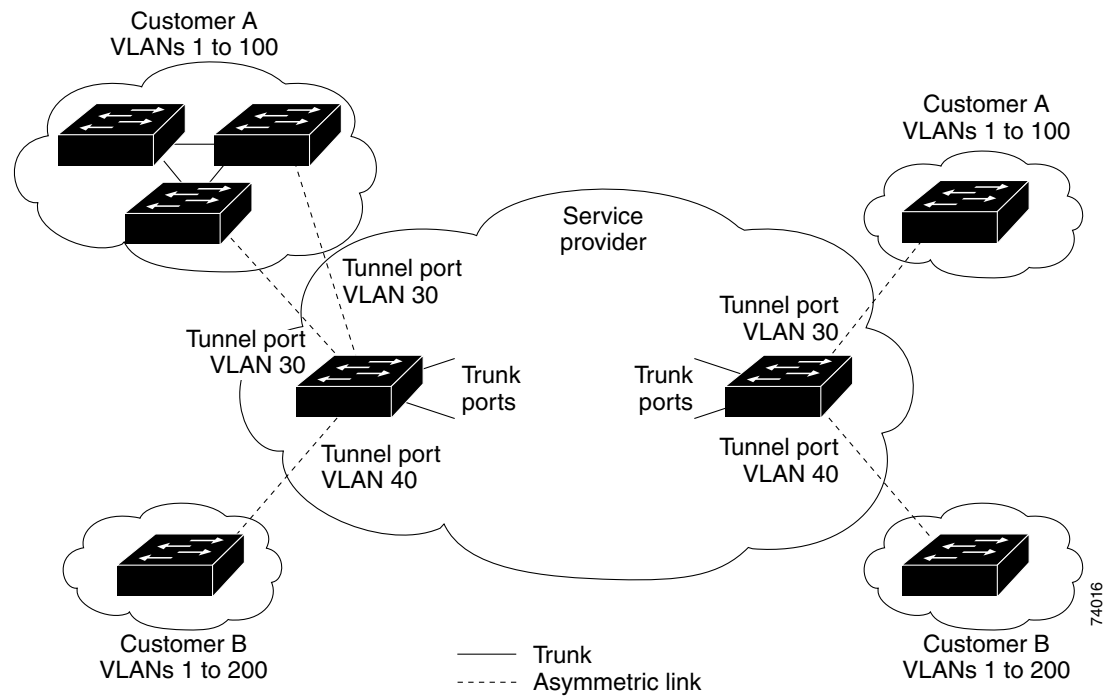
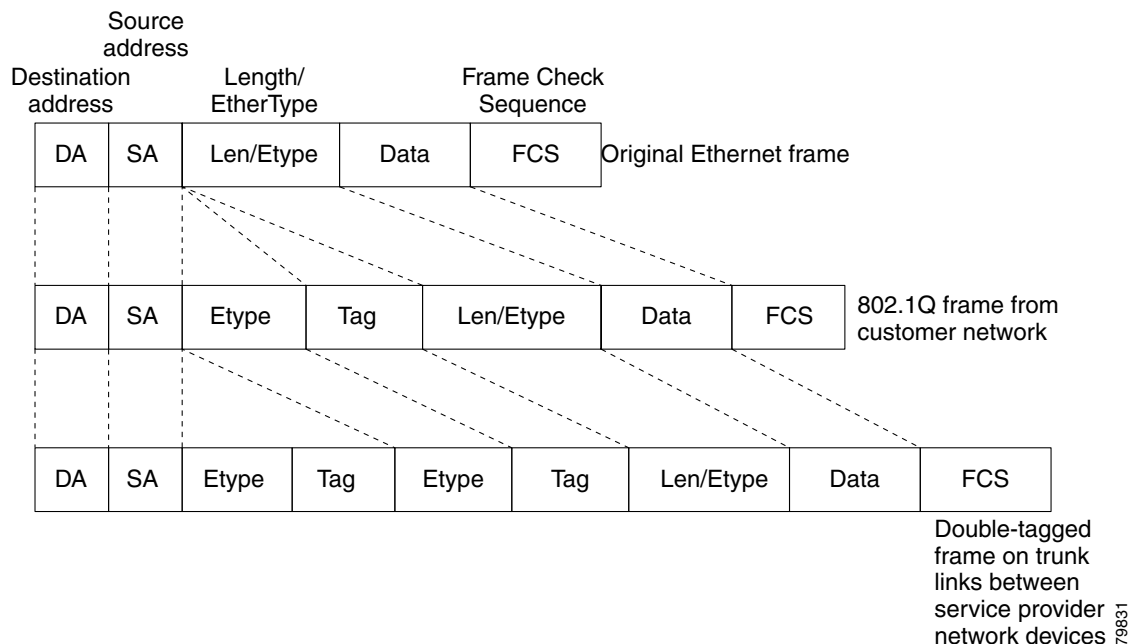


Figure 18-2 Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames

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 10-10](#).
 - Take note of the modules listed in the “Configuring Jumbo Frame Support” section that do not support jumbo frames.
- Jumbo frames can be tunneled as long as the jumbo frame length combined with the 802.1Q tag does not exceed the maximum frame size.
- Because tunnel traffic has the added ethertype and length field and retains the 802.1Q tag within the switch, the following restrictions exist:
 - The Layer 3 packet within the Layer 2 frame cannot be identified in tunnel traffic.
 - Layer 3 and higher parameters cannot be identified in tunnel traffic (for example, Layer 3 destination and source addresses).
 - Because the Layer 3 addresses cannot be identified within the packet, tunnel traffic cannot be routed.
 - The switch can provide only MAC-layer filtering for tunnel traffic (VLAN IDs and source and destination MAC addresses).
 - The switch can provide only MAC-layer access control and QoS for tunnel traffic.
 - QoS cannot detect the received CoS value in the 802.1Q 2-byte Tag Control Information field.
- On an asymmetrical link, the Cisco Discovery Protocol (CDP) reports a native VLAN mismatch if the VLAN of the tunnel port does not match the native VLAN of the 802.1Q trunk. The 802.1Q tunnel feature does not require that the VLANs match. Ignore the messages if your configuration requires nonmatching VLANs.
- Asymmetrical links do not support the Dynamic Trunking Protocol (DTP) because only one port on the link is a trunk. Configure the 802.1Q trunk port on an asymmetrical link to trunk unconditionally.
- The 802.1Q tunneling feature cannot be configured on ports configured to support private VLANs.
- The following Layer 2 protocols work between devices connected by an asymmetrical link:
 - CDP
 - UniDirectional Link Detection (UDLD)
 - Port Aggregation Protocol (PAgP)
 - Link Aggregation Control Protocol (LACP)

- 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 19, “Configuring Layer 2 Protocol Tunneling,”](#) for configuration details.

- To configure an EtherChannel as an asymmetrical link, all ports in the EtherChannel must have the same tunneling configuration. Because the Layer 3 packet within the Layer 2 frame cannot be identified, you must configure the EtherChannel to use MAC-address-based frame distribution.

The following configuration guidelines are *required* for your Layer 2 protocol tunneling configuration:

- On all the service provider edge switches, PortFast BPDU filtering must be enabled on the 802.1Q tunnel ports as follows:

```
Router(config-if)# spanning-tree bpdupfilter 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 switches, tag native VLAN egress traffic and drop untagged native VLAN ingress traffic by entering the following command:

```
Router(config)# vlan dot1q tag native
```

- On all the customer switches, *either* enable or disable the global **vlan dot1q tag native** option.



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

The following configuration guidelines are *optional* for your Layer 2 protocol tunneling configuration:

- Because all the BPDUs are being dropped, spanning tree PortFast can be enabled on Layer 2 protocol tunnel ports as follows:

```
Router(config-if)# spanning-tree portfast trunk
```

- If the service provider does not want the customer to see its switches, CDP should be disabled on the 802.1Q tunnel port as follows:

```
Router(config-if)# no cdp enable
```

Configuring 802.1Q Tunneling

These sections describe 802.1Q tunneling configuration:

- [Configuring 802.1Q Tunnel Ports, page 18-6](#)
- [Configuring the Switch to Tag Native VLAN Traffic, page 18-6](#)



Caution

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

Configuring 802.1Q Tunnel Ports

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

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching: <ul style="list-style-type: none"> • You must enter the switchport command once without any keywords to configure the LAN port as a Layer 2 interface before you can enter additional switchport commands with keywords. • Required only if you have not entered the switchport command already for the interface.
Step 3	Router(config-if)# switchport mode dot1q-tunnel Router(config-if)# no switchport mode dot1q-tunnel	Configures the Layer 2 port as a tunnel port. Clears the tunnel port configuration.
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show dot1q-tunnel [{ interface <i>type</i> <i>interface-number</i> }]	Verifies the configuration.

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

This example shows how to configure tunneling on port 4/1 and verify the configuration:

```
Router# configure terminal
Router(config)# interface fastethernet 4/1
Router(config-if)# switchport mode dot1q-tunnel
Router(config-if)# end
Router# show dot1q-tunnel interface
```

Configuring the Switch to Tag Native VLAN Traffic

The **vlan dot1q tag native** command is a global command that configures the switch to tag native VLAN traffic, and admit only 802.1Q tagged frames on 802.1Q trunks, dropping any untagged traffic, including untagged traffic in the native VLAN.

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

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

This example shows how to configure the switch to tag native VLAN traffic and verify the configuration:

```
Router# configure terminal
Router(config)# vlan dot1q tag native
Router(config)# end
Router# show vlan dot1q tag native
```




Configuring Layer 2 Protocol Tunneling

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



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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:

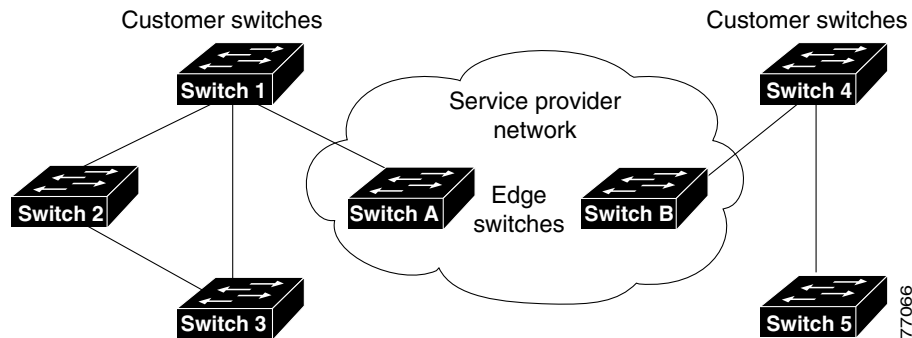
- [Understanding How Layer 2 Protocol Tunneling Works, page 19-1](#)
- [Configuring Support for Layer 2 Protocol Tunneling, page 19-2](#)

Understanding How Layer 2 Protocol Tunneling Works

Layer 2 protocol tunneling allows Layer 2 protocol data units (PDUs) (CDP, STP, and VTP) to be tunneled through a network. This section uses the following terminology:

- Edge switch—The switch connected to the customer switch and placed on the boundary of the service provider network (see [Figure 19-1](#)).
- Layer 2 protocol tunnel port—A port on the edge switch on which a specific tunneled protocol can be encapsulated or deencapsulated. The Layer 2 protocol tunnel port is configured through CLI commands.
- Tunneled PDU—A CDP, STP, or VTP PDU.

Without Layer 2 protocol tunneling, tunnel ports drop STP and VTP packets and process CDP packets. This handling of the PDUs creates different spanning tree domains (different spanning tree roots) for the customer switches. For example, STP for a VLAN on switch 1 (see [Figure 19-1](#)) 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 19-1 Layer 2 Protocol Tunneling Network Configuration

GBPT provides a scalable approach to PDU tunneling by software encapsulating the PDUs in the ingress edge switches and then multicasting them in hardware. All switches inside the service provider network treat these encapsulated frames as data packets and forward them to the other end. The egress edge switch listens for these special encapsulated frames and deencapsulates them; they are then forwarded out of the tunnel.

The encapsulation involves rewriting the destination media access control (MAC) address in the PDU. An ingress edge switch rewrites the destination MAC address of the PDUs received on a Layer 2 tunnel port with the Cisco proprietary multicast address (01-00-0c-cd-cd-d0). The PDU is then flooded to the native VLAN of the Layer 2 tunnel port. If you enable Layer 2 protocol tunneling on a port, PDUs of an enabled protocol are not sent out. If you disable Layer 2 protocol tunneling on a port, the disabled protocols function the same way they were functioning before Layer 2 protocol tunneling was disabled on the port.

Configuring Support for Layer 2 Protocol Tunneling



Note

- Encapsulated PDUs received by an 802.1Q tunnel port are transmitted from other tunnel ports in the same VLAN on the switch.
- Configure jumbo frame support on Layer 2 protocol tunneling ports:
 - See the [“Configuring Jumbo Frame Support”](#) section on page 10-10.
 - Take note of the modules listed in the “Configuring Jumbo Frame Support” section that do not support jumbo frames.

To configure Layer 2 protocol tunneling on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching: <ul style="list-style-type: none"> 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 [<i>packets</i>] shutdown-threshold [<i>packets</i>] 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 <i>type</i> ¹ <i>slot/port</i> summary]	Verifies the configuration.

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

When you configure a Layer 2 port as a Layer 2 protocol tunnel port, note the following information:

- Optionally, you may specify a drop threshold for the port. The drop threshold value, from 1 to 4096, determines the number of packets to be processed for that protocol on that interface in one second. When the drop threshold is exceeded, PDUs for the specified protocol are dropped for the remainder of the 1-second period. If a shutdown threshold is not specified, the value is 0 (shutdown threshold disabled).
- Optionally, you may specify a shutdown threshold for the port. The shutdown threshold value, from 1 to 4096, determines the number of packets to be processed for that protocol on that interface in one second. When the shutdown threshold is exceeded, the port is put in errdisable state. If a shutdown threshold is not specified, the value is 0 (shutdown threshold disabled).



Note

Refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX 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#

```



Configuring Standard-Compliant IEEE MST

This chapter describes how to configure the standard-compliant IEEE Multiple Spanning Tree (MST) protocol on Catalyst 6500 series switches.



Note

- The IEEE MST protocol has transitioned from a prestandard state to a released state. This chapter describes the standard-compliant MST implementation supported in Release 12.2(18)SXF and later releases. [Chapter 21, “Configuring STP and Prestandard IEEE 802.1s MST,”](#) describes the prestandard MST implementation supported in releases earlier than Release 12.2(18)SXF.
- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL: <http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding MST, page 20-1](#)
- [Understanding RSTP, page 20-9](#)
- [Configuring MST, page 20-15](#)
- [Displaying the MST Configuration and Status, page 20-28](#)

Understanding MST

These sections describe MST:

- [MST Overview, page 20-2](#)
- [MST Regions, page 20-2](#)
- [IST, CIST, and CST, page 20-3](#)
- [Hop Count, page 20-6](#)
- [Boundary Ports, page 20-6](#)
- [Standard-Compliant MST Implementation, page 20-7](#)
- [Interoperability with IEEE 802.1D-1998 STP, page 20-9](#)

MST Overview

MST maps multiple VLANs into a spanning tree instance, with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of spanning tree instances required to support a large number of VLANs. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

The most common initial deployment of MST is in the backbone and distribution layers of a Layer 2 switched network. This deployment provides the kind of highly available network that is required in a service-provider environment.

MST provides rapid spanning tree convergence through explicit handshaking, which eliminates the 802.1D forwarding delay and quickly transitions root bridge ports and designated ports to the forwarding state.

MST improves spanning tree operation and maintains backward compatibility with these STP versions:

- Original 802.1D spanning tree
- Existing Cisco-proprietary Multiple Instance STP (MISTP)
- Existing Cisco per-VLAN spanning tree plus (PVST+)
- Rapid per-VLAN spanning tree plus (rapid PVST+)

For information about PVST+ and rapid PVST+, see [Chapter 21, “Configuring STP and Prestandard IEEE 802.1s MST.”](#) For information about other spanning tree features such as Port Fast, UplinkFast, root guard, and so forth, see [Chapter 22, “Configuring Optional STP Features.”](#)

**Note**

- IEEE 802.1w defined the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.
- IEEE 802.1s defined MST and was incorporated into IEEE 802.1Q.

MST Regions

For switches to participate in MST instances, you must consistently configure the switches with the same MST configuration information. A collection of interconnected switches that have the same MST configuration comprises an MST region as shown in [Figure 20-1 on page 20-5](#).

The MST configuration controls to which MST region each switch belongs. The configuration includes the name of the region, the revision number, and the MST VLAN-to-instance assignment map.

A region can have one or multiple members with the same MST configuration; each member must be capable of processing RSTP bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network, but each region can support up to 65 spanning tree instances. Instances can be identified by any number in the range from 0 to 4094. You can assign a VLAN to only one spanning tree instance at a time.

IST, CIST, and CST

These sections describe internal spanning tree (IST), common and internal spanning tree (CIST), and common spanning tree (CST):

- [IST, CIST, and CST Overview, page 20-3](#)
- [Spanning Tree Operation Within an MST Region, page 20-3](#)
- [Spanning Tree Operations Between MST Regions, page 20-4](#)
- [IEEE 802.1s Terminology, page 20-5](#)

IST, CIST, and CST Overview

Unlike other spanning tree protocols, in which all the spanning tree instances are independent, MST establishes and maintains IST, CIST, and CST spanning trees:

- An IST is the spanning tree that runs in an MST region.

Within each MST region, MST maintains multiple spanning tree instances. Instance 0 is a special instance for a region, known as the IST. All other MST instances are numbered from 1 to 4094.

The IST is the only spanning tree instance that sends and receives BPDUs. All of the other spanning tree instance information is contained in MSTP records (M-records), which are encapsulated within MST BPDUs. Because the MST BPDU carries information for all instances, the number of BPDUs that need to be processed to support multiple spanning tree instances is significantly reduced.

All MST instances within the same region share the same protocol timers, but each MST instance has its own topology parameters, such as root bridge ID, root path cost, and so forth. By default, all VLANs are assigned to the IST.

An MST instance is local to the region; for example, MST instance 1 in region A is independent of MST instance 1 in region B, even if regions A and B are interconnected.

- A CIST is a collection of the ISTs in each MST region.
- The CST interconnects the MST regions and single spanning trees.

The spanning tree computed in a region appears as a subtree in the CST that encompasses the entire switched domain. The CIST is formed by the spanning tree algorithm running among switches that support the 802.1w, 802.1s, and 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

For more information, see the [“Spanning Tree Operation Within an MST Region” section on page 20-3](#) and the [“Spanning Tree Operations Between MST Regions” section on page 20-4](#).

Spanning Tree Operation Within an MST Region

The IST connects all the MST switches in a region. When the IST converges, the root of the IST becomes the CIST regional root (called the *IST master* before the implementation of the 802.1s standard) as shown in [Figure 20-1 on page 20-5](#). The CIST regional root is also the CIST root if there is only one region in the network. If the CIST root is outside the region, one of the MST switches at the boundary of the region is selected as the CIST regional root.

When an MST switch initializes, it sends BPDUs that identify itself as the root of the CIST and the CIST regional root, with both of the path costs to the CIST root and to the CIST regional root set to zero. The switch also initializes all of its MST instances and claims to be the root for all of them. If the switch receives superior MST root information (lower switch ID, lower path cost, and so forth) than currently stored for the port, it relinquishes its claim as the CIST regional root.

During initialization, a region might have many subregions, each with its own CIST regional root. As switches receive superior IST information from a neighbor in the same region, they leave their old subregions and join the new subregion that contains the true CIST regional root, which causes all subregions to shrink except for the one that contains the true CIST regional root.

For correct operation, all switches in the MST region must agree on the same CIST regional root. Therefore, any two switches in the region only synchronize their port roles for an MST instance if they converge to a common CIST regional root.

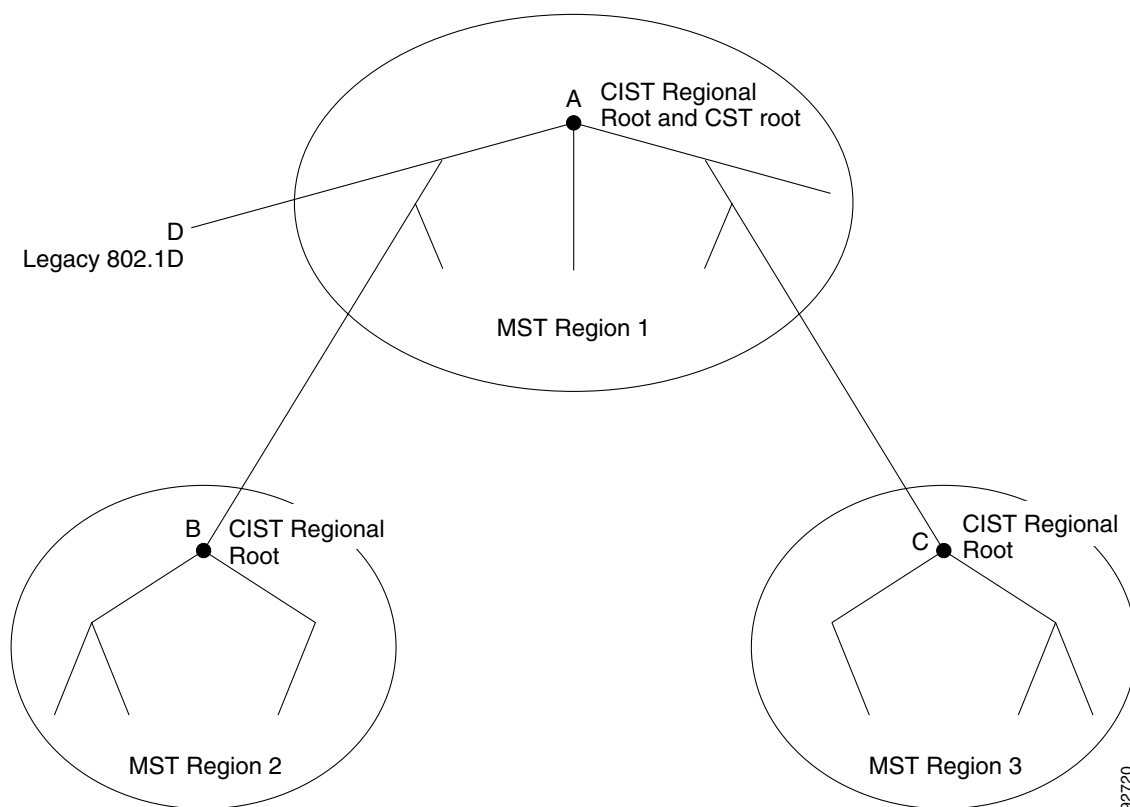
Spanning Tree Operations Between MST Regions

If there are multiple regions or 802.1D switches within the network, MST establishes and maintains the CST, which includes all MST regions and all 802.1D STP switches in the network. The MST instances combine with the IST at the boundary of the region to become the CST.

The IST connects all the MST switches in the region and appears as a subtree in the CIST that encompasses the entire switched domain. The root of the subtree is the CIST regional root. The MST region appears as a virtual switch to adjacent STP switches and MST regions.

[Figure 20-1](#) shows a network with three MST regions and an 802.1D switch (D). The CIST regional root for region 1 (A) is also the CIST root. The CIST regional root for region 2 (B) and the CIST regional root for region 3 (C) are the roots for their respective subtrees within the CIST.

Figure 20-1 MST Regions, CIST Regional Roots, and CST Root



Only the CST instance sends and receives BPDUs, and MST instances add their spanning tree information into the BPDUs to interact with neighboring switches and compute the final spanning tree topology. Because of this, the spanning tree parameters related to BPDU transmission (for example, hello time, forward time, max-age, and max-hops) are configured only on the CST instance but affect all MST instances. Parameters related to the spanning tree topology (for example, switch priority, port VLAN cost, and port VLAN priority) can be configured on both the CST instance and the MST instance.

MST switches use Version 3 BPDUs or 802.1D STP BPDUs to communicate with 802.1D switches. MST switches use MST BPDUs to communicate with MST switches.

IEEE 802.1s Terminology

Some MST naming conventions used in the prestandard implementation have been changed to include identification of some *internal* and *regional* parameters. These parameters are used only within an MST region, compared to external parameters that are used throughout the whole network. Because the CIST is the only spanning tree instance that spans the whole network, only the CIST parameters require the external qualifiers and not the internal or regional qualifiers.

- The CIST root is the root bridge for the the CIST, which is the unique instance that spans the whole network.
- The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. Remember that an MST region looks like a single switch to the CIST. The CIST external root path cost is the root path cost calculated between these virtual switches and switches that do not belong to any region.

- The CIST regional root was called the IST master in the prestandard implementation. If the CIST root is in the region, the CIST regional root is the CIST root. Otherwise, the CIST regional root is the closest switch to the CIST root in the region. The CIST regional root acts as a root bridge for the IST.
- The CIST internal root path cost is the cost to the CIST regional root in a region. This cost is only relevant to the IST, instance 0.

Table 20-1 compares the IEEE standard and the Cisco prestandard terminology.

Table 20-1 Prestandard and Standard Terminology

IEEE Standard Definition	Cisco Prestandard Implementation	Cisco Standard Implementation
CIST regional root	IST master	CIST regional root
CIST internal root path cost	IST master path cost	CIST internal path cost
CIST external root path cost	Root path cost	Root path cost
MSTI regional root	Instance root	Instance root
MSTI internal root path cost	Root path cost	Root path cost

Hop Count

MST does not use the message-age and maximum-age information in the configuration BPDU to compute the spanning tree topology. Instead, they use the path cost to the root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism.

By using the **spanning-tree mst max-hops** global configuration command, you can configure the maximum hops inside the region and apply it to the IST and all MST instances in that region. The hop count achieves the same result as the message-age information (triggers a reconfiguration). The root bridge of the instance always sends a BPDU (or M-record) with a cost of 0 and the hop count set to the maximum value. When a switch receives this BPDU, it decrements the received remaining hop count by one and propagates this value as the remaining hop count in the BPDUs it generates. When the count reaches zero, the switch discards the BPDU and ages the information held for the port.

The message-age and maximum-age information in the RSTP portion of the BPDU remain the same throughout the region, and the same values are propagated by the region-designated ports at the boundary.

Boundary Ports

In the Cisco prestandard implementation, a boundary port connects an MST region to one of these STP regions:

- A single spanning tree region running RSTP
- A single spanning tree region running PVST+ or rapid PVST+
- Another MST region with a different MST configuration

A boundary port also connects to a LAN, the designated switch of which is either a single spanning tree switch or a switch with a different MST configuration.

There is no definition of a boundary port in the 802.1s standard. The 802.1Q-2002 standard identifies two kinds of messages that a port can receive: internal (coming from the same region) and external. When a message is external, it is received only by the CIST. If the CIST role is root or alternate, or if

the external BPDU is a topology change, it could have an impact on the MST instances. When a message is internal, the CIST part is received by the CIST, and each MST instance receives its respective M-record. The Cisco prestandard implementation treats a port that receives an external message as a boundary port, which means a port cannot receive a mix of internal and external messages.

An MST region includes both switches and LANs. A segment belongs to the region of its designated port. Therefore, a port in a different region from the designated port for a segment is a boundary port. This definition allows two ports internal to a region to share a segment with a port belonging to a different region, creating the possibility of receiving both internal and external messages on a port.

The primary change from the Cisco prestandard implementation is that a designated port is not defined as boundary unless it is running in an STP-compatible mode.

**Note**

If there is an 802.1D STP switch on the segment, messages are always considered external.

The other change from the prestandard implementation is that the CIST regional root bridge ID field is now inserted where an RSTP or legacy 802.1s switch has the sender switch ID. The whole region performs like a single virtual switch by sending a consistent sender switch ID to neighboring switches. In this example, switch C would receive a BPDU with the same consistent sender switch ID of root, whether or not A or B is designated for the segment.

Standard-Compliant MST Implementation

The standard-compliant MST implementation includes features required to meet the standard, as well as some of the desirable prestandard functionality that is not yet incorporated into the published standard. These sections describe the standard-compliant MST implementation:

- [Changes in Port-Role Naming, page 20-7](#)
- [Spanning Tree Interoperation Between Legacy and Standard-Compliant Switches, page 20-8](#)
- [Detecting Unidirectional Link Failure, page 20-8](#)

Changes in Port-Role Naming

The boundary role was deleted from the final MST standard, but this boundary concept is maintained in the standard-compliant implementation. However, an MST instance (MSTI) port at a boundary of the region might not follow the state of the corresponding CIST port. The following two situations currently exist:

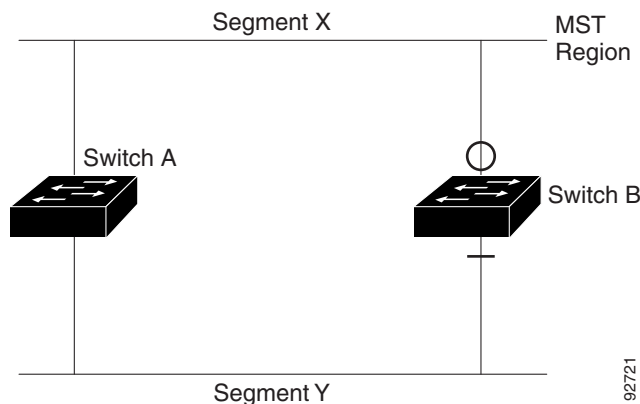
- The boundary port is the root port of the CIST regional root—When the CIST instance port is proposed and is synchronized, it can send back an agreement and move to the forwarding state only after all the corresponding MSTI ports are synchronized (and thus forwarding). The MSTI ports now have a special *master* role.
- The boundary port is not the root port of the CIST regional root—The MSTI ports follow the state and role of the CIST port. The standard provides less information, and it might be difficult to understand why an MSTI port can be alternately blocking when it receives no BPDUs (M-records). In this situation, although the boundary role no longer exists, when you enter the **show** commands, they identify a port as boundary in the *type* column of the output.

Spanning Tree Interoperation Between Legacy and Standard-Compliant Switches

Because automatic detection of prestandard switches can fail, you can use an interface configuration command to identify prestandard ports. A region cannot be formed between a standard and a prestandard switch, but they can interoperate before using the CIST. Only the capability of load balancing over different instances is lost in this specific situation. The CLI displays different flags depending on the port configuration when the port receives prestandard BPDUs. A syslog message also appears the first time a switch receives a prestandard BPDU on a port that has not been configured for prestandard BPDU transmission.

Figure 20-2 illustrates a standard-compliant switch connected to a prestandard switch. Assume that A is the standard-compliant switch and B is a prestandard switch, both configured to be in the same region. A is the root bridge for the CIST, and so B has a root port (BX) on segment X and an alternate port (BY) on segment Y. If segment Y flaps, and the port on BY becomes the alternate before sending out a single prestandard BPDU, AY cannot detect that a prestandard switch is connected to Y and continues to send standard BPDUs. The port BY is fixed in a boundary, and no load balancing is possible between A and B. The same problem exists on segment X, but B might transmit topology changes.

Figure 20-2 Standard-Compliant and Prestandard Switch Interoperation



Note

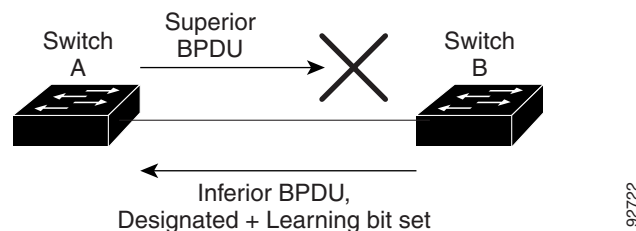
We recommend that you minimize the interaction between standard and prestandard MST implementations.

Detecting Unidirectional Link Failure

This feature is not yet present in the IEEE MST standard, but it is included in the standard-compliant implementation. 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 20-3 illustrates a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to switch B. RSTP and MST BPDUs include the role and state of the sending port. With this information, switch A can detect that switch B does not react to the superior BPDUs it sends and that switch B is the designated, not root bridge. As a result, switch A blocks (or keeps blocking) its port, thus preventing the bridging loop.

Figure 20-3 Detecting Unidirectional Link Failure

Interoperability with IEEE 802.1D-1998 STP

A switch running MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D switches. If this switch receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MST switch also can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an RSTP BPDU (Version 2).

However, the switch does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D switch has been removed from the link unless the 802.1D switch is the designated switch. A switch might also continue to assign a boundary role to a port when the switch to which this switch is connected has joined the region. To restart the protocol migration process (force the renegotiation with neighboring switches), use the **clear spanning-tree detected-protocols** privileged EXEC command.

If all the 802.1D switches on the link are RSTP switches, they can process MST BPDUs as if they are RSTP BPDUs. Therefore, MST switches send either a Version 0 configuration and topology change notification (TCN) BPDUs or Version 3 MST BPDUs on a boundary port. A boundary port connects to a LAN, the designated switch of which is either a single spanning tree switch or a switch with a different MST configuration.

Understanding RSTP

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).

This section describes how the RSTP works:

- [Port Roles and the Active Topology, page 20-10](#)
- [Rapid Convergence, page 20-11](#)
- [Synchronization of Port Roles, page 20-12](#)
- [Bridge Protocol Data Unit Format and Processing, page 20-13](#)

Port Roles and the Active Topology

The RSTP provides rapid convergence of the spanning tree by assigning port roles and by learning the active topology. The RSTP builds upon the 802.1D STP to select the switch with the highest switch priority (lowest numerical priority value) as the root bridge as described in the [“Election of the Root Bridge”](#) section on page 21-4. The RSTP then assigns one of these port roles to individual ports:

- Root port—Provides the best path (lowest cost) when the switch forwards packets to the root bridge.
- Designated port—Connects to the designated switch, which incurs the lowest path cost when forwarding packets from that LAN to the root bridge. The port through which the designated switch is attached to the LAN is called the designated port.
- Alternate port—Offers an alternate path toward the root bridge to that provided by the current root port.
- Backup port—Acts as a backup for the path provided by a designated port toward the leaves of the spanning tree. A backup port can exist only when two ports are connected in a loopback by a point-to-point link or when a switch has two or more connections to a shared LAN segment.
- Disabled port—Has no role within the operation of the spanning tree.

A port with the root or a designated port role is included in the active topology. A port with the alternate or backup port role is excluded from the active topology.

In a stable topology with consistent port roles throughout the network, the RSTP ensures that every root port and designated port immediately transition to the forwarding state while all alternate and backup ports are always in the discarding state (equivalent to blocking in 802.1D). The port state controls the operation of the forwarding and learning processes. [Table 20-2](#) provides a comparison of 802.1D and RSTP port states.

Table 20-2 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 switch, a switch port, or a LAN. It provides rapid convergence for edge ports, new root ports, and ports connected through point-to-point links as follows:

- **Edge ports**—If you configure a port as an edge port on an RSTP switch by using the **spanning-tree portfast** interface configuration command, the edge port immediately transitions to the forwarding state. An edge port is the same as a Port Fast-enabled port, and you should enable it only on ports that connect to a single end station.
- **Root ports**—If the RSTP selects a new root port, it blocks the old root port and immediately transitions the new root port to the forwarding state.
- **Point-to-point links**—If you connect a port to another port through a point-to-point link and the local port becomes a designated port, it negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology.

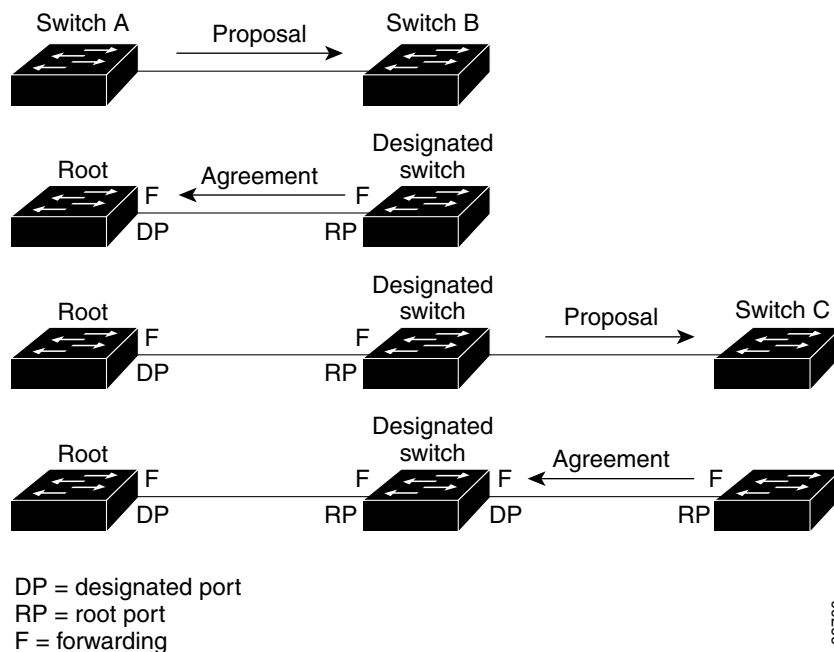
As shown in [Figure 20-4](#), switch A is connected to switch B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of switch A is a smaller numerical value than the priority of switch B. Switch A sends a proposal message (a configuration BPDU with the proposal flag set) to switch B, proposing itself as the designated switch.

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

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

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

The switch learns the link type from the port duplex mode: a full-duplex port is considered to have a point-to-point connection and a half-duplex port is considered to have a shared connection. You can override the default setting that is controlled by the duplex setting by using the **spanning-tree link-type** interface configuration command.

Figure 20-4 Proposal and Agreement Handshaking for Rapid Convergence

88760

Synchronization of Port Roles

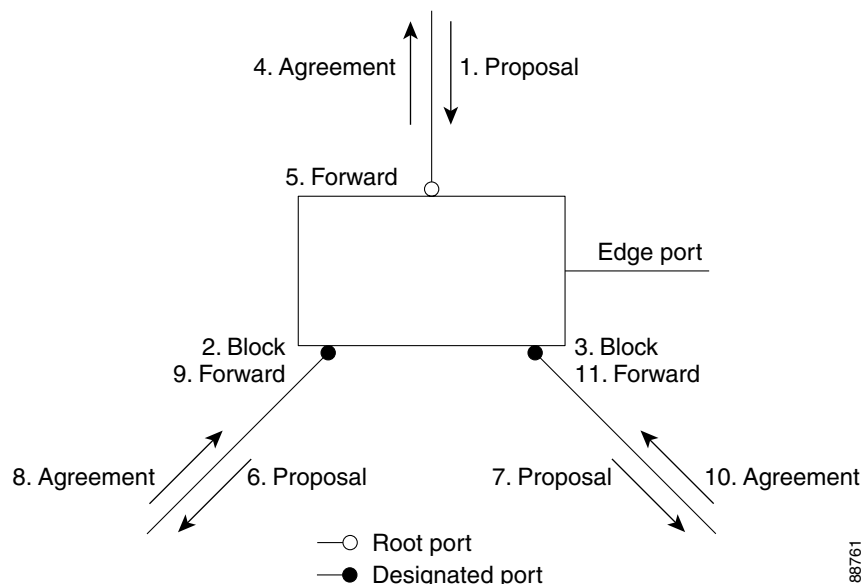
When the switch receives a proposal message on one of its ports and that port is selected as the new root port, the RSTP forces all other ports to synchronize with the new root information.

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

- That port is in the blocking state.
- It is an edge port (a port configured to be at the edge of the network).

If a designated port is in the forwarding state and is not configured as an edge port, it transitions to the blocking state when the RSTP forces it to synchronize with new root information. In general, when the RSTP forces a port to synchronize with root information and the port does not satisfy any of the above conditions, its port state is set to blocking.

After ensuring that all of the ports are synchronized, the switch sends an agreement message to the designated switch corresponding to its root port. When the switches connected by a point-to-point link are in agreement about their port roles, the RSTP immediately transitions the port states to forwarding. The sequence of events is shown in [Figure 20-5](#).

Figure 20-5 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 20-13](#)
- [Processing Superior BPDU Information, page 20-14](#)
- [Processing Inferior BPDU Information, page 20-14](#)

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 20-3](#) describes the RSTP flag fields.

Table 20-3 RSTP BPDU Flags

Bit	Function
0	Topology change (TC)
1	Proposal
2–3:	Port role:
00	Unknown
01	Alternate port or backup port
10	Root port
11	Designated port
4	Learning
5	Forwarding

Table 20-3 RSTP BPDUs Flags (continued)

Bit	Function
6	Agreement
7	Topology change acknowledgement (TCA)

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

The sending switch sets the agreement flag in the RSTP BPDUs 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 BPDUs. It uses the topology change (TC) flag to show the topology changes. However, for interoperability with 802.1D switches, the RSTP switch processes and generates TCN BPDUs.

The learning and forwarding flags are set according to the state of the sending port.

Processing Superior BPDUs Information

A superior BPDUs is a BPDUs 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 BPDUs, 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 BPDUs received is an RSTP BPDUs with the proposal flag set, the switch sends an agreement message after all of the other ports are synchronized. If the BPDUs is an 802.1D BPDUs, the switch does not set the proposal flag and starts the forward-delay timer for the port. The new root port requires twice the forward-delay time to transition to the forwarding state.

If the superior information received on the port causes the port to become a backup port or an alternate port, RSTP sets the port to the blocking state and sends an agreement message. The designated port continues sending BPDUs with the proposal flag set until the forward-delay timer expires, at which time the port transitions to the forwarding state.

Processing Inferior BPDUs Information

An inferior BPDUs is a BPDUs 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 BPDUs, it immediately replies with its own information.

Topology Changes

These are the differences between the RSTP and the 802.1D in handling spanning tree topology changes:

- **Detection**—Unlike 802.1D in which *any* transition between the blocking and the forwarding state causes a topology change, *only* transitions from the blocking to the forwarding state cause a topology change with RSTP (only an increase in connectivity is considered a topology change). State changes on an edge port do not cause a topology change. When an RSTP switch detects a topology change, it deletes the learned information on all of its nonedge ports except on those from which it received the TC notification.
- **Notification**—The RSTP does not use TCN BPDUs, unlike 802.1D. However, for 802.1D interoperability, an RSTP switch processes and generates TCN BPDUs.
- **Acknowledgement**—When an RSTP switch receives a TCN message on a designated port from an 802.1D switch, it replies with an 802.1D configuration BPDU with the TCA bit set. However, if the TC-while timer (the same as the TC timer in 802.1D) is active on a root port connected to an 802.1D switch and a configuration BPDU with the TCA set is received, the TC-while timer is reset.

This method of operation is only required to support 802.1D switches. The RSTP BPDUs never have the TCA bit set.

- **Propagation**—When an RSTP switch receives a TC message from another switch through a designated or root port, it propagates the change to all of its nonedge, designated ports and to the root port (excluding the port on which it is received). The switch starts the TC-while timer for all such ports and flushes the information learned on them.
- **Protocol migration**—For backward compatibility with 802.1D switches, RSTP selectively sends 802.1D configuration BPDUs and TCN BPDUs on a per-port basis.

When a port is initialized, the migrate-delay timer is started (specifies the minimum time during which RSTP BPDUs are sent), and RSTP BPDUs are sent. While this timer is active, the switch processes all BPDUs received on that port and ignores the protocol type.

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

Configuring MST

These sections describe how to configure MST:

- [Default MST Configuration, page 20-16](#)
- [MST Configuration Guidelines and Restrictions, page 20-16](#)
- [Specifying the MST Region Configuration and Enabling MST, page 20-17](#) (required)
- [Configuring the Root Bridge, page 20-19](#) (optional)
- [Configuring a Secondary Root Bridge, page 20-20](#) (optional)
- [Configuring Port Priority, page 20-21](#) (optional)
- [Configuring Path Cost, page 20-22](#) (optional)
- [Configuring the Switch Priority, page 20-23](#) (optional)
- [Configuring the Hello Time, page 20-24](#) (optional)

- [Configuring the Transmit Hold Count, page 20-25](#) (optional)
- [Configuring the Maximum-Aging Time, page 20-26](#) (optional)
- [Configuring the Maximum-Hop Count, page 20-26](#) (optional)
- [Specifying the Link Type to Ensure Rapid Transitions, page 20-26](#) (optional)
- [Designating the Neighbor Type, page 20-27](#) (optional)
- [Restarting the Protocol Migration Process, page 20-28](#) (optional)

Default MST Configuration

Table 20-4 shows the default MST configuration.

Table 20-4 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 switch within the MST region through the command-line interface (CLI) or SNMP.
- For load balancing across redundant paths in the network to work, all VLAN-to-instance mapping assignments must match; otherwise, all traffic flows on a single link.
- All MST boundary ports must be forwarding for load balancing between a PVST+ and an MST cloud or between a rapid-PVST+ and an MST cloud. For this to occur, the CIST regional root of the MST cloud must be the root of the CST. If the MST cloud consists of multiple MST regions, one of the MST regions must contain the CST root, and all of the other MST regions must have a better path to the root contained within the MST cloud than a path through the PVST+ or rapid-PVST+ cloud.

- Partitioning the network into a large number of regions is not recommended. However, if this situation is unavoidable, we recommend that you partition the switched LAN into smaller LANs interconnected by non-Layer 2 devices.


Specifying the MST Region Configuration and Enabling MST

For two or more switches to be in the same MST region, they must have the same VLAN-to-instance mapping, the same configuration revision number, and the same MST name.

A region can have one member or multiple members with the same MST configuration; each member must be capable of processing RSTP BPDUs. There is no limit to the number of MST regions in a network, but each region can only support up to 65 spanning tree instances. You can assign a VLAN to only one spanning tree instance at a time.

To specify the MST region configuration and enable MST, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# spanning-tree mst configuration	Enters MST configuration mode.
Step 3	Router(config-mst)# instance <i>instance_id</i> vlan <i>vlan_range</i>	Maps VLANs to an MST instance. <ul style="list-style-type: none"> • For <i>instance_id</i>, the range is 0 to 4094. • For vlan <i>vlan_range</i>, 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 <i>instance_name</i>	Specifies the instance name. The <i>name</i> string has a maximum length of 32 characters and is case sensitive.
Step 5	Router(config-mst)# revision <i>version</i>	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.

	Command	Purpose
Step 8	Router(config)# spanning-tree mode mst	Enables MST and RSTP. <div>  <div> <div>Caution</div> <div>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.</div> </div> </div> <p>You cannot run both MST and PVST+ or both MST and rapid PVST+ at the same time.</p>
Step 9	Router(config)# end	Returns to privileged EXEC mode.
Step 10	Router# show running-config	Verifies your entries.
Step 11	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return to defaults, do the following:

- To return to the default MST region configuration, use the **no spanning-tree mst configuration** global configuration command.
- To return to the default VLAN-to-instance map, use the **no instance *instance_id* [vlan *vlan_range*]** MST configuration command.
- To return to the default name, use the **no name** MST configuration command.
- To return to the default revision number, use the **no revision** MST configuration command.
- To reenable PVST+, use the **no spanning-tree mode** or the **spanning-tree mode pvst** global configuration command.

This example shows how to enter MST configuration mode, map VLANs 10 to 20 to MST instance 1, name the region *region1*, set the configuration revision to 1, display the pending configuration, apply the changes, and return to global configuration mode:

```

Router(config)# spanning-tree mst configuration
Router(config-mst)# instance 1 vlan 10-20
Router(config-mst)# name region1
Router(config-mst)# revision 1
Router(config-mst)# show pending
Pending MST configuration
Name      [region1]
Revision  1
Instances configured 2
Instance  Vlan  Mapped
-----  -
0         1-9,21-4094
1         10-20
-----

Router(config-mst)# exit
Router(config)#
    
```

Configuring the Root Bridge

The switch maintains a spanning tree instance for the group of VLANs mapped to it. A switch ID, consisting of the switch priority and the switch MAC address, is associated with each instance. For a group of VLANs, the switch with the lowest switch ID becomes the root bridge.

To configure a switch to become the root bridge, use the **spanning-tree mst instance_id root** global configuration command to modify the switch priority from the default value (32768) to a significantly lower value so that the switch becomes the root bridge for the specified spanning tree instance. When you enter this command, the switch checks the switch priorities of the root bridges. Because of the extended system ID support, the switch sets its own priority for the specified instance to 24576 if this value will cause this switch to become the root bridge for the specified spanning tree instance.

If any root bridge for the specified instance has a switch priority lower than 24576, the switch sets its own priority to 4096 less than the lowest switch priority. (4096 is the value of the least-significant bit of a 4-bit switch priority value as shown in [Table 21-2 on page 21-3](#).)

If your network consists of switches that both do and do not support the extended system ID, it is unlikely that the switch with the extended system ID support will become the root bridge. The extended system ID increases the switch priority value every time the VLAN number is greater than the priority of the connected switches running older software.

The root bridge for each spanning tree instance should be a backbone or distribution switch. Do not configure an access switch as the spanning tree primary root bridge.

Use the **diameter** keyword, which is available only for MST instance 0, to specify the Layer 2 network diameter (that is, the maximum number of Layer 2 hops between any two end stations in the Layer 2 network). When you specify the network diameter, the switch automatically sets an optimal hello time, forward-delay time, and maximum-age time for a network of that diameter, which can significantly reduce the convergence time. You can use the **hello** keyword to override the automatically calculated hello time.



Note

With the switch configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time with the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

To configure a switch as the root bridge, perform this task:

	Command	Purpose
Step 1	Router(config)# configure terminal	Enters global configuration mode.
Step 2	Router(config-config)# spanning-tree mst <i>instance_id</i> root primary [diameter <i>net_diameter</i> [hello-time <i>seconds</i>]]	(Optional) Configures a switch as the root bridge. <ul style="list-style-type: none"> For <i>instance_id</i>, 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 <i>net_diameter</i>, 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 <i>seconds</i>, 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 <i>instance_id</i>	Verifies your entries.
Step 5	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst *instance_id* root** global configuration command.

Configuring a Secondary Root Bridge

When you configure a switch with the extended system ID support as the secondary root, the switch priority is modified from the default value (32768) to 28672. The switch is then likely to become the root bridge for the specified instance if the primary root bridge fails. This is assuming that the other network switches use the default switch priority of 32768 and therefore are unlikely to become the root bridge.

You can execute this command on more than one switch to configure multiple backup root bridges. Use the same network diameter and hello-time values that you used when you configured the primary root bridge with the **spanning-tree mst *instance_id* root primary** global configuration command.

To configure a switch as the secondary root bridge, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# spanning-tree mst <i>instance_id</i> root secondary [diameter <i>net_diameter</i> [hello-time <i>seconds</i>]]	(Optional) Configures a switch as the secondary root bridge. <ul style="list-style-type: none"> For <i>instance_id</i>, 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 <i>net_diameter</i>, specify the maximum number of switches between any two end stations. The range is 2 to 7. This keyword is available only for MST instance 0. (Optional) For hello-time <i>seconds</i>, 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 20-19.
Step 3	Router(config)# end	Returns to privileged EXEC mode.
Step 4	Router# show spanning-tree mst <i>instance_id</i>	Verifies your entries.
Step 5	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst** *instance_id* **root** global configuration command.

Configuring Port Priority

If a loop occurs, MST uses the port priority when selecting an interface to put into the forwarding state. You can assign higher priority values (lower numerical values) to interfaces that you want selected first and lower priority values (higher numerical values) that you want selected last. If all interfaces have the same priority value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

To configure the MST port priority of an interface, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# interface {{ <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }}	(Optional) Specifies an interface to configure, and enters interface configuration mode.

	Command	Purpose
Step 3	Router(config-if)# spanning-tree mst <i>instance_id</i> port-priority <i>priority</i>	Configures the port priority. <ul style="list-style-type: none"> For <i>instance_id</i>, 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 <i>priority</i>, the range is 0 to 240 in increments of 16. The default is 128. The lower the number, the higher the priority. <p>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.</p>
Step 4	Router(config-if)# end	Returns to privileged EXEC mode.
Step 5	Router# show spanning-tree mst interface <i>interface_id</i> or Router# show spanning-tree mst <i>instance_id</i>	Verifies your entries.
Step 6	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

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



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 {{ <i>type</i> ¹ <i>slot/port</i> } port-channel <i>number</i> }}	(Optional) Specifies an interface to configure, and enters interface configuration mode.

	Command	Purpose
Step 3	Router(config-if)# spanning-tree mst <i>instance_id</i> cost <i>cost</i>	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. <ul style="list-style-type: none"> For <i>instance_id</i>, 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 <i>cost</i>, 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 <i>interface_id</i> or Router# show spanning-tree mst <i>instance_id</i>	Verifies your entries.
Step 6	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

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



Note

The **show spanning-tree mst interface** *interface_id* privileged EXEC command displays information only for ports that are in a link-up operative state. Otherwise, you can use the **show running-config** privileged EXEC command to confirm the configuration.

To return the interface to its default setting, use the **no spanning-tree mst** *instance_id* **cost** interface configuration command.

Configuring the Switch Priority

You can configure the switch priority so that it is more likely that a switch is chosen as the root bridge.



Note

Exercise care when using this command. For most situations, we recommend that you use the **spanning-tree mst** *instance_id* **root primary** and the **spanning-tree mst** *instance_id* **root secondary** global configuration commands to modify the switch priority.

To configure the switch priority, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# spanning-tree mst <i>instance_id</i> priority <i>priority</i>	(Optional) Configures the switch priority. <ul style="list-style-type: none"> For <i>instance_id</i>, 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 <i>priority</i>, the range is 0 to 61440 in increments of 4096; the default is 32768. The lower the number, the more likely the switch will be chosen as the root bridge. <p>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.</p>
Step 3	Router(config)# end	Returns to privileged EXEC mode.
Step 4	Router# show spanning-tree mst <i>instance_id</i>	Verifies your entries.
Step 5	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst *instance_id* priority** global configuration command.

Configuring the Hello Time

You can configure the interval between the generation of configuration messages by the root bridge by changing the hello time.



Note

Exercise care when using this command. For most situations, we recommend that you use the **spanning-tree mst *instance_id* root primary** and the **spanning-tree mst *instance_id* root secondary** global configuration commands to modify the hello time.

To configure the hello time for all MST instances, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# spanning-tree mst hello-time <i>seconds</i>	(Optional) Configures the hello time for all MST instances. The hello time is the interval between the generation of configuration messages by the root bridge. These messages mean that the switch is alive. For <i>seconds</i> , the range is 1 to 10; the default is 2.
Step 3	end	Returns to privileged EXEC mode.

	Command	Purpose
Step 4	Router# show spanning-tree mst	Verifies your entries.
Step 5	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst hello-time** global configuration command.

Configuring the Forwarding-Delay Time

To configure the forwarding-delay time for all MST instances, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# spanning-tree mst forward-time <i>seconds</i>	(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 <i>seconds</i> , the range is 4 to 30; the default is 15.
Step 3	Router(config)# end	Returns to privileged EXEC mode.
Step 4	Router# show spanning-tree mst	Verifies your entries.
Step 5	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst forward-time** global configuration command.

Configuring the Transmit Hold Count

To configure the transmit hold count for all MST instances, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# spanning-tree transmit hold-count <i>hold_count_value</i>	Configures the transmit hold count for all MST instances. For <i>hold_count_value</i> , the range is 1 to 20; the default is 6.
Step 3	Router(config)# end	Returns to privileged EXEC mode.
Step 4	Router# show spanning-tree mst	Verifies your entries.
Step 5	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree transmit hold-count** global configuration command.

Configuring the Maximum-Aging Time

To configure the maximum-aging time for all MST instances, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# spanning-tree mst max-age <i>seconds</i>	(Optional) Configures the maximum-aging time for all MST instances. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration. For <i>seconds</i> , the range is 6 to 40; the default is 20.
Step 3	Router(config)# end	Returns to privileged EXEC mode.
Step 4	Router# show spanning-tree mst	Verifies your entries.
Step 5	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst max-age** global configuration command.

Configuring the Maximum-Hop Count

To configure the maximum-hop count for all MST instances, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# spanning-tree mst max-hops <i>hop_count</i>	(Optional) Specifies the number of hops in a region before the BPDU is discarded, and the information held for a port is aged. For <i>hop_count</i> , the range is 1 to 255; the default is 20.
Step 3	Router(config)# end	Returns to privileged EXEC mode.
Step 4	Router# show spanning-tree mst	Verifies your entries.
Step 5	Router# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the switch to its default setting, use the **no spanning-tree mst max-hops** global configuration command.

Specifying the Link Type to Ensure Rapid Transitions

If you connect a port to another port through a point-to-point link and the local port becomes a designated port, the RSTP negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology as described in the [“Rapid Convergence” section on page 20-11](#).

By default, the link type is controlled from the duplex mode of the interface: a full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection. If you have a half-duplex link physically connected point-to-point to a single port on a remote switch running MST, you can override the default setting of the link type and enable rapid transitions to the forwarding state.

To override the default link-type setting, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# interface {{type ¹ slot/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.

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

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 {{type ¹ slot/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.

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

To return the port to its default setting, use the **no spanning-tree mst prestandard** interface configuration command.

Restarting the Protocol Migration Process

A switch running MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D switches. If this switch receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MST switch also can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an RST BPDU (Version 2).

However, the switch does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D switch has been removed from the link unless the 802.1D switch is the designated switch. A switch also might continue to assign a boundary role to a port when the switch to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches) on the switch, use the **clear spanning-tree detected-protocols** privileged EXEC command.

To restart the protocol migration process on a specific interface, use the **clear spanning-tree detected-protocols interface *interface_id*** privileged EXEC command.

Displaying the MST Configuration and Status

To display the spanning-tree status, use one or more of the privileged EXEC commands that are described in [Table 20-5](#).

Table 20-5 *Commands for Displaying MST Status*

Command	Purpose
<code>show spanning-tree mst configuration</code>	Displays the MST region configuration.
<code>show spanning-tree mst configuration digest</code>	Displays the MD5 digest included in the current MSTCI.
<code>show spanning-tree mst <i>instance_id</i></code>	Displays MST information for the specified instance.
<code>show spanning-tree mst interface <i>interface_id</i></code>	Displays MST information for the specified interface.



Configuring STP and Prestandard IEEE 802.1s MST

This chapter describes how to configure the Spanning Tree Protocol (STP) and prestandard IEEE 802.1s Multiple Spanning Tree (MST) protocol on Catalyst 6500 series switches.



Note

- The IEEE 802.1s MST protocol has transitioned from a prestandard state to a released state. [Chapter 20, “Configuring Standard-Compliant IEEE MST,”](#) describes the standard-compliant MST implementation supported in Release 12.2(18)SXF and later releases. This chapter describes the prestandard MST implementation supported in releases earlier than Release 12.2(18)SXF.
- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL: <http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How STP Works, page 21-1](#)
- [Understanding How IEEE 802.1w RSTP Works, page 21-13](#)
- [Understanding How Prestandard IEEE 802.1s MST Works, page 21-14](#)
- [Default STP Configuration, page 21-20](#)
- [STP and MST Configuration Guidelines and Restrictions, page 21-21](#)
- [Configuring STP, page 21-21](#)
- [Configuring Prestandard IEEE 802.1s MST, page 21-33](#)



Note

For information on configuring the PortFast, UplinkFast, and BackboneFast STP enhancements, see [Chapter 22, “Configuring Optional STP Features.”](#)

Understanding How STP Works

These sections describe how STP works:

- [STP Overview, page 21-2](#)
- [Understanding the Bridge ID, page 21-2](#)
- [Understanding Bridge Protocol Data Units, page 21-4](#)

- [Election of the Root Bridge, page 21-4](#)
- [STP Protocol Timers, page 21-5](#)
- [Creating the Spanning Tree Topology, page 21-5](#)
- [STP Port States, page 21-6](#)
- [STP and IEEE 802.1Q Trunks, page 21-12](#)

STP Overview

STP is a Layer 2 link-management protocol that provides path redundancy while preventing undesirable loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. STP operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

Catalyst 6500 series switches use STP (the IEEE 802.1D bridge protocol) on all VLANs. By default, a single instance of STP runs on each configured VLAN (provided you do not manually disable STP). You can enable and disable STP on a per-VLAN basis.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The STP algorithm calculates the best loop-free path throughout a switched Layer 2 network. Layer 2 LAN ports send and receive STP frames at regular intervals. Network devices do not forward these frames, but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and network devices might learn end station MAC addresses on multiple Layer 2 LAN ports. These conditions result in an unstable network.

STP defines a tree with a root bridge and a loop-free path from the root to all network devices in the Layer 2 network. STP forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the STP algorithm recalculates the spanning tree topology and activates the standby path.

When two Layer 2 LAN ports on a network device are part of a loop, the STP port priority and port path cost setting determine which port is put in the forwarding state and which port is put in the blocking state. The STP port priority value represents the location of a port in the network topology and how efficiently that location allows the port to pass traffic. The STP port path cost value represents media speed.

Understanding the Bridge ID

Each VLAN on each network device has a unique 64-bit bridge ID consisting of a bridge priority value, an extended system ID, and an STP MAC address allocation.

This section contains these topics:

- [Bridge Priority Value, page 21-3](#)
- [Extended System ID, page 21-3](#)
- [STP MAC Address Allocation, page 21-3](#)

Bridge Priority Value

The bridge priority is a 4-bit value when the extended system ID is enabled (see [Table 21-2 on page 21-3](#) and the “[Configuring the Bridge Priority of a VLAN](#)” section on page 21-29).

Extended System ID

A 12-bit extended system ID field is part of the bridge ID (see [Table 21-2 on page 21-3](#)). Chassis that support only 64 MAC addresses always use the 12-bit extended system ID. On chassis that support 1024 MAC addresses, you can enable use of the extended system ID. STP uses the VLAN ID as the extended system ID. See the “[Enabling the Extended System ID](#)” section on page 21-23.

Table 21-1 Bridge Priority Value with the Extended System ID Disabled

Bridge Priority Value															
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

Table 21-2 Bridge Priority Value and Extended System ID with the Extended System ID Enabled

Bridge Priority Value				Extended System ID (Set Equal to the VLAN ID)											
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

STP MAC Address Allocation

Catalyst 6500 series switch chassis have either 64 or 1024 MAC addresses available to support software features such as STP. To view the MAC address range on your chassis, enter the **show catalyst6000 chassis-mac-address** command.

For chassis with 64 MAC addresses, STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

Earlier releases support chassis with 1024 MAC addresses. With earlier releases, STP uses one MAC address per VLAN to make the bridge ID unique for each VLAN.

If you have a network device in your network with MAC address reduction enabled, you should also enable MAC address reduction on all other Layer 2 connected network devices to avoid undesirable root bridge election and spanning tree topology issues.

When MAC address reduction is enabled, the root bridge priority becomes a multiple of 4096 plus the VLAN ID. With MAC address reduction enabled, a switch bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge, the lowest being preferred) can only be specified as a multiple of 4096. Only the following values are possible: 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440.

If another bridge in the same spanning tree domain does not run the MAC address reduction feature, it could win root bridge ownership because of the finer granularity in the selection of its bridge ID.

Understanding Bridge Protocol Data Units

Bridge protocol data units (BPDUs) are transmitted in one direction from the root bridge. Each network device sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the network device that the transmitting network device believes to be the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- Message age
- The identifier of the transmitting port
- Values for the hello, forward delay, and max-age protocol timers

When a network device transmits a BPDU frame, all network devices connected to the LAN on which the frame is transmitted receive the BPDU. When a network device receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU, and, if the topology changes, initiate a BPDU transmission.

A BPDU exchange results in the following:

- One network device is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each network device based on the path cost.
- A designated bridge for each LAN segment is selected. This is the network device closest to the root bridge through which frames are forwarded to the root.
- A root port is selected. This is the port providing the best path from the bridge to the root bridge.
- Ports included in the spanning tree are selected.

Election of the Root Bridge

For each VLAN, the network device with the highest bridge ID (the lowest numerical ID value) is elected as the root bridge. If all network devices are configured with the default priority (32768), the network device with the lowest MAC address in the VLAN becomes the root bridge. The bridge priority value occupies the most significant bits of the bridge ID.

When you change the bridge priority value, you change the probability that the switch will be elected as the root bridge. Configuring a higher value increases the probability; a lower value decreases the probability.

The STP root bridge is the logical center of the spanning tree topology in a Layer 2 network. All paths that are not needed to reach the root bridge from anywhere in the Layer 2 network are placed in STP blocking mode.

BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. STP uses this information to elect the root bridge for the Layer 2 network, to elect the root port leading to the root bridge, and to determine the designated port for each Layer 2 segment.

STP Protocol Timers

Table 21-3 describes the STP protocol timers that affect STP performance.

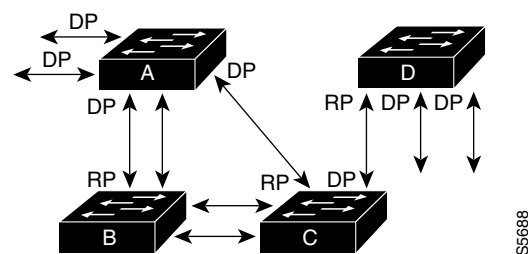
Table 21-3 STP Protocol Timers

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 21-1, 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 21-1 Spanning Tree Topology



RP = Root Port
DP = Designated Port

When the spanning tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change. The goal is to make the fastest link the root port.

For example, assume that one port on Switch B is a fiber-optic link, and another port on Switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the STP port priority on the fiber-optic port to a higher priority (lower numerical value) than the root port, the fiber-optic port becomes the new root port.

STP Port States

These sections describe the STP port states:

- [STP Port State Overview, page 21-6](#)
- [Blocking State, page 21-7](#)
- [Listening State, page 21-8](#)
- [Learning State, page 21-9](#)
- [Forwarding State, page 21-10](#)
- [Disabled State, page 21-11](#)

STP Port State Overview

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a Layer 2 LAN port transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames. They must allow the frame lifetime to expire for frames that have been forwarded using the old topology.

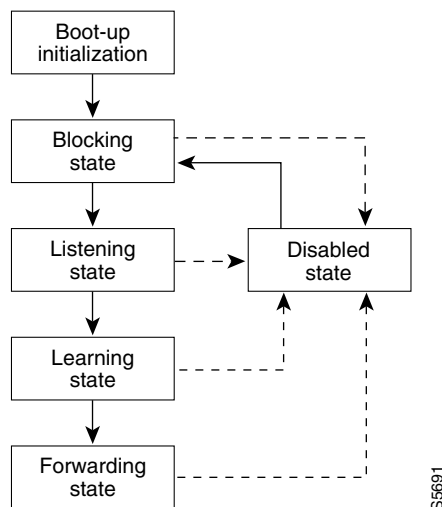
Each Layer 2 LAN port on a Catalyst 6500 series switch using STP exists in one of the following five states:

- **Blocking**—The Layer 2 LAN port does not participate in frame forwarding.
- **Listening**—First transitional state after the blocking state when STP determines that the Layer 2 LAN port should participate in frame forwarding.
- **Learning**—The Layer 2 LAN port prepares to participate in frame forwarding.
- **Forwarding**—The Layer 2 LAN port forwards frames.
- **Disabled**—The Layer 2 LAN port does not participate in STP and is not forwarding frames.

A Layer 2 LAN port moves through these five states as follows:

- From initialization to blocking
- From blocking to listening or to disabled
- From listening to learning or to disabled
- From learning to forwarding or to disabled
- From forwarding to disabled

[Figure 21-2](#) illustrates how a Layer 2 LAN port moves through the five states.

Figure 21-2 STP Layer 2 LAN Interface States

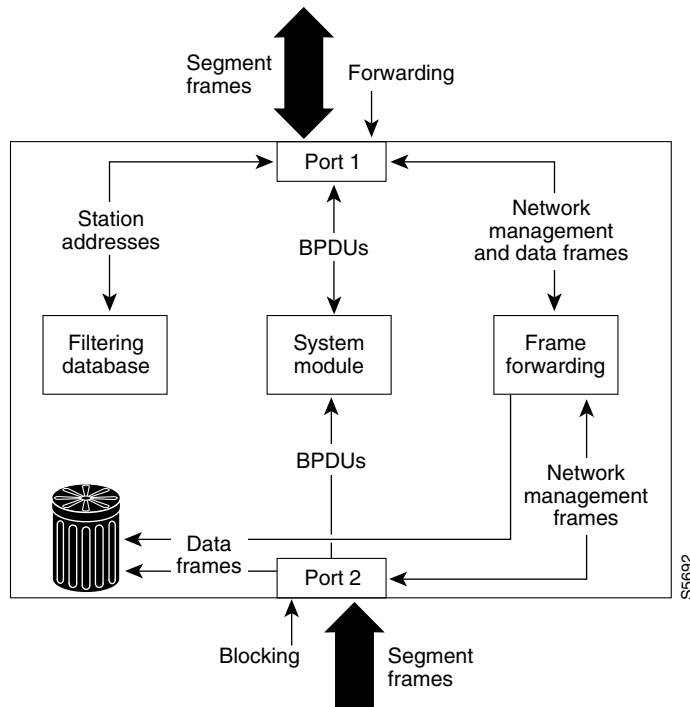
When you enable STP, every port in the Catalyst 6500 series switch, VLAN, and network goes through the blocking state and the transitory states of listening and learning at power up. If properly configured, each Layer 2 LAN port stabilizes to the forwarding or blocking state.

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

1. The Layer 2 LAN port is put into the listening state while it waits for protocol information that suggests it should go to the blocking state.
2. The Layer 2 LAN port waits for the forward delay timer to expire, moves the Layer 2 LAN port to the learning state, and resets the forward delay timer.
3. In the learning state, the Layer 2 LAN port continues to block frame forwarding as it learns end station location information for the forwarding database.
4. The Layer 2 LAN port waits for the forward delay timer to expire and then moves the Layer 2 LAN port to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A Layer 2 LAN port in the blocking state does not participate in frame forwarding, as shown in [Figure 21-3](#). 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.

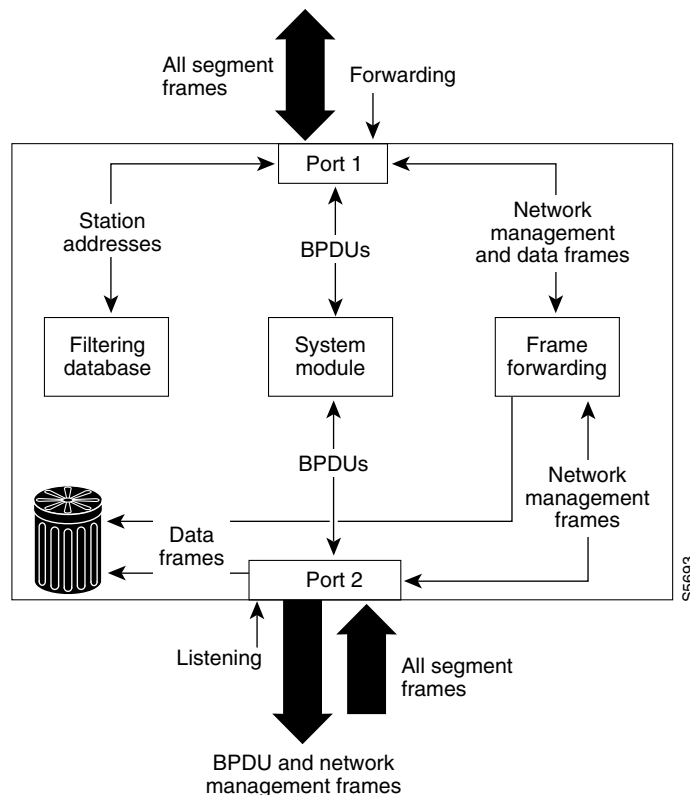
Figure 21-3 Interface 2 in Blocking State

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 21-4](#) shows a Layer 2 LAN port in the listening state.

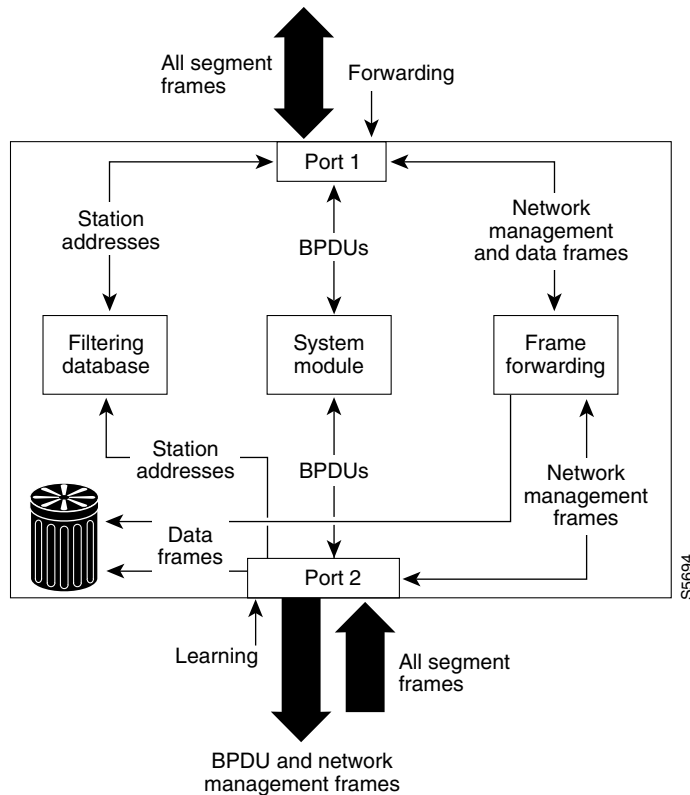
Figure 21-4 Interface 2 in Listening State

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 21-5](#) shows a Layer 2 LAN port in the learning state.

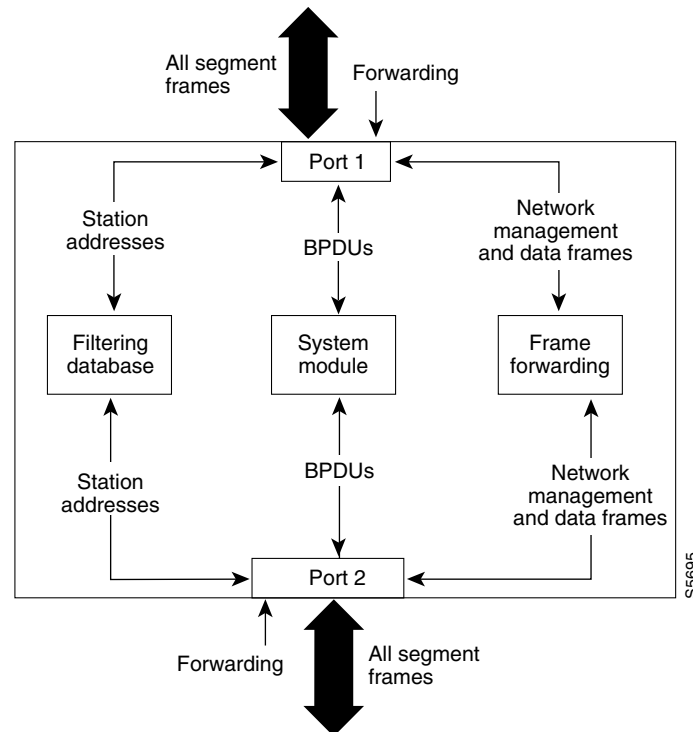
Figure 21-5 Interface 2 in Learning State

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 21-6](#). The Layer 2 LAN port enters the forwarding state from the learning state.

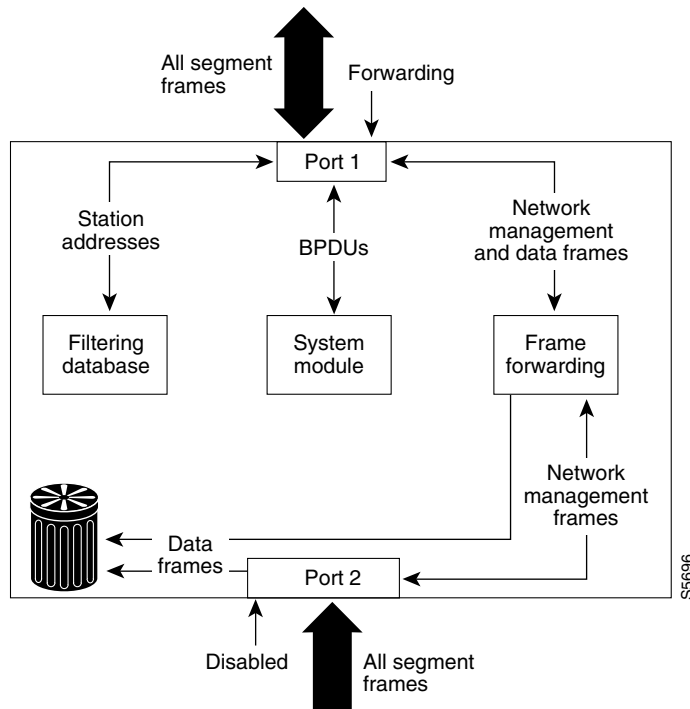
Figure 21-6 Interface 2 in Forwarding State

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 21-7](#). A Layer 2 LAN port in the disabled state is virtually nonoperational.

Figure 21-7 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 11, “Configuring LAN Ports for Layer 2 Switching.”](#)

Understanding How IEEE 802.1w RSTP Works

**Note**

RSTP is available as a standalone protocol in Rapid-Per-VLAN-Spanning Tree (Rapid-PVST) mode. In this mode, the switch runs an RSTP instance on each VLAN, which follows the usual PVST+ approach.

These sections describe Rapid Spanning Tree Protocol (RSTP):

- [IEEE 802.1w RSTP Overview, page 21-13](#)
- [RSTP Port Roles, page 21-13](#)
- [RSTP Port States, page 21-14](#)
- [Rapid-PVST, page 21-14](#)

IEEE 802.1w RSTP Overview

RSTP significantly reduces the time to reconfigure the active topology of the network when changes occur to the physical topology or its configuration parameters. RSTP selects one switch as the root of a spanning tree-connected active topology and assigns port roles to individual ports of the switch, depending on whether that port is part of the active topology.

RSTP provides rapid connectivity following the failure of a switch, switch port, or a LAN. A new root port and the designated port on the other side of the bridge transition to forwarding using an explicit handshake between them. RSTP allows switch port configuration so that the ports can transition to forwarding directly when the switch reinitializes.

RSTP as specified in 802.1w supersedes STP specified in 802.1D, but remains compatible with STP.

RSTP provides backward compatibility with 802.1D bridges as follows:

- RSTP selectively sends 802.1D-configured BPDUs and topology change notification (TCN) BPDUs on a per-port basis.
- When a port initializes, the migration-delay timer starts and RSTP BPDUs are transmitted. While the migration-delay timer is active, the bridge processes all BPDUs received on that port.
- If the bridge receives an 802.1D BPDU after a port's migration-delay timer expires, the bridge assumes it is connected to an 802.1D bridge and starts using only 802.1D BPDUs.
- When RSTP uses 802.1D BPDUs on a port and receives an RSTP BPDU after the migration-delay expires, RSTP restarts the migration-delay timer and begins using RSTP BPDUs on that port.

RSTP Port Roles

RSTP uses the following definitions for port roles:

- **Root**—A forwarding port elected for the spanning tree topology.
- **Designated**—A forwarding port elected for every switched LAN segment.
- **Alternate**—An alternate path to the root bridge to that provided by the current root port.
- **Backup**—A backup for the path provided by a designated port toward the leaves of the spanning tree. Backup ports can exist only where two ports are connected together in a loopback by a point-to-point link or bridge with two or more connections to a shared LAN segment.
- **Disabled**—A port that has no role within the operation of spanning tree.

Port roles are assigned as follows:

- A root port or designated port role includes the port in the active topology.
- An alternate port or backup port role excludes the port from the active topology.

RSTP Port States

The port state controls the forwarding and learning processes and provides the values of discarding, learning, and forwarding. [Table 21-4](#) provides a comparison between STP port states and RSTP port states.

Table 21-4 Comparison Between STP and RSTP Port States

Operational Status	STP Port State	RSTP Port State	Port Included in Active Topology
Enabled	Blocking ¹	Discarding ²	No
Enabled	Listening	Discarding	No
Enabled	Learning	Learning	Yes
Enabled	Forwarding	Forwarding	Yes
Disabled	Disabled	Discarding	No

1. IEEE 802.1D port state designation.

2. IEEE 802.1w port state designation. Discarding is the same as blocking in RSTP and MST.

In a stable topology, RSTP ensures that every root port and designated port transition to forwarding, and ensures that all alternate ports and backup ports are always in the discarding state.

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 How Prestandard IEEE 802.1s MST Works

These sections describe Multiple Spanning Tree (MST):

- [IEEE 802.1s MST Overview, page 21-15](#)
- [MST-to-PVST Interoperability, page 21-16](#)
- [Common Spanning Tree, page 21-17](#)
- [MST Instances, page 21-18](#)
- [MST Configuration Parameters, page 21-18](#)
- [MST Regions, page 21-18](#)

- [Message Age and Hop Count, page 21-20](#)
- [Default STP Configuration, page 21-20](#)

IEEE 802.1s MST Overview

MST in this release is based on the draft version of the IEEE standard. 802.1s for MST is an amendment to 802.1Q. MST extends the IEEE 802.1w rapid spanning tree (RST) algorithm to multiple spanning trees. This extension provides both rapid convergence and load balancing in a VLAN environment. MST converges faster than PVST+. MST is backward compatible with 802.1D STP, 802.1w (rapid spanning tree protocol [RSTP]), and the Cisco PVST+ architecture.

MST allows you to build multiple spanning trees over trunks. You can group and associate VLANs to spanning tree instances. Each instance can have a topology independent of other spanning tree instances. This new architecture provides multiple forwarding paths for data traffic and enables load balancing. Network fault tolerance is improved because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

In large networks, you can more easily administer the network and use redundant paths by locating different VLAN and spanning tree instance assignments in different parts of the network. A spanning tree instance can exist only on bridges that have compatible VLAN instance assignments. You must configure a set of bridges with the same MST configuration information, which allows them to participate in a specific set of spanning tree instances. Interconnected bridges that have the same MST configuration are referred to as an *MST region*.

MST uses the modified RSTP version called the Multiple Spanning Tree Protocol (MSTP). The MST feature has these characteristics:

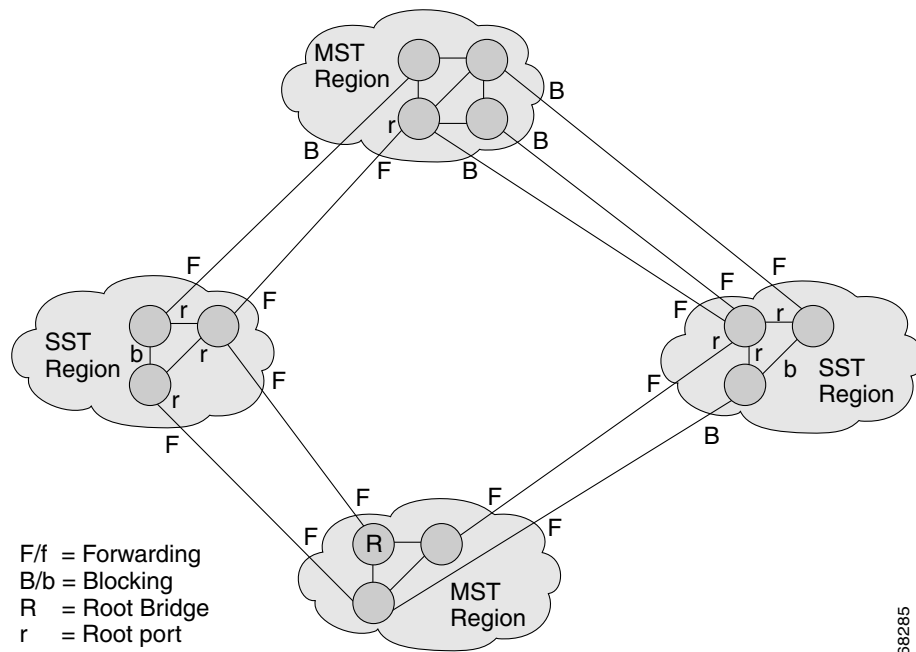
- MST runs a variant of spanning tree called internal spanning tree (IST). IST augments the common spanning tree (CST) information with internal information about the MST region. The MST region appears as a single bridge to adjacent single spanning tree (SST) and MST regions.
- A bridge running MST provides interoperability with single spanning tree bridges as follows:
 - MST bridges run IST, which augments the common spanning tree (CST) information with internal information about the MST region.
 - IST connects all the MST bridges in the region and appears as a subtree in the CST that includes the whole bridged domain. The MST region appears as a virtual bridge to adjacent SST bridges and MST regions.
 - The common and internal spanning tree (CIST) is the collection of ISTs in each MST region, the CST that interconnects the MST regions, and the SST bridges. CIST is the same as an IST inside an MST region and the same as CST outside an MST region. The STP, RSTP, and MSTP together elect a single bridge as the root of the CIST.
- MST establishes and maintains additional spanning trees within each MST region. These spanning trees are referred to as MST instances (MSTIs). The IST is numbered 0, and the MSTIs are numbered 1,2,3, and so on. Any MSTI is local to the MST region that is independent of MSTIs in another region, even if the MST regions are interconnected. MST instances combine with the IST at the boundary of MST regions to become the CST as follows:
 - Spanning tree information for an MSTI is contained in an MSTP record (M-record).
M-records are always encapsulated within MST BPDUs (MST BPDUs). The original spanning trees computed by MSTP are called M-trees. M-trees are active only within the MST region. M-trees merge with the IST at the boundary of the MST region and form the CST.
- MST provides interoperability with PVST+ by generating PVST+ BPDUs for the non-CST VLANs.

- MST supports some of the PVST+ extensions in MSTP as follows:
 - UplinkFast and BackboneFast are not available in MST mode; they are included in RSTP.
 - PortFast is supported.
 - BPDU filter and BPDU guard are supported in MST mode.
 - Loop guard and root guard are supported in MST. MST preserves the VLAN 1 disabled functionality except that BPDUs are still transmitted in VLAN 1.
 - MST switches operate as if MAC reduction is enabled.
 - For private VLANs (PVLANS), secondary VLANs must be mapped to the same instance as the primary.

MST-to-PVST Interoperability

A virtual bridged LAN may contain interconnected regions of single spanning tree (SST) and MST bridges. [Figure 21-8](#) shows this relationship.

Figure 21-8 Network with Interconnected SST and MST Regions



An MST region appears as an SST or pseudobridge to STP running in the SST region. Pseudobridges operate as follows:

- The same values for root identifiers and root path costs are sent in all BPDUs of all the pseudobridge ports. Pseudobridges differ from a single SST bridge as follows:
 - The pseudobridge BPDUs have different bridge identifiers. This difference does not affect STP operation in the neighboring SST regions because the root identifier and root cost are the same.
 - BPDUs sent from the pseudobridge ports may have significantly different message ages. Because the message age increases by 1 second for each hop, the difference in the message age is in the order of seconds.

- Data traffic from one port of a pseudobridge (a port at the edge of a region) to another port follows a path entirely contained within the pseudobridge or MST region.
- Data traffic belonging to different VLANs may follow different paths within the MST regions established by MST.
- Loop prevention is achieved by either of the following:
 - Blocking the appropriate pseudobridge ports by allowing one forwarding port on the boundary and blocking all other ports.
 - Setting the CST partitions to block the ports of the SST regions.
- A pseudobridge differs from a single SST bridge because the BPDUs sent from the pseudobridge's ports have different bridge identifiers. The root identifier and root cost are the same for both bridges.

These guidelines apply in a topology where you configure MST switches (all in the same region) to interact with PVST+ switches:

- Configure the root for all VLANs inside the MST region as shown in this example:

```
Router# show spanning-tree mst interface gigabitethernet 1/1

GigabitEthernet1/1 of MST00 is root forwarding
Edge port: no (trunk) port guard : none (default)
Link type: point-to-point (auto) bpdu filter: disable (default)
Boundary : boundary (PVST) bpdu guard : disable (default)
Bpdus sent 10, received 310

Instance Role Sts Cost Prio.Nbr Vlns mapped
-----
0 Root FWD 20000 128.1 1-2,4-2999,4000-4094
3 Boun FWD 20000 128.1 3,3000-3999
```

The ports that belong to the MST switch at the boundary simulate PVST+ and send PVST+ BPDUs for all the VLANs.

If you enable loop guard on the PVST+ switches, the ports might change to a loop-inconsistent state when the MST switches change their configuration. To correct the loop-inconsistent state, you must disable and reenab le loop guard on that PVST+ switch.

- Do not locate the root for some or all of the VLANs inside the PVST+ side of the MST switch because when the MST switch at the boundary receives PVST+ BPDUs for all or some of the VLANs on its designated ports, root guard sets the port to the blocking state. Do not designate switches with a slower CPU running PVST+ as a switch running MST.

When you connect a PVST+ switch to two different MST regions, the topology change from the PVST+ switch does not pass beyond the first MST region. In this case, the topology changes are only propagated in the instance to which the VLAN is mapped. The topology change stays local to the first MST region and the CAM entries in the other region are not flushed. To make the topology change visible throughout other MST regions, you can map that VLAN to IST or connect the PVST+ switch to the two regions through access links.

Common Spanning Tree

CST (802.1Q) is a single spanning tree for all the VLANs. In a Catalyst 6500 series switch running PVST+, the VLAN 1 spanning tree corresponds to CST. In a Catalyst 6500 series switch running MST, IST (instance 0) corresponds to CST.

MST Instances

This release supports up to 16 instances; each spanning tree instance is identified by an instance ID that ranges from 0 to 15. Instance 0 is mandatory and is always present. Instances 1 through 15 are optional.

MST Configuration Parameters

MST configuration includes these three parts:

- Name—A 32-character string (null padded) identifying the MST region.
- Revision number—An unsigned 16-bit number that identifies the revision of the current MST configuration.

**Note**

You must set the revision number when required as part of the MST configuration. The revision number is not incremented automatically each time you commit the MST configuration.

- MST configuration table—An array of 4096 bytes. Each byte, interpreted as an unsigned integer, corresponds to a VLAN. The value is the instance number to which the VLAN is mapped. The first byte that corresponds to VLAN 0 and the 4096th byte that corresponds to VLAN 4095 are unused and always set to zero.

You must configure each byte manually. You can use SNMP or the CLI to perform the configuration.

MST BPDUs contain the MST configuration ID and the checksum. An MST bridge accepts an MST BPDU only if the MST BPDU configuration ID and the checksum match its own MST region configuration ID and checksum. If one value is different, the MST BPDU is considered to be an SST BPDU.

MST Regions

These sections describe MST regions:

- [MST Region Overview, page 21-18](#)
- [Boundary Ports, page 21-19](#)
- [IST Master, page 21-19](#)
- [Edge Ports, page 21-19](#)
- [Link Type, page 21-20](#)

MST Region Overview

Interconnected bridges that have the same MST configuration are referred to as an MST region. There is no limit on the number of MST regions in the network.

To form an MST region, bridges can be either of the following:

- An MST bridge that is the only member of the MST region.
- An MST bridge interconnected by a LAN. A LAN's designated bridge has the same MST configuration as an MST bridge. All the bridges on the LAN can process MST BPDUs.

If you connect two MST regions with different MST configurations, the MST regions do the following:

- Load balance across redundant paths in the network. If two MST regions are redundantly connected, all traffic flows on a single connection with the MST regions in a network.
- Provide an RSTP handshake to enable rapid connectivity between regions. However, the handshaking is not as fast as between two bridges. To prevent loops, all the bridges inside the region must agree upon the connections to other regions. This situation introduces a delay. We do not recommend partitioning the network into a large number of regions.

Boundary Ports

A boundary port is a port that connects to a LAN, the designated bridge, of which is either an SST bridge, or a bridge with a different MST configuration. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement message from an RST or MST bridge with a different configuration.

At the boundary, the role of MST ports do not matter; their state is forced to be the same as the IST port state. If the boundary flag is set for the port, the MSTP port-role selection process assigns a port role to the boundary and assigns the same state as the state of the IST port. The IST port at the boundary can take up any port role except a backup port role.

IST Master

The IST master of an MST region is the bridge with the lowest bridge identifier and the least path cost to the CST root. If an MST bridge is the root bridge for CST, then it is the IST master of that MST region. If the CST root is outside the MST region, then one of the MST bridges at the boundary is selected as the IST master. Other bridges on the boundary that belong to the same region eventually block the boundary ports that lead to the root.

If two or more bridges at the boundary of a region have an identical path to the root, you can set a slightly lower bridge priority to make a specific bridge the IST master.

The root path cost and message age inside a region stay constant, but the IST path cost is incremented and the IST remaining hops are decremented at each hop. To display the information about the IST master, path cost, and remaining hops for the bridge, enter the **show spanning-tree mst** command.

Edge Ports

An edge port is a port that is connected to a nonbridging device (for example, a host or a router). A port that connects to a hub is also an edge port if the hub or any LAN that is connected by it does not have a bridge. An edge port can start forwarding as soon as the link is up.

MST requires that you configure all ports for each host or router. To establish rapid connectivity after a failure, you need to block the nonedge designated ports of an intermediate bridge. If the port connects to another bridge that can send back an agreement, then the port starts forwarding immediately. Otherwise, the port needs twice the forward delay time to start forwarding again. You must explicitly configure the ports that are connected to the hosts and routers as edge ports while using MST.

To prevent a misconfiguration, the PortFast operation is turned off if the port receives a BPDU. To display the configured and operational status of PortFast, enter the **show spanning-tree mst interface** command.

Link Type

Rapid connectivity is established only on point-to-point links. You must configure ports explicitly to a host or router. However, cabling in most networks meets this requirement, and you can avoid explicit configuration by treating all full-duplex links as point-to-point links by entering the **spanning-tree linktype** command.

Message Age and Hop Count

IST and MST instances do not use the message age and maximum age timer settings in the BPDU. IST and MST use a separate hop-count process that is very similar to the IP TTL process. You can configure each MST bridge with a maximum hop count. The root bridge of the instance sends a BPDU (or M-record) with the remaining hop count that is equal to the maximum hop count. When a bridge receives a BPDU (or M-record), it decrements the received remaining hop count by one. The bridge discards the BPDU (M-record) and ages out the information held for the port if the count reaches zero after decrementing. The nonroot bridges propagate the decremented count as the remaining hop count in the BPDUs (M-records) they generate.

The message age and maximum age timer settings in the RST portion of the BPDU remain the same throughout the region, and the same values are propagated by the region's designated ports at the boundary.

Default STP Configuration

Table 21-5 shows the default STP configuration.

Table 21-5 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)	<ul style="list-style-type: none"> • 10-Gigabit Ethernet: 2 • Gigabit Ethernet: 4 • Fast Ethernet: 19 • Ethernet: 100
STP VLAN port priority (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	128
STP VLAN port cost (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports)	<ul style="list-style-type: none"> • 10-Gigabit Ethernet: 2 • Gigabit Ethernet: 4 • Fast Ethernet: 19 • Ethernet: 100

Table 21-5 STP Default Configuration (continued)

Feature	Default Value
Hello time	2 seconds
Forward delay time	15 seconds
Maximum aging time	20 seconds
Mode	PVST

STP and MST Configuration Guidelines and Restrictions

When configuring MST, follow these guidelines and restrictions:

- Do not disable spanning tree on any VLAN in any of the PVST bridges.
- Do not use PVST bridges as the root of CST.
- Ensure that all PVST spanning tree root bridges have lower (numerically higher) priority than the CST root bridge.
- Ensure that trunks carry all of the VLANs mapped to an instance or do not carry any VLANs at all for this instance.
- Do not connect switches with access links because access links may partition a VLAN.
- Any MST configuration involving a large number of either existing or new logical VLAN ports should be completed during a maintenance window because the complete MST database gets reinitialized for any incremental change (such as adding new VLANs to instances or moving VLANs across instances).

Configuring STP

These sections describe how to configure STP on VLANs:

- [Enabling STP, page 21-22](#)
- [Enabling the Extended System ID, page 21-23](#)
- [Configuring the Root Bridge, page 21-24](#)
- [Configuring a Secondary Root Bridge, page 21-25](#)
- [Configuring STP Port Priority, page 21-26](#)
- [Configuring STP Port Cost, page 21-28](#)
- [Configuring the Bridge Priority of a VLAN, page 21-29](#)
- [Configuring the Hello Time, page 21-31](#)
- [Configuring the Forward-Delay Time for a VLAN, page 21-31](#)
- [Configuring the Maximum Aging Time for a VLAN, page 21-32](#)
- [Enabling Rapid-PVST, page 21-32](#)

**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.

Enabling STP



Note

STP is enabled by default on VLAN 1 and on all newly created VLANs.

You can enable STP on a per-VLAN basis. The Catalyst 6500 series switch maintains a separate instance of STP for each VLAN (except on VLANs on which you disable STP).

To enable STP on a per-VLAN basis, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i>	Enables STP on a per-VLAN basis. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 21-5 on page 21-20).
	Router(config)# default spanning-tree vlan <i>vlan_ID</i>	Reverts all STP parameters to default values for the specified VLAN.
	Router(config)# no spanning-tree vlan <i>vlan_ID</i>	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 <i>vlan_ID</i>	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
-----
Fa4/4          Desg FWD 200000    128.196 P2p
Fa4/5          Back BLK 200000    128.197 P2p
```

```
Router#
```



Note

You must have at least one interface that is active in VLAN 200 to create a VLAN 200 spanning tree. In this example, two interfaces are active in VLAN 200.

Enabling the Extended System ID



Note

The extended system ID is enabled permanently on chassis that support 64 MAC addresses.

You can enable the extended system ID on chassis that support 1024 MAC addresses (see the [“Understanding the Bridge ID”](#) section on page 21-2).

To enable the extended system ID, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree extend system-id	Enables the extended system ID.
	Router(config)# no spanning-tree extend system-id	Disables the extended system ID.
		Note You cannot disable the extended system ID on chassis that support 64 MAC addresses or when you have configured extended range VLANs (see “STP Default Configuration” section on page 21-20).
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i>	Verifies the configuration.



Note

When you enable or disable the extended system ID, the bridge IDs of all active STP instances are updated, which might change the spanning tree topology.

This example shows how to enable the extended system ID:

```
Router# configure terminal
Router(config)# spanning-tree extend system-id
Router(config)# end
Router#
```

This example shows how to verify the configuration:

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

Configuring the Root Bridge

Catalyst 6500 series switches maintain a separate instance of STP for each active VLAN. A bridge ID, consisting of the bridge priority and the bridge MAC address, is associated with each instance. For each VLAN, the network device with the lowest bridge ID becomes the root bridge for that VLAN.

To configure a VLAN instance to become the root bridge, enter the **spanning-tree vlan *vlan_ID* root** command to modify the bridge priority from the default value (32768) to a significantly lower value.

When you enter the **spanning-tree vlan *vlan_ID* root** command, the switch checks the bridge priority of the current root bridges for each VLAN. When the extended system ID is disabled, the switch sets the bridge priority for the specified VLANs to 8192 if this value will cause the switch to become the root for the specified VLANs. When the extended system ID is enabled, the switch sets the bridge priority for the specified VLANs to 24576 if this value will cause the switch to become the root for the specified VLANs.

If the extended system ID is disabled and if any root bridge for the specified VLANs has a bridge priority lower than 8192, the switch sets the bridge priority for the specified VLANs to 1 less than the lowest bridge priority.

If the extended system ID is enabled and if any root bridge for the specified VLANs has a bridge priority lower than 24576, the switch sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority. (4096 is the value of the least significant bit of a 4-bit bridge priority value; see [Table 21-2 on page 21-3](#).)



Note

The **spanning-tree vlan *vlan_ID* root** command fails if the value required to be the root bridge is less than 1.

The **spanning-tree vlan *vlan_ID* root** command can cause the following effects:

- If the extended system ID is disabled, and if all network devices in VLAN 100 have the default priority of 32768, entering the **spanning-tree vlan 100 root primary** command on the switch sets the bridge priority for VLAN 100 to 8192, which causes the switch to become the root bridge for VLAN 100.
- If the extended system ID is enabled, and if all network devices in VLAN 20 have the default priority of 32768, entering the **spanning-tree vlan 20 root primary** command on the switch sets the bridge priority to 24576, which causes the switch to become the root bridge for VLAN 20.



Caution

The root bridge for each instance of STP should be a backbone or distribution switch. Do not configure an access switch as the STP primary root.

Use the **diameter** keyword to specify the Layer 2 network diameter (that is, the maximum number of bridge hops between any two end stations in the Layer 2 network). When you specify the network diameter, the Catalyst 6500 series switch automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can use the **hello** keyword to override the automatically calculated hello time.

**Note**

To preserve a stable STP topology, we recommend that you avoid configuring the hello time, forward delay time, and maximum age time manually after configuring the Catalyst 6500 series switch as the root bridge.

To configure a Catalyst 6500 series switch as the root bridge, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i> root primary [diameter <i>hops</i> [hello-time <i>seconds</i>]]	Configures a Catalyst 6500 series switch as the root bridge. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 21-5 on page 21-20).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> root	Clears the root bridge configuration.
Step 2	Router(config)# end	Exits configuration mode.

This example shows how to configure the Catalyst 6500 series switch as the root bridge for VLAN 10, with a network diameter of 4:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# end
Router#
```

Configuring a Secondary Root Bridge

When you configure a Catalyst 6500 series switch as the secondary root, the STP bridge priority is modified from the default value (32768) so that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other network devices in the network use the default bridge priority of 32768).

If the extended system ID is enabled, STP sets the bridge priority to 28672. If the extended system ID is disabled, STP sets the bridge priority to 16384.

You can run this command on more than one Catalyst 6500 series switch to configure multiple backup root bridges. Use the same network diameter and hello time values as you used when configuring the primary root bridge.

To configure a Catalyst 6500 series switch as the secondary root bridge, perform this task:

	Command	Purpose
Step 1	Router(config)# [no] spanning-tree vlan <i>vlan_ID</i> root secondary [diameter <i>hops</i> [hello-time <i>seconds</i>]]	Configures a Catalyst 6500 series switch as the secondary root bridge. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> root	Clears the root bridge configuring.
Step 2	Router(config)# end	Exits configuration mode.

This example shows how to configure the Catalyst 6500 series switch as the secondary root bridge for VLAN 10, with a network diameter of 4:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root secondary diameter 4
Router(config)# end
Router#
```

Configuring STP Port Priority

If a loop occurs, STP considers port priority when selecting a LAN port to put into the forwarding state. You can assign higher priority values to LAN ports that you want STP to select first and lower priority values to LAN ports that you want STP to select last. If all LAN ports have the same priority value, STP puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports. The possible priority range is 0 through 240 (default 128), configurable in increments of 16.

Cisco IOS uses the port priority value when the LAN port is configured as an access port and uses VLAN port priority values when the LAN port is configured as a trunk port.

To configure the STP port priority of a Layer 2 LAN interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ <i>slot/port</i> } {port-channel <i>port_channel_number</i> }}	Selects an interface to configure.
Step 2	Router(config-if)# spanning-tree port-priority <i>port_priority</i>	Configures the port priority for the LAN interface. The <i>port_priority</i> 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 <i>vlan_ID</i> port-priority <i>port_priority</i>	Configures the VLAN port priority for the LAN interface. The <i>port_priority</i> value can be from 1 to 252 in increments of 4. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
	Router(config-if)# [no] spanning-tree vlan <i>vlan_ID</i> 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 {type ¹ slot/port} {port-channel port_channel_number} Router# show spanning-tree vlan vlan_ID	Verifies the configuration.

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

This example shows how to configure the STP port priority of Fast Ethernet port 4/4:

```
Router# configure terminal
Router(config)# interface fastethernet 4/4
Router(config-if)# spanning-tree port-priority 160
Router(config-if)# end
Router#
```

This example shows how to verify the configuration of Fast Ethernet port 4/4:

```
Router# show spanning-tree interface fastethernet 4/4
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#
```

Fastethernet 4/4 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 Fast Ethernet port 4/4:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree vlan 200 port-priority 64
Router(config-if)# ^Z
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 fastethernet 4/4
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 fastEthernet 4/4
Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          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 {{type ¹ slot/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 <i>port_cost</i> value can be from 1 to 200000000 (1 to 65535 in Release 12.1(2)E and earlier releases).
	Router(config-if)# no spanning-tree cost	Reverts to the default port cost.
Step 3	Router(config-if)# [no] spanning-tree vlan vlan_ID cost port_cost	Configures the VLAN port cost for the LAN interface. The <i>port_cost</i> value can be from 1 to 200000000. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
Step 4	Router(config-if)# no spanning-tree vlan vlan_ID cost	Reverts to the default VLAN port cost.
Step 5	Router(config-if)# end	Exits configuration mode.
Step 6	Router# show spanning-tree interface {type ¹ slot/port} {port-channel port_channel_number} show spanning-tree vlan vlan_ID	Verifies the configuration.

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

This example shows how to change the STP port cost of Fast Ethernet port 4/4:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree cost 1000
Router(config-if)# ^Z
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree interface fastEthernet 4/4
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 fastEthernet 4/4
Router(config-if)# spanning-tree vlan 200 cost 2000
Router(config-if)# ^Z
Router#

```

This example shows how to verify the configuration:

```

Router# show spanning-tree vlan 200 interface fastEthernet 4/4
Interface          Role Sts Cost          Prio.Nbr Status
-----
Fa4/4              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 fastEthernet 4/4
Interface          Role Sts Cost          Prio.Nbr Status
-----
Fa4/4              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 when the extended system ID is disabled, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree vlan <i>vlan_ID</i> priority <i>bridge_priority</i>	Configures the bridge priority of a VLAN when the extended system ID is disabled. The <i>bridge_priority</i> value can be from 1 to 65535. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> priority	Reverts to the default bridge priority value.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> bridge [detail]	Verifies the configuration.

To configure the STP bridge priority of a VLAN when the extended system ID is enabled, perform this task:

	Command	Purpose
Step 1	Router(config)# [no] spanning-tree vlan <i>vlan_ID</i> 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 <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> priority	Reverts to the default bridge priority value.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> 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 33792
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree vlan 200 bridge
                                Hello Max  Fwd
Vlan                          Time Age Delay  Protocol
-----
VLAN200                      33792 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 <i>vlan_ID</i> hello-time <i>hello_time</i>	Configures the hello time of a VLAN. The <i>hello_time</i> value can be from 1 to 10 seconds. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> hello-time	Reverts to the default hello time.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> 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                          Time Age Delay Protocol
-----
VLAN200                      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 <i>vlan_ID</i> forward-time <i>forward_time</i>	Configures the forward time of a VLAN. The <i>forward_time</i> value can be from 4 to 30 seconds. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> forward-time	Reverts to the default forward time.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> 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
```

Vlan	Bridge ID	Hello Time	Max Age	Fwd 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 <i>vlan_ID</i> max-age <i>max_age</i>	Configures the maximum aging time of a VLAN. The <i>max_age</i> value can be from 6 to 40 seconds. The <i>vlan_ID</i> value can be 1 through 4094, except reserved VLANs (see Table 15-1 on page 15-2).
	Router(config)# no spanning-tree vlan <i>vlan_ID</i> max-age	Reverts to the default maximum aging time.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree vlan <i>vlan_ID</i> 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
```

Vlan	Bridge ID	Hello Time	Max Age	Fwd Delay	Protocol
VLAN200	49152 0050.3e8d.64c8	2	36	15	ieee

Router#

Enabling Rapid-PVST

Rapid-PVST uses the existing PVST+ framework for configuration and interaction with other features. It also supports some of the PVST+ extensions.

To enable Rapid-PVST mode on the switch, enter the **spanning-tree mode rapid-pvst** command in privileged mode. To configure the switch in Rapid-PVST mode, see the “Configuring STP” section on [page 21-21](#).

Specifying the Link Type

Rapid connectivity is established only on point-to-point links. Spanning tree views a point-to-point link as a segment connecting only two switches running the spanning tree algorithm. Because the switch assumes that all full-duplex links are point-to-point links and that half-duplex links are shared links, you can avoid explicitly configuring the link type. To configure a specific link type, enter the **spanning-tree linktype** command.

Restarting Protocol Migration

A switch running both MSTP and RSTP supports a built-in protocol migration process that enables the switch to interoperate with legacy 802.1D switches. If this switch receives a legacy 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MSTP switch can also detect that a port is at the boundary of a region when it receives a legacy BPDU, or an MST BPDU (version 3) associated with a different region, or an RST BPDU (version 2).

However, the switch does not automatically revert to the MSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. A switch also might continue to assign a boundary role to a port when the switch to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches) on the entire switch, you can use the **clear spanning-tree detected-protocols** privileged EXEC command. To restart the protocol migration process on a specific interface, enter the **clear spanning-tree detected-protocols interface** *interface-id* privileged EXEC command.

Configuring Prestandard IEEE 802.1s MST

Release 12.2SX supports MST. These sections describe how to configure MST:

- [Enabling MST, page 21-33](#)
- [Displaying MST Configurations, page 21-35](#)
- [Configuring MST Instance Parameters, page 21-38](#)
- [Configuring MST Instance Port Parameters, page 21-39](#)
- [Restarting Protocol Migration, page 21-39](#)

Enabling MST

To enable and configure MST, perform these tasks in privileged mode:

	Command	Purpose
Step 1	Router# show spanning-tree mst configuration	Displays the current MST configuration.
Step 2	Router(config)# spanning-tree mode mst	Configures MST mode.
Step 3	Router(config)# spanning-tree mst configuration	Configures the MST region by entering the MST configuration submenu.
	Router(config)# no spanning-tree mst configuration	Clears the MST configuration.

	Command	Purpose
Step 4	Router(config-mst)# show current	Displays the current MST configuration from within the MST configuration submode.
Step 5	Router(config-mst)# name <i>name</i> revision <i>revision_number</i> instance <i>instance_number</i> vlan <i>vlan_range</i>	Enters the MST configuration.
Step 6	Router(config-mst)# no instance <i>instance_number</i>	(Optional) Unmaps all VLANs that were mapped to an instance.
Step 7	Router(config-mst)# no instance <i>instance_number</i> vlan <i>vlan_number</i>	(Optional) Unmaps a VLAN from an instance.
Step 8	Router(config-mst)# end	Applies the configuration and exit configuration mode.
Step 9	Router# show spanning-tree mst config	Shows the MST configuration from the global configuration mode.

These examples show how to enable MST:

```

Router# show spanning-tree mst configuration
% Switch is not in mst mode
Name      []
Revision  0
Instance  Vlans mapped
-----
0          1-4094
-----

Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# spanning-tree mode mst

Router(config)# spanning-tree mst configuration

Router(config-mst)# show current
Current MST configuration
Name      []
Revision  0
Instance  Vlans mapped
-----
0          1-4094
-----

Router(config-mst)# name cisco
Router(config-mst)# revision 2
Router(config-mst)# instance 1 vlan 1
Router(config-mst)# instance 2 vlan 1-1000
Router(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0          1001-4094
2          1-1000
-----

Router(config-mst)# no instance 2
Router(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0          1-4094
-----

Router(config-mst)# instance 1 vlan 2000-3000

```



```

Router(config-mst)# no instance 1 vlan 2500
Router(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0          1-1999,2500,3001-4094
1          2000-2499,2501-3000
-----

Router(config)# exit
Router(config)# no spanning-tree mst configuration
Router(config)# do show spanning-tree mst configuration
Name      []
Revision  0
Instance  Vlans mapped
-----
0          1-4094
-----

```

Displaying MST Configurations

To display MST configurations, perform these tasks in MST mode:

	Command	Purpose
Step 1	Router# show spanning-tree mst configuration	Displays the active configuration.
Step 2	Router# show spanning-tree mst [<i>detail</i>]	Displays information about the MST instances currently running.
Step 3	Router# show spanning-tree mst <i>instance-id</i> [<i>detail</i>]	Displays information about a specific MST instance.
Step 4	Router# show spanning-tree mst interface <i>interface name</i> [<i>detail</i>]	Displays information for a given port.
Step 5	Router# show spanning-tree mst <i>number interface</i> <i>interface name</i> [<i>detail</i>]	Displays MST information for a given port and a given instance.
Step 6	Router# show spanning-tree mst [<i>x</i>] [<i>interface Y</i>] <i>detail</i>	Displays detailed MST information.
Step 7	Router# show spanning-tree vlan <i>vlan_ID</i>	Displays VLAN information in MST mode.

These examples show how to display spanning tree VLAN configurations in MST mode:

```

Router(config)# spanning-tree mst configuration
Router(config-mst)# instance 1 vlan 1-10
Router(config-mst)# name cisco
Router(config-mst)# revision 1
Router(config-mst)# ^Z

Router# show spanning-tree mst configuration
Name      [cisco]
Revision  1
Instance  Vlans mapped
-----
0          11-4094
1          1-10
-----

```

Router# **show spanning-tree mst**

```
##### MST00          vlans mapped: 11-4094
Bridge      address 00d0.00b8.1400 priority 32768 (32768 sysid 0)
Root        address 00d0.004a.3c1c priority 32768 (32768 sysid 0)
              port Fa4/48          path cost 203100
IST master  this switch
Operational hello time 2, forward delay 15, max age 20, max hops 20
Configured  hello time 2, forward delay 15, max age 20, max hops 20
```

Interface	Role	Sts	Cost	Prio.Nbr	Status
Fa4/4	Back	BLK	1000	160.196	P2p
Fa4/5	Desg	FWD	200000	128.197	P2p
Fa4/48	Root	FWD	200000	128.240	P2p Bound(STP)

```
##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400 priority 32769 (32768 sysid 1)
Root        this switch for MST01
```

Interface	Role	Sts	Cost	Prio.Nbr	Status
Fa4/4	Back	BLK	1000	160.196	P2p
Fa4/5	Desg	FWD	200000	128.197	P2p
Fa4/48	Boun	FWD	200000	128.240	P2p Bound(STP)

Router# **show spanning-tree mst 1**

```
##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400 priority 32769 (32768 sysid 1)
Root        this switch for MST01
```

Interface	Role	Sts	Cost	Prio.Nbr	Status
Fa4/4	Back	BLK	1000	160.196	P2p
Fa4/5	Desg	FWD	200000	128.197	P2p
Fa4/48	Boun	FWD	200000	128.240	P2p Bound(STP)

Router# **show spanning-tree mst interface fastEthernet 4/4**

```
FastEthernet4/4 of MST00 is backup blocking
Edge port:no          (default)          port guard :none          (default)
Link type:point-to-point (auto)          bpdu filter:disable      (default)
Boundary :internal    bpdu guard :disable      (default)
Bpdus sent 2, received 368
```

Instance	Role	Sts	Cost	Prio.Nbr	Vlans mapped
0	Back	BLK	1000	160.196	11-4094
1	Back	BLK	1000	160.196	1-10

Router# **show spanning-tree mst 1 interface fastEthernet 4/4**

```
FastEthernet4/4 of MST01 is backup blocking
Edge port:no          (default)          port guard :none          (default)
Link type:point-to-point (auto)          bpdu filter:disable      (default)
Boundary :internal    bpdu guard :disable      (default)
Bpdus (MRecords) sent 2, received 364
```

Instance	Role	Sts	Cost	Prio.Nbr	Vlans mapped
1	Back	BLK	1000	160.196	1-10

Router# **show spanning-tree mst 1 detail**

```
##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400 priority 32769 (32768 sysid 1)
Root        this switch for MST01

FastEthernet4/4 of MST01 is backup blocking
Port info   port id      160.196 priority 160 cost      1000
Designated root address 00d0.00b8.1400 priority 32769 cost      0
Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.197
Timers:message expires in 5 sec, forward delay 0, forward transitions 0
Bpdus (MRecords) sent 123, received 1188

FastEthernet4/5 of MST01 is designated forwarding
Port info   port id      128.197 priority 128 cost      200000
Designated root address 00d0.00b8.1400 priority 32769 cost      0
Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.197
Timers:message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 1188, received 123

FastEthernet4/48 of MST01 is boundary forwarding
Port info   port id      128.240 priority 128 cost      200000
Designated root address 00d0.00b8.1400 priority 32769 cost      0
Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.240
Timers:message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 78, received 0
```

Router# **show spanning-tree vlan 10**

```
MST01
Spanning tree enabled protocol mstp
Root ID    Priority    32769
           Address     00d0.00b8.1400
           This bridge is the root
           Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
           Address     00d0.00b8.1400
           Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Interface   Role Sts Cost      Prio.Nbr Status
-----
Fa4/4       Back BLK 1000    160.196 P2p
Fa4/5       Desg FWD 200000    128.197 P2p
```

Router# **show spanning-tree summary**

```
Root bridge for:MST01
EtherChannel misconfiguration guard is enabled
Extended system ID is enabled
Portfast is disabled 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
-----
MST00          1          0          0          2          3
MST01          1          0          0          2          3
-----
2 msts          2          0          0          4          6
Router#
```

Configuring MST Instance Parameters

To configure MST instance parameters, perform these tasks:

	Command	Purpose
Step 1	Router(config)# spanning-tree mst X priority Y	Configures the priority for an MST instance.
Step 2	Router(config)# spanning-tree mst X root [primary secondary]	Configures the bridge as root for an MST instance.
Step 3	Router# show spanning-tree mst	Verifies the configuration.

This example shows how to configure MST instance parameters:

```
Router(config)# spanning-tree mst 1 priority ?
<0-61440> bridge priority in increments of 4096

Router(config)# spanning-tree mst 1 priority 1
% Bridge Priority must be in increments of 4096.
% Allowed values are:
  0      4096  8192  12288  16384  20480  24576  28672
 32768  36864  40960  45056  49152  53248  57344  61440

Router(config)# spanning-tree mst 1 priority 49152
Router(config)#

Router(config)# spanning-tree mst 0 root primary
mst 0 bridge priority set to 24576
mst bridge max aging time unchanged at 20
mst bridge hello time unchanged at 2
mst bridge forward delay unchanged at 15
Router(config)# ^Z
Router#

Router# show spanning-tree mst

##### MST00          vlans mapped: 11-4094
Bridge      address 00d0.00b8.1400 priority 24576 (24576 sysid 0)
Root        this switch for CST and IST
Configured  hello time 2, forward delay 15, max age 20, max hops 20

Interface    Role Sts Cost      Prio.Nbr Status
-----
Fa4/4        Back BLK 1000      160.196 P2p
Fa4/5        Desg FWD 200000      128.197 P2p
Fa4/48       Desg FWD 200000      128.240 P2p Bound(STP)

##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400 priority 49153 (49152 sysid 1)
Root        this switch for MST01

Interface    Role Sts Cost      Prio.Nbr Status
-----
Fa4/4        Back BLK 1000      160.196 P2p
Fa4/5        Desg FWD 200000      128.197 P2p
Fa4/48       Boun FWD 200000      128.240 P2p Bound(STP)

Router#
```

Configuring MST Instance Port Parameters

To configure MST instance port parameters, perform these tasks:

	Command	Purpose
Step 1	Router(config-if)# spanning-tree mst x cost y	Configures the MST instance port cost.
Step 2	Router(config-if)# spanning-tree mst x port-priority y	Configures the MST instance port priority.
Step 3	Router# show spanning-tree mst x interface y	Verifies the configuration.

This example shows how to configure MST instance port parameters:

```
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree mst 1 ?
    cost          Change the interface spanning tree path cost for an instance
    port-priority Change the spanning tree port priority for an instance

Router(config-if)# spanning-tree mst 1 cost 1234567
Router(config-if)# spanning-tree mst 1 port-priority 240
Router(config-if)# ^Z

Router# show spanning-tree mst 1 interface fastEthernet 4/4

FastEthernet4/4 of MST01 is backup blocking
Edge port:no (default) port guard :none (default)
Link type:point-to-point (auto) bpdu filter:disable (default)
Boundary :internal bpdu guard :disable (default)
Bpdus (MRecords) sent 125, received 1782

Instance Role Sts Cost Prio.Nbr Vlans mapped
-----
1 Back BLK 1234567 240.196 1-10

Router#
```

Restarting Protocol Migration

A switch running both MSTP and RSTP supports a built-in protocol migration mechanism that enables the switch to interoperate with legacy 802.1D switches. If this switch receives a legacy 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. An MSTP switch can also detect that a port is at the boundary of a region when it receives a legacy BPDU, an MST BPDU (version 3) associated with a different region, or an RST BPDU (version 2).

However, the switch does not automatically revert to the MSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether the legacy switch has been removed from the link unless the legacy switch is the designated switch. A switch also might continue to assign a boundary role to a port when the switch to which it is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches) on the entire switch, you can use the **clear spanning-tree detected-protocols** privileged EXEC command. Use the **clear spanning-tree detected-protocols interface interface-id** privileged EXEC command to restart the protocol migration process on a specific interface.

This example shows how to restart protocol migration:

```
Router# clear spanning-tree detected-protocols interface fastEthernet 4/4  
Router#
```



Configuring Optional STP Features

This chapter describes how to configure optional STP features.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How PortFast Works, page 22-2](#)
- [Understanding How BPDU Guard Works, page 22-2](#)
- [Understanding How PortFast BPDU Filtering Works, page 22-2](#)
- [Understanding How UplinkFast Works, page 22-3](#)
- [Understanding How BackboneFast Works, page 22-4](#)
- [Understanding How EtherChannel Guard Works, page 22-6](#)
- [Understanding How Root Guard Works, page 22-6](#)
- [Understanding How Loop Guard Works, page 22-6](#)
- [Enabling PortFast, page 22-8](#)
- [Enabling PortFast BPDU Filtering, page 22-10](#)
- [Enabling BPDU Guard, page 22-11](#)
- [Enabling UplinkFast, page 22-12](#)
- [Enabling BackboneFast, page 22-13](#)
- [Enabling EtherChannel Guard, page 22-14](#)
- [Enabling Root Guard, page 22-14](#)
- [Enabling Loop Guard, page 22-15](#)



Note

For information on configuring the spanning tree protocol (STP), see [Chapter 21, “Configuring STP and Prestandard IEEE 802.1s MST.”](#)

Understanding How PortFast Works

STP PortFast causes a Layer 2 LAN port configured as an access port to enter the forwarding state immediately, bypassing the listening and learning states. You can use PortFast on Layer 2 access ports connected to a single workstation or server to allow those devices to connect to the network immediately, instead of waiting for STP to converge. Interfaces connected to a single workstation or server should not receive bridge protocol data units (BPDUs). When configured for PortFast, a port is still running the spanning tree protocol. A PortFast enabled port can immediately transition to the blocking state if necessary (this could happen on receipt of a superior BPDU). PortFast can be enabled on trunk ports. PortFast can have an operational value that is different from the configured value.

**Caution**

Because the purpose of PortFast is to minimize the time that access ports must wait for STP to converge, it should only be used on access ports. If you enable PortFast on a port connected to a switch, you might create a temporary bridging loop.

Understanding How BPDU Guard Works

When enabled on a port, BPDU Guard shuts down a port that receives a BPDU. When configured globally, BPDU Guard is only effective on ports in the operational PortFast state. In a valid configuration, PortFast Layer 2 LAN interfaces do not receive BPDUs. Reception of a BPDU by a PortFast Layer 2 LAN interface signals an invalid configuration, such as connection of an unauthorized device. BPDU Guard provides a secure response to invalid configurations, because the administrator must manually put the Layer 2 LAN interface back in service. BPDU Guard can be configured at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the PortFast configuration.

**Note**

When enabled globally, BPDU Guard applies to all interfaces that are in an operational PortFast state.

Understanding How PortFast BPDU Filtering Works

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

When configured globally, PortFast BPDU filtering applies to all operational PortFast ports. Ports in an operational PortFast state are supposed to be connected to hosts, that typically drop BPDUs. If an operational PortFast port receives a BPDU, it immediately loses its operational PortFast status. In that case, PortFast BPDU filtering is disabled on this port and STP resumes sending BPDUs on this port.

PortFast BPDU filtering can also be configured on a per-port basis. When PortFast BPDU filtering is explicitly configured on a port, it does not send any BPDUs and drops all BPDUs it receives.

**Caution**

Explicate configuring PortFast BPDU filtering on a port that is not connected to a host can result in bridging loops as the port will ignore any BPDU it receives and go to forwarding.

When you enable PortFast BPDU filtering globally and set the port configuration as the default for PortFast BPDU filtering (see the [“Enabling PortFast BPDU Filtering” section on page 22-10](#)), 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 22-1 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 22-1 PortFast BPDU Filtering Port Configurations

Per-Port Configuration	Global Configuration	PortFast State	PortFast BPDU Filtering State
Default	Enable	Enable	Enable ¹
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

1. The port transmits at least 10 BPDUs. If this port receives any BPDUs, then PortFast and PortFast BPDU filtering are disabled.

Understanding How UplinkFast Works

UplinkFast provides fast convergence after a direct link failure and achieves load balancing between redundant Layer 2 links using uplink groups. An uplink group is a set of Layer 2 LAN interfaces (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

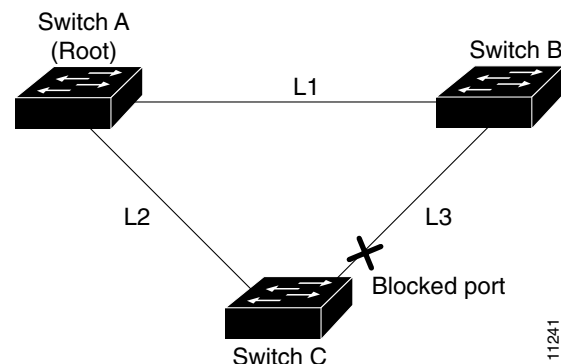


Note

UplinkFast is most useful in wiring-closet switches. This feature may not be useful for other types of applications.

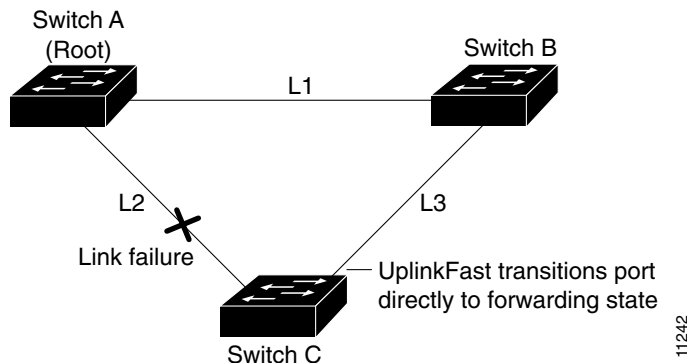
Figure 22-1 shows an example topology with no link failures. Switch A, the root bridge, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 22-1 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 22-2. This switchover takes approximately one to five seconds.

Figure 22-2 UplinkFast Example After Direct Link Failure



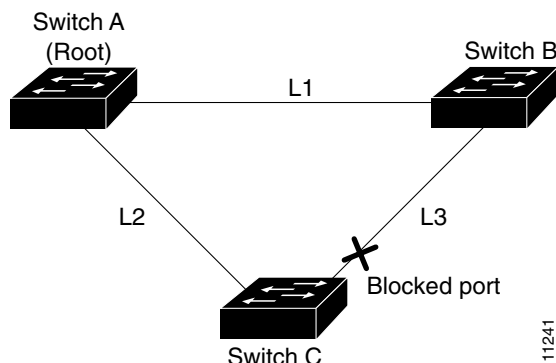
Understanding How BackboneFast Works

BackboneFast is initiated when a root port or blocked port on a network device receives inferior BPDU 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.

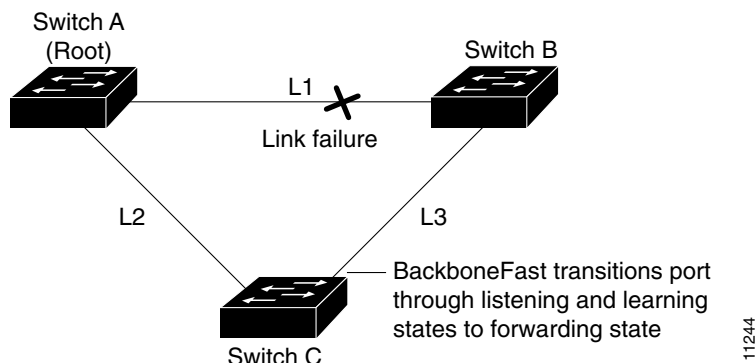
The network device tries to determine if it has an alternate path to the root bridge. If the inferior BPDU arrives on a blocked port, the root port and other blocked ports on the network device become alternate paths to the root bridge. (Self-looped ports are not considered alternate paths to the root 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.

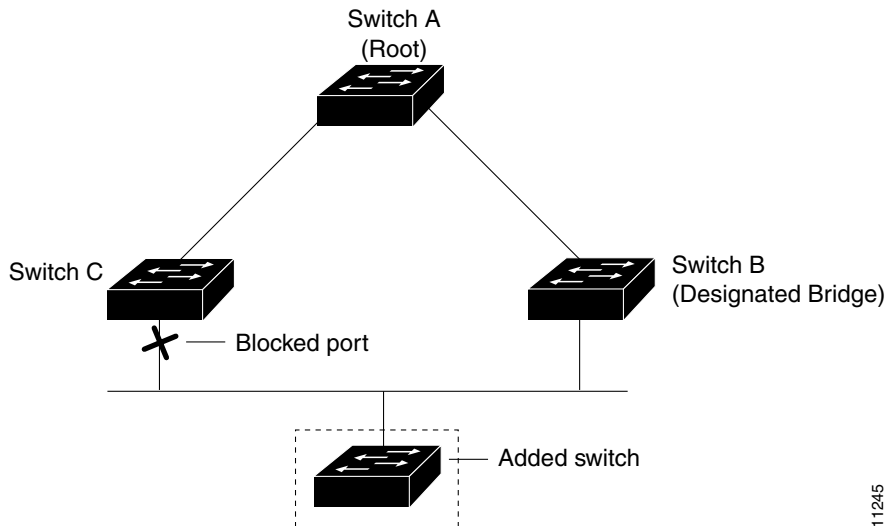
Figure 22-3 shows an example topology with no link failures. Switch A, the root bridge, connects directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that connects directly to Switch B is in the blocking state.

Figure 22-3 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 22-4 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 22-4 BackboneFast Example After Indirect Link Failure

If a new network device is introduced into a shared-medium topology as shown in Figure 22-5, 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 22-5 Adding a Network Device in a Shared-Medium Topology

11245

Understanding How EtherChannel Guard Works

EtherChannel guard detects a misconfigured EtherChannel where interfaces on the Catalyst 6500 series switch are configured as an EtherChannel while interfaces on the other device are not or not all the interfaces on the other device are in the same EtherChannel.

In response to misconfiguration detected on the other device, EtherChannel guard puts interfaces on the Catalyst 6500 series switch into the errdisabled state.

Understanding How Root Guard Works

The STP root guard feature prevents a port from becoming root port or blocked port. If a port configured for root guard receives a superior BPDU, the port immediately goes to the root-inconsistent (blocked) state.

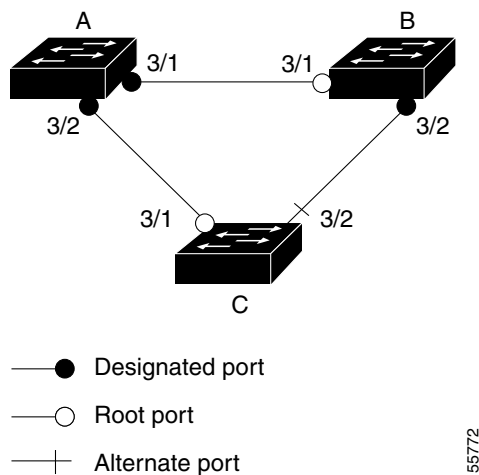
Understanding How Loop Guard Works

Loop guard helps prevent bridging loops that could occur because of a uni-directional link failure on a point-to-point link. When enabled globally, the loop guard applies to all point-to-point ports on the system. Loop guard detects root ports and blocked ports and ensures that they keep receiving BPDUs from their designated port on the segment. If a loop guard enabled root or blocked port stop 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 22-6](#) shows loop guard in a triangle switch configuration.

Figure 22-6 Triangle Switch Configuration with Loop Guard



[Figure 22-6](#) illustrates the following configuration:

- Switches A and B are distribution switches.
- Switch C is an access switch.
- Loop guard is enabled on ports 3/1 and 3/2 on Switches A, B, and C.

Enabling loop guard on a root switch has no effect but provides protection when a root switch becomes a nonroot switch.

When using loop guard, follow these guidelines:

- You cannot enable loop guard on PortFast-enabled ports.
- You cannot enable loop guard if root guard is enabled.

Loop guard interacts with other features as follows:

- Loop guard does not affect the functionality of UplinkFast or BackboneFast.
- Enabling loop guard on ports that are not connected to a point-to-point link will not work.
- Root guard forces a port to be always designated as the root port. Loop guard is effective only if the port is a root port or an alternate port. You cannot enable loop guard and root guard on a port at the same time.
- Loop guard uses the ports known to spanning tree. Loop guard can take advantage of logical ports provided by the Port Aggregation Protocol (PAgP). However, to form a channel, all the physical ports grouped in the channel must have compatible configurations. PAgP enforces uniform configurations of root guard or loop guard on all the physical ports to form a channel.

These caveats apply to loop guard:

- Spanning tree always chooses the first operational port in the channel to send the BPDUs. If that link becomes unidirectional, loop guard blocks the channel, even if other links in the channel are functioning properly.
- If a set of ports that are already blocked by loop guard are grouped together to form a channel, spanning tree loses all the state information for those ports and the new channel port may obtain the forwarding state with a designated role.

- If a channel is blocked by loop guard and the channel breaks, spanning tree loses all the state information. The individual physical ports may obtain the forwarding state with the designated role, even if one or more of the links that formed the channel are unidirectional.



Note You can enable UniDirectional Link Detection (UDLD) to help isolate the link failure. A loop may occur until UDLD detects the failure, but loop guard will not be able to detect it.

- Loop guard has no effect on a disabled spanning tree instance or a VLAN.

Enabling PortFast



Caution

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

To enable PortFast on a Layer 2 access port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {type ¹ slot/port} { port-channel port_channel_number}	Selects a port to configure.
Step 2	Router(config-if)# spanning-tree portfast	Enables PortFast on a Layer 2 access port connected to a single workstation or server.
Step 3	Router(config-if)# spanning-tree portfast default	Disables 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 bpdupfilter default	Enables BPDU filtering globally on the switch.
Step 2	Router# show spanning-tree summary totals	Verifies the configuration.

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


Note

For PVST+ information, see [Chapter 21, “Configuring STP and Prestandard IEEE 802.1s MST.”](#)

```
Router(config)# spanning-tree portfast bpdupfilter 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 bpduguard 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 bpduguard enable
Router(config-if)# ^Z

Router# show spanning-tree interface fastEthernet 4/4

Vlan          Role Sts Cost      Prio.Nbr Status
-----
VLAN0010      Desg FWD 1000      160.196 Edge P2p
Router# show spanning-tree interface fastEthernet 4/4 detail
Router# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  BPDU:sent 0, received 0
Router#
```

Enabling BPDU Guard

To enable BPDU Guard globally, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree portfast bpduguard default Router(config)# no spanning-tree portfast bpduguard default	Enables BPDU Guard globally. Disables BPDU Guard globally.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree summary totals	Verifies the configuration.

This example shows how to enable BPDU Guard:

```
Router# configure terminal
Router(config)# spanning-tree portfast bpduguard
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary totals default
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID is disabled
Portfast is enabled by default
PortFast BPDU Guard is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard is disabled by default
UplinkFast is disabled
BackboneFast is disabled
Pathcost method used is long

Name Blocking Listening Learning Forwarding STP Active
-----
2 vlans 0 0 0 3 3
Router#
```

Enabling UplinkFast

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

To enable UplinkFast, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree uplinkfast [max-update-rate <i>max_update_rate</i>]	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 <i>vlan_ID</i>	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 <i>vlan_ID</i>	Verifies that UplinkFast is enabled.

This example shows how to enable BackboneFast:

```
Router# configure terminal
Router(config)# spanning-tree backbonefast
Router(config)# end
Router#
```

This example shows how to verify that BackboneFast is enabled:

```
Router# show spanning-tree backbonefast
BackboneFast is enabled

BackboneFast statistics
-----
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)    : 0
Number of RLQ request PDUs received (all VLANs)   : 0
Number of RLQ response PDUs received (all VLANs)  : 0
Number of RLQ request PDUs sent (all VLANs)       : 0
Number of RLQ response PDUs sent (all VLANs)      : 0
Router#
```

Enabling EtherChannel Guard

To enable EtherChannel guard, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree etherchannel guard misconfig	Enables EtherChannel guard.
	Router(config)# no spanning-tree etherchannel guard misconfig	Disables EtherChannel guard.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree summary include EtherChannel	Verifies that EtherChannel guard is enabled.

This example shows how to enable EtherChannel guard:

```
Router# configure terminal
Router(config)# spanning-tree etherchannel guard misconfig
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary | include EtherChannel
EtherChannel misconfiguration guard is enabled
```

To display the interfaces that are in the errdisable state, enter the **show interface status err-disable** command.

After the misconfiguration has been cleared, interfaces in the errdisable state might automatically recover. To manually return a port to service, enter a **shutdown** and then a **no shutdown** command for the interface.

Enabling Root Guard

To enable root guard, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {type ¹ slot/port} {port-channel port_channel_number}	Selects a port to configure.
Step 2	Router(config-if)# spanning-tree guard root	Enables root guard.
	Router(config-if)# no spanning-tree guard root	Disables root guard.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show spanning-tree Router# show running interface {type ¹ slot/port} {port-channel port_channel_number}	Verifies the configuration.

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

To display ports that are in the root-inconsistent state, enter the **show spanning-tree inconsistentports** command.

Enabling Loop Guard

To enable loop guard globally on the switch, perform this task:

	Command	Purpose
Step 1	Router(config)# spanning-tree loopguard default	Enables loop guard globally on the switch.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show spanning-tree interface 4/4 detail	Verifies the configuration impact on a port.

This example shows how to enable loop guard globally:

```
Router# configure terminal
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Router(config)# spanning-tree loopguard default
```

```
Router(config)# ^Z
```

```
Router# show spanning-tree interface fastEthernet 4/4 detail
```

```
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  Loop guard is enabled by default on the port
  BPDU:sent 0, received 0
```

To enable loop guard on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {type ¹ slot/port} {port-channel port_channel_number}	Selects a port to configure.
Step 2	Router(config-if)# spanning-tree guard loop	Configures loop guard.
Step 3	Router(config)# end	Exits configuration mode.
Step 4	Router# show spanning-tree interface 4/4 detail	Verifies the configuration impact on that port.

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

This example shows how to enable loop guard:

```
Router# configure terminal
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Router(config)# interface fastEthernet 4/4
```

```
Router(config-if)# spanning-tree guard loop
```

```
Router(config-if)# ^Z
```

This example shows how to verify the configuration:

```
Router# show spanning-tree interface fastEthernet 4/4 detail
```

```
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
```

```
Designated root has priority 32768, address 00d0.00b8.140a
Designated bridge has priority 32768, address 00d0.00b8.140a
Designated port id is 160.196, designated path cost 0
Timers:message age 0, forward delay 0, hold 0
Number of transitions to forwarding state:1
The port is in the portfast mode by portfast trunk configuration
Link type is point-to-point by default
Bpdu filter is enabled
Loop guard is enabled on the port
BPDU:sent 0, received 0
Router#
```



Configuring Layer 3 Interfaces

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



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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:

- [Layer 3 Interface Configuration Guidelines and Restrictions](#), page 23-1
- [Configuring Subinterfaces on Layer 3 Interfaces](#), page 23-2
- [Configuring IPv4 Routing and Addresses](#), page 23-4
- [Configuring IPX Routing and Network Numbers](#), page 23-7
- [Configuring AppleTalk Routing, Cable Ranges, and Zones](#), page 23-8
- [Configuring Other Protocols on Layer 3 Interfaces](#), page 23-9

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.
- Release 12.2(18)SX and later releases supports the **ip unnumbered** command on Layer 3 VLAN interfaces.
- Release 12.2(18)SX and later releases supports the **[no] ip dhcp route [connected | static]** command.
- With Release 12.2(18)SX and later releases, creating a VLAN interface for a Layer 2 VLAN that is not defined in the VLAN database automatically defines the Layer 2 VLAN in the VLAN database. (CSCef93306)

- To support VLAN interfaces, create and configure VLANs and assign VLAN membership to Layer 2 LAN ports. For more information, see [Chapter 15, “Configuring VLANs”](#) and [Chapter 14, “Configuring VTP.”](#)
- Catalyst 6500 series switches do not support:
 - Integrated routing and bridging (IRB)
 - Concurrent routing and bridging (CRB)
 - Remote source-route bridging (RSRB)
- Use bridge groups on VLAN interfaces, sometimes called fall-back bridging, to bridge nonrouted protocols. Bridge groups on VLAN interfaces are supported in software on the MSFC.
- Catalyst 6500 series switches do not support the IEEE bridging protocol for bridge groups. Configure bridge groups to use the VLAN-bridge or the DEC spanning-tree protocol.

Configuring Subinterfaces on Layer 3 Interfaces

When configuring Layer 3 subinterfaces, follow these guidelines and restrictions:

- Release 12.2(18)SX and later releases support these features on LAN port subinterfaces:
 - IPv4 unicast forwarding, including MPLS VPN
 - IPv4 multicast forwarding, including MPLS VPN
 - 6PE
 - EoMPLS
 - IPv4 unnumbered
 - Counters for subinterfaces in MIBS and with the **show vlans** command
 - iBGP and eBGP
 - OSPF
 - EIGRP
 - RIPv1/v2
 - RIPv2
 - ISIS
 - Static routing
 - Unidirectional link routing (UDLR)
 - IGMPv1, IGMPv2, IGMPv3
 - PIMv1, PIMv2
 - SSM IGMPv3lite and URD
 - Stub IP multicast routing
 - IGMP join
 - IGMP static group
 - Multicast routing monitor (MRM)
 - Multicast source discovery protocol (MSDP)
 - SSM

- IPv4 Ping
- IPv6 Ping
- These restrictions apply to releases earlier than Release 12.2(18)SXE:
 - Configure subinterfaces only to support MPLS.
 - Do not configure ISL encapsulation.
- Always use the **native** keyword when the VLAN ID is the ID of the IEEE 802.1Q native VLAN. Do not configure encapsulation on the native VLAN of an IEEE 802.1Q trunk without the **native** keyword.
- Because VLAN IDs are global to the switch, you can use a VLAN internally, on a subinterface, or with a Layer 3 VLAN interface.
 - You cannot configure an internal VLAN on a subinterface or a Layer 3 VLAN interface.
 - You cannot configure a subinterface VLAN on a Layer 3 VLAN interface.
 - You cannot configure a VLAN used with a Layer 3 VLAN interface on a subinterface.



Note You cannot configure a VLAN used on one interface or subinterface on another interface or subinterface.

- You can configure subinterfaces with any normal range or extended range VLAN ID in VTP transparent mode. Because VLAN IDs 1 to 1005 are global in the VTP domain and can be defined on other network devices in the VTP domain, you can use only extended range VLANs with subinterfaces in VTP client or server mode. In VTP client or server mode, normal range VLANs are excluded from subinterfaces.



Note If you configure normal range VLANs on subinterfaces, you cannot change the VTP mode from transparent.

To configure a subinterface, perform this task:

	Command	Purpose
Step 1	Router> enable	Enters privileged EXEC mode.
Step 2	Router# configure terminal	Enters global configuration mode.
Step 3	Router(config)# interface {{type ¹ slot/port.subinterface} {port-channel port_channel_number.subinterface}}	Selects an interface and enters subinterface configuration mode.
Step 4	Router(config-subif)# encapsulation dot1q vlan_ID [native]	Configures 802.1Q encapsulation for the subinterface.
Step 5	Router(config-if)# exit	Returns to global configuration mode.

1. type = ethernet, fastethernet, gigabitethernet, tengigabitethernet, or ge-wan

Configuring IPv4 Routing and Addresses

For complete information and procedures, refer to these publications:

- *Cisco IOS IP and IP Routing Configuration Guide*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/index.htm
- *Cisco IOS IP and IP Routing Command Reference*, Release 12.2, at these URLs:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipras_r/index.htm
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fiprrp_r/index.htm

When configuring IPv4 routing and addresses, follow these guidelines and restrictions:

- For information about the **maximum paths** command, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX 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/fqcpbr1/qcftpbr.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	Exits IPv4 routing protocol configuration mode.
Step 5	Router(config)# interface {vlan vlan_ID} {type¹ 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.

	Command	Purpose
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. 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 unreachable 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 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
- *Cisco IOS AppleTalk and Novell IPX Command Reference*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fatipx_r/index.htm

To configure routing for Internetwork Packet Exchange (IPX) and configure IPX on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# ipx routing	Enables IPX routing.
Step 2	Router(config)# router ipx_routing_protocol	Specifies an IP routing protocol. This step might include other commands, such as specifying the networks to route with the network command.
Step 3	Router(config)# interface {vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}	Selects an interface to configure.
Step 4	Router(config-if)# ipx network [network unnumbered] encapsulation encapsulation_type	Configures the IPX network number. This enables IPX routing on the interface. When you enable IPX routing on the interface, you can also specify an encapsulation type.
Step 5	Router(config-if)# no shutdown	Enables the interface.
Step 6	Router(config-if)# end	Exits configuration mode.
Step 7	Router# show interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}] Router# show ipx interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}] Router# show running-config interfaces [{vlan vlan_ID} {type ¹ slot/port} {port-channel port_channel_number}]	Verifies the configuration.

1. type = ethernet, fastethernet, gigabitethernet, or tengigabitethernet, or ge-wan

This example shows how to enable IPX routing and assign an IPX network address to interface VLAN 100:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ipx routing
Router(config)# ipx router rip
Router(config-ipx-router)# network all
Router(config-ipx-router)# interface vlan 100
Router(config-if)# ipx network 100 encapsulation snap
Router(config-if)# no shutdown
Router(config-if)# end
Router# copy running-config startup-config
```

Configuring AppleTalk Routing, Cable Ranges, and Zones

For complete information and procedures, refer to these publications:

- *Cisco IOS AppleTalk and Novell IPX Configuration Guide*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fatipx_c/index.htm
- *Cisco IOS AppleTalk and Novell IPX Command Reference*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fatipx_r/index.htm

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/univercd/cc/td/doc/product/software/ios122/122cgcr/fapolo_c/index.htm

- *Cisco IOS Apollo Domain, VINES, DECnet, ISO CLNS, and XNS Command Reference*, Release 12.2, at this URL:

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



Configuring UDE and UDLR

This chapter describes how to configure unidirectional Ethernet (UDE) and unidirectional link routing (UDLR) on the Catalyst 6500 series switch. Release 12.2(18)SXE and later releases support UDE and UDLR.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

These sections describe UDE and UDLR:

- [Understanding UDE and UDLR, page 24-1](#)
- [Configuring UDE and UDLR, page 24-3](#)

Understanding UDE and UDLR

These sections describe UDE and UDLR:

- [UDE and UDLR Overview, page 24-1](#)
- [Supported Hardware, page 24-2](#)
- [Understanding UDE, page 24-2](#)
- [Understanding UDLR, page 24-3](#)

UDE and UDLR Overview

Routing protocols support unidirectional links only if the unidirectional links emulate bidirectional links because routing protocols expect to send and receive traffic through the same interface.

Unidirectional links are advantageous because when you transmit mostly unacknowledged unidirectional high-volume traffic (for example, a video broadcast stream) over a high-capacity full-duplex bidirectional link, you use both the link from the source to the receiver and the equally high-capacity reverse-direction link, called the “back channel,” that carries the few acknowledgements from the receiver back to the source.

UDE and UDLR support use of a high-capacity unidirectional link for the high-volume traffic without consuming a similar high-capacity link for the back channel. UDE provides a high-capacity unidirectional link. UDLR provides the back channel through a tunnel that is configured over a regular-capacity link, and also provides bidirectional link emulation by transparently making the back channel appear to be on the same interface as the high-capacity unidirectional link.

Supported Hardware

On Catalyst 6500 series switches, UDE and UDLR are supported on the interfaces of these switching modules:

- WS-X6704-10GE 4-port 10-Gigabit Ethernet
- WS-X6816-GBIC 16-port Gigabit Ethernet
- WS-X6516A-GBIC 16-port Gigabit Ethernet
- WS-X6516-GBIC 16-port Gigabit Ethernet

Understanding UDE

These sections describe UDE:

- [UDE Overview, page 24-2](#)
- [Understanding Hardware-Based UDE, page 24-2](#)
- [Understanding Software-Based UDE, page 24-3](#)

UDE Overview

On Catalyst 6500 series switches, you can implement UDE with hardware or in software. Hardware-based UDE and software-based UDE both use only one strand of fiber instead of the two strands of fiber required by bidirectional traffic.

The unidirectional transceiver determines whether hardware-based UDE is receive-only or transmit-only. You can configure software-based UDE as either transmit-only or receive-only.

You do not need to configure software-based UDE on ports where you implement hardware-based UDE.

**Note**

Refer to the [“Supported Hardware” section on page 24-2](#) for a list of the module with interfaces that support hardware-based UDE and software-based UDE.

Understanding Hardware-Based UDE

You can create a unidirectional link by using a unidirectional transceiver. Unidirectional transceivers are less expensive than bidirectional transceivers. Release 12.2(18)SX and later releases support these 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 24-3](#)
- [Configuring UDLR, page 24-6](#)

**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 24-4](#)
- [Configuring Hardware-Based UDE, page 24-4](#)
- [Configuring Software-Based UDE, page 24-5](#)

UDE Configuration Guidelines

When configuring UDE, follow these guidelines:

- UDE is supported on the Supervisor Engine 720. UDE is not supported on the Supervisor Engine 2.
- STP cannot prevent Layer 2 loops in topologies that include unidirectional links.
- Send-only ports always transition to the STP forwarding state, because send-only ports never receive BPDUs.
- Receive-only ports cannot send BPDUs.
- Unidirectional ports do not support any features or protocols that require negotiation with the port at the other end of the link, including these:
 - Speed and duplex mode autonegotiation
 - Link negotiation
 - IEEE 802.3Z flow control
 - Dynamic trunking protocol (DTP)

You must manually configure the parameters that are typically controlled by Layer 2 protocols.

- A topology that includes unidirectional links only supports the VLAN Trunking Protocol (VTP) when the VTP server can send VTP frames to all switches in the VTP domain.
- Disable VTP pruning on switches that have send-only ports, because VTP pruning depends on a bidirectional exchange of information.
- Unidirectional EtherChannels cannot support PAgP or LACP. To create a unidirectional EtherChannel, you must configure the EtherChannel “on” mode.
- You can configure software-based UDE on the physical ports in an EtherChannel. You cannot configure software-based UDE on any nonphysical interfaces (for example, port-channel interfaces).
- When you implement hardware-based UDE on a port or configure software-based UDE on a port, UDLD is automatically disabled on the port.
- CDP sends CDP frames from send-only ports and receives CDP frames from receive-only ports, which means that the switch on the send-only side of a unidirectional link never receives CDP information.
- SPAN does not restrict configuration of unidirectional ports as sources or destinations.
 - Send-only ports can be SPAN destinations.
 - Receive-only ports can be SPAN sources.
- Unidirectional ports do not support IEEE 802.1X port-based authentication.
- IGMP snooping does not support topologies where there are unidirectional links between the switch and the hosts that are receiving multicast traffic.
- Configure UDLR with UDE to support communication over unidirectional links between IGMP snooping on the switch and a multicast router.
- Unidirectional links do not support ARP.

Configuring Hardware-Based UDE

There are no software configuration procedures required to support hardware-based UDE. Install a unidirectional transceiver to implement hardware-based UDE.

To verify hardware-based UDE on a port, perform this task:

Command	Purpose
Router# show interfaces [{ gigabitethernet tengigabitethernet } <i>slot/interface</i>] 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
Gil/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 } <i>slot/interface</i>]	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 } <i>slot/interface</i>] 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 13 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 13 port unidirectional mode will automatically disable ip routing on the port. You must manually configure static ip route and arp entry in order to route ip traffic.

This example shows how to verify the configuration:

```
Router> show interface tengigabitethernet 1/1 unidirectional
Unidirectional configuration mode: send only
CDP neighbour unidirectional configuration mode: receive only
```

This example shows how to disable UDE on 10 Gigabit Ethernet interface 1/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface tengigabitethernet 1/1
Router(config-if)# no unidirectional
Router(config-if)# end
```

This example shows the result of entering the **show interface** command for a port that does not support unidirectional Ethernet:

```
Router# show interface fastethernet 6/1 unidirectional
Unidirectional Ethernet is not supported on FastEthernet6/1
```

Configuring UDLR

These sections describe how to configure UDLR:

- [UDLR Back-Channel Tunnel Configuration Guidelines, page 24-6](#)
- [Configuring a Receive-Only Tunnel Interface for a UDE Send-Only Port, page 24-7](#)
- [Configuring a Send-Only Tunnel Interface for a UDE Receive-Only Port, page 24-7](#)

UDLR Back-Channel Tunnel Configuration Guidelines

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

- The PFC3 does not provide hardware support for UDLR back-channel tunnels. The MSFC3 supports UDLR back-channel tunnels in software.
- Configure a UDLR back-channel tunnel for each unidirectional link.
- On UDE send-only interfaces, configure the UDLR back-channel tunnel interface to receive.
- On UDE receive-only interfaces, configure the UDLR back-channel tunnel interface to send.
- You must configure IPv4 addresses on UDLR back-channel tunnel interfaces.
- You must configure source and destination IPv4 addresses on UDLR back-channel tunnel interfaces.
- The UDLR back-channel tunnel default mode is GRE.
- UDLR back-channel tunnels do not support IPv6 or MPLS.

Configuring a Receive-Only Tunnel Interface for a UDE Send-Only Port

To configure a receive-only tunnel interface for a UDE send-only port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface tunnel <i>number</i>	Selects the tunnel interface.
Step 2	Router(config-if)# tunnel udlr receive-only <i>ude_send_only_port</i>	Associates the tunnel receive-only interface with the UDE send-only port.
Step 3	Router(config-if)# ip address <i>ipv4_address</i>	Configures the tunnel IPv4 address.
Step 4	Router(config-if)# tunnel source { <i>ipv4_address</i> <i>type number</i> }	Configures the tunnel source.
Step 5	Router(config-if)# tunnel destination { <i>hostname</i> <i>ipv4_address</i> }	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 <i>number</i>	Selects the tunnel interface.
Step 2	Router(config-if)# tunnel udlr send-only <i>ude_receive_only_port</i>	Associates the tunnel send-only interface with the UDE receive-only port.
Step 3	Router(config-if)# ip address <i>ipv4_address</i>	Configures the tunnel IPv4 address.
Step 4	Router(config-if)# tunnel source { <i>ipv4_address</i> <i>type number</i> }	Configures the tunnel source.
Step 5	Router(config-if)# tunnel destination { <i>hostname</i> <i>ipv4_address</i> }	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
```




Configuring PFC3BXL and PFC3B Mode Multiprotocol Label Switching

This chapter describes how to configure PFC3BXL and PFC3B mode Multiprotocol Label Switching (MPLS) on a Catalyst 6500 series switch.



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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:

- [PFC3BXL and PFC3B Mode MPLS Label Switching, page 25-1](#)
- [PFC3BXL or PFC3B Mode VPN Switching, page 25-9](#)
- [Any Transport over MPLS, page 25-13](#)

PFC3BXL and PFC3B Mode MPLS Label Switching

These sections describe PFC3BXL and PFC3B mode MPLS label switching:

- [Understanding MPLS, page 25-2](#)
- [Understanding PFC3BXL and PFC3B Mode MPLS Label Switching, page 25-2](#)
- [Supported Hardware Features, page 25-4](#)
- [Supported Cisco IOS Features, page 25-5](#)
- [MPLS Guidelines and Restrictions, page 25-7](#)
- [Configuring MPLS, page 25-7](#)
- [MPLS Per-Label Load Balancing, page 25-7](#)
- [MPLS Configuration Examples, page 25-8](#)

Understanding MPLS

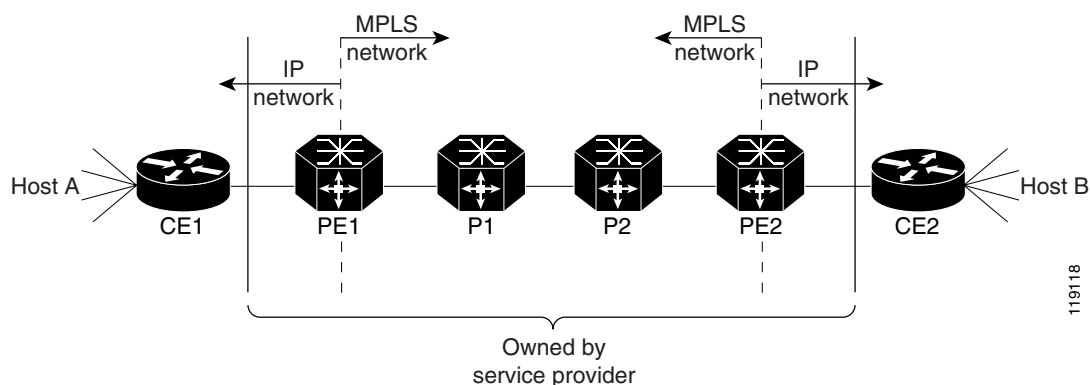
MPLS uses label switching to forward packets over various link-level technologies such as Packet-over-SONET (POS), Frame Relay, ATM, and Ethernet. Labels are assigned to packets based on groupings or forwarding equivalence classes (FECs). The label is added between the Layer 2 and the Layer 3 header.

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 25-1 shows an MPLS network of a service provider that connects two sites of a customer network.

Figure 25-1 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 PFC3BXL and PFC3B Mode MPLS Label Switching

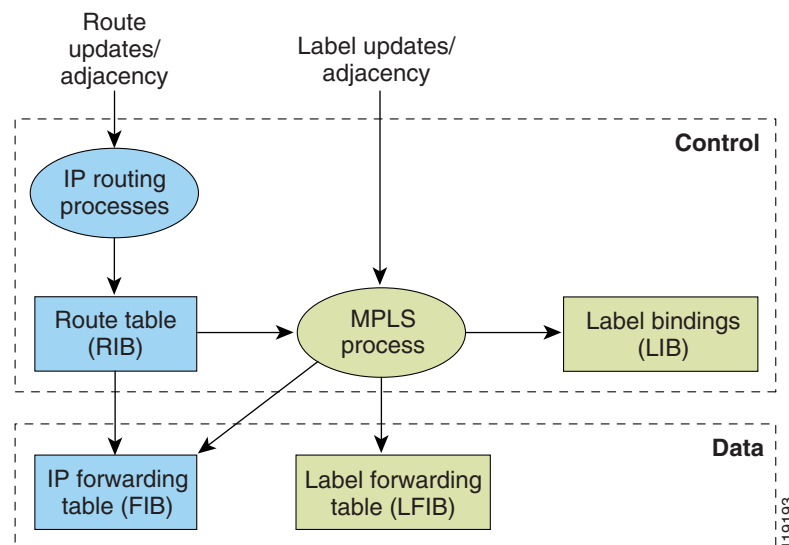
A Supervisor Engine 720 in PFC3BXL or PFC3B mode supports Layer 3 Multiprotocol Label Switching (MPLS) virtual private networks (VPNs), and Layer 2 Ethernet over MPLS (EoMPLS), with quality of service (QoS) and security.

The MSFC3 on the Supervisor Engine 720 performs Layer 3 control-plane functions, including address resolution and routing protocols. The MSFC3 processes information from the Routing and Label Distribution Protocols and builds the IP forwarding (FIB) table and the label forwarding (LFIB) table. The MSFC3 distributes the information in both tables to the PFC3BXL or PFC3B.

The PFC3BXL or PFC3B 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 25-2 shows the various functional blocks that support PFC3BXL and PFC3B mode 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 25-2 MPLS Forwarding, Control and Data Planes



IP to MPLS

At the ingress to the MPLS network, the PFC3BXL or PFC3B examines the IP packets and performs a route lookup in the FIB TCAM. The lookup result is the pointer to a particular adjacency entry. The adjacency entry contains the appropriate information for label pushing (for IP to MPLS path) and encapsulation. The PFC3BXL or PFC3B generates a result containing the imposition label(s) needed to switch the MPLS packet.



Note

If MPLS load sharing is configured, the adjacency may point to a load-balanced path. See [“Basic MPLS Load Balancing”](#) section on page 25-8.

MPLS to MPLS

At the core of an MPLS network, the PFC3BXL or PFC3B uses the topmost label to perform a lookup in the FIB TCAM. The successful lookup points to an adjacency that swaps the top label in the packet with a new label as advertised by the downstream label switch router (LSR). If the router is the penultimate hop LSR router (the upstream LSR next to the egress LER), the adjacency instructs the PFCBXL to pop the topmost label, resulting in either an MPLS packet with the remaining label for any VPN or AToM use or a native IP packet.

MPLS to IP

At the egress of the MPLS network there are several possibilities.

For a native IP packet (when the penultimate router has popped the label), the PFC3BXL or PFC3B performs a route lookup in the FIB TCAM.

For a MPLS VPN packet, after the Interior Gateway Protocol (IGP) label is popped at penultimate router, the VPN label remains. The operation that the PFC3BXL or PFC3B performs depends on the VPN label type. Packets carrying aggregate labels require a second lookup based on the IP header after popping the aggregate label. For a nonaggregate label, the PFC3BXL or PFC3B performs a route lookup in the FIB TCAM to obtain the IP next hop information.

For the case of a packet with an IGP label and a VPN label, when there is no penultimate hop popping (PHP), the packet carries the explicit-null label on top of the VPN label. The PFC3BXL or PFC3B looks up the top label in the FIB TCAM and recirculates the packet. Then the PFC3BXL or PFC3B handles the remaining label as described in the preceding paragraph, depending on whether it is an aggregate or nonaggregate label.

Packets with the explicit-null label for the cases of EoMPLS, MPLS, and MPLS VPN an MPLS are handled the same way.

MPLS VPN Forwarding

There are two types of VPN labels: aggregate labels for directly connected network or aggregate routes, and nonaggregate labels. Packets carrying aggregate labels require a second lookup based on the IP header after popping the aggregate label. The VPN information (VPN-IPv4 address, extended community, and label) is distributed through the Multiprotocol-Border Gateway Protocol (MP-BGP).

Recirculation

In certain cases, the PFC3BXL or PFC3B provides the capability to recirculate the packets. Recirculation can be used to perform additional lookups in the ACL or QoS TCAMs, the Netflow table, or the FIB TCAM table. Recirculation is necessary in these situations:

- To push more than three labels on imposition
- To pop more than two labels on disposition
- To pop an explicit null top label
- When the VPN Routing and Forwarding (VRF) number is more than 511
- For IP ACL on the egress interface (for nonaggregate (per-prefix) labels only)

Packet recirculation occurs only on a particular packet flow; other packet flows are not affected. The rewrite of the packet occurs on the modules; the packets are then forwarded back to the PFC3BXL or PFC3B for additional processing.

Supported Hardware Features

The following hardware features are supported:

- Label operation— Any number of labels can be pushed or popped, although for best results, up to three labels can be pushed, and up to two labels can be popped in the same operation.
- IP to MPLS path—IP packets can be received and sent to the MPLS path.
- MPLS to IP path—Labeled packets can be received and sent to the IP path.
- MPLS to MPLS path—Labeled packets can be received and sent to the label path.

- MPLS Traffic Engineering (MPLS TE)—Enables an MPLS backbone to replicate and expand the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks.
- Time to live (TTL) operation—At the ingress edge of the MPLS network, the TTL value in the MPLS frame header can be received from either the TTL field of the IP packet header or the user-configured value from the adjacency entry. At the egress of the MPLS network, the final TTL equals the minimum (label TTL and IP TTL)-1.



Note With the Uniform mode, the TTL is taken from the IP TTL; with the Pipe mode, a value of 255, taken from the hardware register, is used for the outgoing label.

- QoS—Information on Differentiated Services (DiffServ) and ToS from IP packets can be mapped to MPLS EXP field.
- MPLS/VPN Support—Up to 1024 VRFs can be supported (over 511 VRFs requires recirculation).
- Ethernet over MPLS—The Ethernet frame can be encapsulated at the ingress to the MPLS domain and the Ethernet frame can be decapsulated at the egress.
- Packet recirculation—The PFC3BXL or PFC3B provides the capability to recirculate the packets. See the “[Recirculation](#)” section on page 25-4.
- 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 in PFC3BXL or PFC3B mode:



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

- Multi-VRF for CE Routers (VRF Lite)—VRF-lite is a feature that enables a service provider to support two or more VPNs (using only VRF-based IPv4), where IP addresses can be overlapped among the VPNs. See this publication:
http://www.cisco.com/en/US/products/hw/routers/ps259/prod_bulletin09186a00800921d7.html.
- MPLS on Cisco routers—This feature provides basic MPLS support for imposing and removing labels on IP packets at label edge routers (LERs) and switching labels at label switch routers (LSRs). See this publication:
http://www.cisco.com/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/fsdse rv3.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/fstef a_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/fsiar ea3.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/120st2 1/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/ftsc8.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/ftsc113. 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/ftias113.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 25-13.

MPLS Guidelines and Restrictions

When configuring PFC3BXL or PFC3B MPLS, follow these guidelines and restrictions:

- PFC3BXL or PFC3B mode supports up to 8 load-shared paths. Cisco IOS releases for other platforms support only 8 load-shared paths.
- PFC3BXL or PFC3B mode supports MTU checking and fragmentation.

**Note**

Fragmentation is supported with software (for IP to MPLS path). See the **mtu** command in the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX.

**Note**

For information on other limitations and restrictions, see the “MPLS VPN Guidelines and Restrictions” section on page 25-11 and the “EoMPLS Guidelines and Restrictions” section on page 25-14.

PFC3BXL and PFC3B Mode MPLS Supported Commands

PFC3BXL and PFC3B mode MPLS supports these commands:

- **mpls ip default route**
- **mpls ip propagate-ttl**
- **mpls ip ttl-expiration pop**
- **mpls label protocol**
- **mpls label range**
- **mpls ip**
- **mpls label protocol**
- **mpls mtu**

For information about these commands, see these publications:

http://www.cisco.com/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://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/xcftagc.htm

MPLS Per-Label Load Balancing

The following sections provide information on basic MPLS, MPLS Layer 2 VPN, and MPLS Layer 3 VPN load balancing.

Basic MPLS Load Balancing

The maximum number of load balancing paths is 8. The PFC3BXL or PFC3B forwards MPLS labeled packets without explicit configuration. If the packet has three labels or less and the underlying packet is IPv4, then the PFC3BXL or PFC3B uses the source and destination IPv4 address. If the underlying packet is not IPv4 or more than three labels are present, the PFC3BXL or PFC3B parses down as deep as the fifth or lowest label and uses it for hashing.

MPLS Layer 2 VPN Load Balancing

Load balancing is based on the VC label in the MPLS core if the first nibble of the MAC address in the customer Ethernet frame is not 4.



Note

Load balancing is not supported at the ingress PE for Layer 2 VPNs.

MPLS Layer 3 VPN Load Balancing

MPLS Layer 3 VPN load balancing is similar to basic MPLS load balancing. For more information, see the [“Basic MPLS Load Balancing” section on page 25-8](#).

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

```

PFC3BXL or PFC3B Mode VPN Switching

These sections describe PFC3BXL or PFC3B mode VPN switching:

- [PFC3BXL or PFC3B Mode VPN Switching Operation, page 25-10](#)
- [MPLS VPN Guidelines and Restrictions, page 25-11](#)
- [PFC3BXL or PFC3B Mode MPLS VPN Supported Commands, page 25-11](#)
- [MPLS VPN Sample Configuration, page 25-12](#)

PFC3BXL or PFC3B Mode VPN Switching Operation

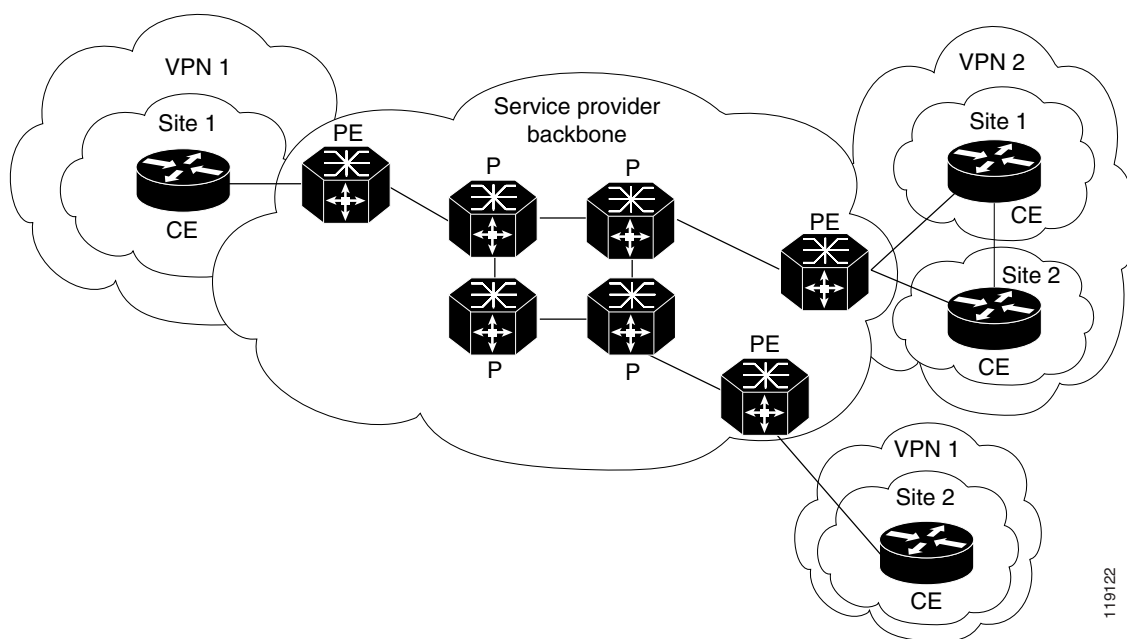
The IP VPN feature for MPLS allows a Cisco IOS network to deploy scalable IP Layer 3 VPN backbone services to multiple sites deployed on a shared infrastructure while also providing the same access or security policies as a private network. VPN based on MPLS technology provides the benefits of routing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

Refer to the Cisco IOS software documentation for a conceptual MPLS VPN overview and configuration details at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/index.htm

A typical MPLS VPN network topology is shown in Figure 25-3.

Figure 25-3 VPNs with Service Provider Backbone



At the ingress PE, the PFC3BXL or PFC3B makes a forwarding decision based on the packet headers. The PFC3BXL or PFC3B contains a table that maps VLANs to VPNs. In the Catalyst 6500 series switch architecture, all physical ingress interfaces in the system are associated with a specific VPN. The PFC3BXL or PFC3B looks up the IP destination address in the CEF table but only against prefixes that are in the specific VPN. (The table entry points to a specific set of adjacencies and one is chosen as part of the load-balancing decision if multiple parallel paths exist.)

The table entry contains the information on the Layer 2 header that the packet needs, as well as the specific MPLS labels to be pushed onto the frame. The information to rewrite the packet goes back to the ingress line card where it is rewritten and forwarded to the egress line interface.

VPN traffic is handled at the egress from the PE based upon the per-prefix labels or aggregate labels. If per-prefix labels are used, then each VPN prefix has a unique label association; this allows the PE to forward the packet to the final destination based upon a label lookup in the FIB.



Note

The PFC3BXL or PFC3B allocates only one aggregate label per VRF.

If aggregate labels are used for disposition in an egress PE, many prefixes on the multiple interfaces may be associated with the label. In this case, the PFC3BXL or PFC3B must perform an IP lookup to determine the final destination. The IP lookup may require recirculation.

MPLS VPN Guidelines and Restrictions

When configuring MPLS VPN, follow these guidelines and restrictions:

- PFC3BXL or PFC3B mode supports a total of 1024 VRFs per chassis with enhanced OSMs; using a nonenhanced OSM causes the system to default to 511 VRFs.
- In PFC3BXL or PFC3B mode, VPNs are recirculated when the number of VPNs is over 511.

PFC3BXL or PFC3B Mode MPLS VPN Supported Commands

PFC3BXL or PFC3B mode MPLS VPN supports these commands:

- **address-family**
- **exit-address-family**
- **import map**
- **ip route vrf**
- **ip route forwarding**
- **ip vrf**
- **neighbor activate**
- **rd**
- **route-target**

For information about these commands, see these publications:

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

Configuring MPLS VPN

For information on configuring MPLS VPN, refer to the *MPLS Virtual Private Networks* feature module at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt3/xcftagc.htm#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 FlexWAN CE-facing interfaces. PFC3BXL or PFC3B mode 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 in PFC3BXL or PFC3B mode:

- Ethernet over MPLS (EoMPLS) (VLAN mode and port mode)
- Frame Relay over MPLS with DLCI-to-DLCI connections
- ATM AAL5 over MPLS
- ATM Cell Relay over MPLS



Note Additional AToM types are planned in future releases.

PFC3BXL or PFC3B mode supports both hardware-based EoMPLS as well as OSM-, FlexWAN, or FlexWAN2-based EoMPLS. For more information, see this publication:

<http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/optical/122sx/mpls.htm#wp1128955>

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

<http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/optical/122sx/mpls.htm#wp1270404>

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

<http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/optical/122sx/mpls.htm#wp1279824>

These sections describe AToM:

- [AToM Load Balancing, page 25-14](#)
- [Understanding EoMPLS, page 25-14](#)
- [EoMPLS Guidelines and Restrictions, page 25-14](#)
- [Configuring EoMPLS, page 25-16](#)

AToM Load Balancing

PFC3BXL or PFC3B mode EoMPLS does not support load balancing at the tunnel ingress; only one Interior Gateway Protocol (IGP) path is selected even if multiple IGP paths are available, but load balancing is available at the MPLS core.

Understanding EoMPLS

EoMPLS is one of the AToM transport types. AToM transports Layer 2 packets over a MPLS backbone using a directed LDP session between edge routers for setting up and maintaining connections. Forwarding occurs through the use of two level labels that provide switching between the edge routers. The external label (tunnel label) routes the packet over the MPLS backbone to the egress PE at the ingress PE. The VC label is a demuxing label that determines the connection at the tunnel endpoint (the particular egress interface on the egress PE as well as the VLAN identifier for an Ethernet frame).

EoMPLS works by encapsulating Ethernet PDUs in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet.



Note

Use OSM-based EoMPLS when you want local Layer 2 switching and EoMPLS on the same VLAN. You need to configure EoMPLS on the SVI; the core-facing card must be an OSM. When local Layer 2 switching is not required, use PFC-based EoMPLS configured on the subinterface or physical interface.

EoMPLS Guidelines and Restrictions

When configuring EoMPLS, follow these guidelines and restrictions:

- Ensure that the maximum transmission unit (MTU) of all intermediate links between endpoints is sufficient to carry the largest Layer 2 packet received.
- EoMPLS supports VLAN packets that conform to the IEEE 802.1Q standard. The 802.1Q specification establishes a standard method for inserting VLAN membership information into Ethernet frames.
- If QoS is disabled globally, both the 802.1p and IP precedence bits are preserved. When the QoS is enabled on a Layer 2 port, either 802.1q P bits or IP precedence bits can be preserved with the trusted configuration. However, by default the unpreserved bits are overwritten by the value of preserved bits. For instance, if you preserve the P bits, the IP precedence bits are overwritten with the value of the P bits. PFC3BXL or PFC3B mode provides a new command that allows you to trust the P bits while preserving the IP precedence bits. To preserve the IP precedence bits, use the **no mls qos rewrite ip dscp** command.

**Note**

The **no mls qos rewrite ip dscp** command is not compatible with the MPLS and MPLS VPN features. See [Chapter 42, “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 switch. Otherwise, the BPDUs are directed to the supervisor engine and not to the EoMPLS cloud.
 - The native VLAN of a trunk must not be configured as an EoMPLS VLAN.
- In PFC3BXL or PFC3B mode, 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 PFC3BXL or PFC3B does not perform any Layer 2 lookup to determine if the destination MAC address resides on the local or remote segment and does not perform any Layer 2 address learning (as traditional LAN bridging does). This functionality (local switching) is available only when using OSM and FlexWAN modules as uplinks.
- In previous releases of AToM, the command used to configure AToM circuits was **mpls l2 transport route**. This command has been replaced with the **xconnect** command. You can use the **xconnect** command to configure EoMPLS circuits.
- The AToM control word is not supported.
- EoMPLS is not supported on Layer 3 VLAN interfaces.
- Point-to-point EoMPLS works with a physical interface and subinterfaces.

Configuring EoMPLS

These sections describe how to configure EoMPLS:

- [Prerequisites, page 25-16](#)
- [Configuring PFC3BXL or PFC3B Mode VLAN-Based EoMPLS, page 25-17](#)
- [Configuring PFC3BXL or PFC3B Mode Port-Based EoMPLS, page 25-20](#)

Prerequisites

Before you configure EoMPLS, ensure that the network is configured as follows:

- Configure IP routing in the core so that the PE routers can reach each other through IP.
- Configure MPLS in the core so that a label switched path (LSP) exists between the PE routers.

EoMPLS works by encapsulating Ethernet PDUs in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet. Two methods are available to configure EoMPLS in PFC3BXL or PFC3B mode:

- 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, PFC3BXL and PFC3B mode EoMPLS does not allow local switching of packets between interfaces unless you use loopback ports.
- A system can have both an OSM or FlexWAN configuration and PFC3BXL or PFC3B 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 FlexWAN-enabled interfaces, OSM or FlexWAN-based EoMPLS connections will not be active; this causes packets for OSM or FlexWAN-based EoMPLS arriving on non-WAN interfaces to be dropped. For information on WAN (FlexWAN and OSM) EoMPLS, see this publication:

<http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/optical/122sx/mpls.htm#wp1128955>

PFC3BXL or PFC3B mode supports MPLS. In PFC3BXL or PFC3B mode, LAN ports can receive Layer 2 traffic, impose labels, and switch the frames into the MPLS core without using an OSM or FlexWAN module.

In PFC3BXL or PFC3B mode, you can configure an OSM or a FlexWAN module to face the core of MPLS network and use either the OSM configuration, the FlexWAN configuration, or the PFC3BXL or PFC3B mode configuration. For more information, see this publication:

<http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/optical/122sx/mpls.htm#wp1128955>

Configuring PFC3BXL or PFC3B Mode VLAN-Based EoMPLS

When configuring PFC3BXL or PFC3B mode VLAN-based EoMPLS, follow these guidelines and restrictions:

- The AToM control word is not supported.
- Ethernet packets with hardware-level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.
- You must configure VLAN-based EoMPLS on subinterfaces.

To configure PFC3BXL or PFC3B mode VLAN-based EoMPLS, perform this task on the provider edge (PE) routers.

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# interface gigabitethernet <i>slot/interface.subinterface</i>	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 <i>vlan_id</i>	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 <i>peer_router_id vcid</i> encapsulation mpls	Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports.

This is a PFC3BXL or PFC3B mode VLAN-based EoMPLS configuration sample:

```
!
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/rcv
    LDP Id: 12.12.12.12:0
Targeted Hellos:
  13.13.13.13 -> 11.11.11.11 (ldp): active/passive, xmit/rcv
    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 \
20     Untagged   12ckt(2)       133093    V12          point2point
21     Untagged   12ckt(3)       185497    V13          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 PFC3BXL or PFC3B Mode Port-Based EoMPLS

When configuring PFC3BXL or PFC3B mode port-based EoMPLS, follow these guidelines and restrictions:

- The AToM control word is not supported.
- Ethernet packets with hardware-level cyclic redundancy check (CRC) errors, framing errors, and runt packets are discarded on input.
- Port-based EoMPLS and VLAN-based EoMPLS are mutually exclusive. If you enable a main interface for port-to-port transport, you also cannot enter commands on a subinterface.

To support 802.1Q-in-802.1Q traffic and Ethernet traffic over EoMPLS in PFC3BXL or PFC3B mode, 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 <i>slot/interface</i>	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 <i>peer_router_id vcid</i> 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 negotiate
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  \
20     Untagged    12ckt(2)         55146580   V12         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
V12	Eth VLAN 2	11.11.11.11	2	UP



Configuring IPv4 Multicast VPN Support

This chapter describes how to configure IPv4 Multicast Virtual Private Network (MVPN) support on Catalyst 6500 series switches. Release 12.2(18)SXE and later releases support MVPN when the switch is operating in PFC3B mode or PFC3BXL mode.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter contains these sections:

- [Understanding How MVPN Works, page 26-1](#)
- [MVPN Configuration Guidelines and Restrictions, page 26-7](#)
- [Configuring MVPN, page 26-8](#)

Understanding How MVPN Works

These sections describe MVPN:

- [MVPN Overview, page 26-1](#)
- [Multicast Routing and Forwarding and Multicast Domains, page 26-2](#)
- [Multicast Distribution Trees, page 26-2](#)
- [Multicast Tunnel Interfaces, page 26-5](#)
- [PE Router Routing Table Support for MVPN, page 26-6](#)
- [Multicast Distributed Switching Support, page 26-6](#)
- [Hardware-Assisted IPv4 Multicast, page 26-6](#)

MVPN Overview

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

1. If a particular multicast stream exceeds the defined threshold, the sending PE router dynamically creates a data MDT for that particular multicast traffic.
2. The sending PE router then transmits a DATA-MDT JOIN request (which is a User Datagram Protocol (UDP) message to port 3232) to the other PE routers, informing them of the new data MDT.
3. Receiving PE routers examine their VRF routing tables to determine if they have any customers interested in receiving this data stream. If so, they use the PIM protocol to transmit a PIM JOIN message for this particular data MDT group (in the global table PIM instance) to accept the stream. Routers that do not currently have any customers for this stream still cache the information, in case any customers request it later on.
4. Three seconds after sending the DATA-MDT JOIN message, the sending PE router removes the high-bandwidth multicast stream from the default MDT and begins transmitting it over the new data MDT.
5. The sending PE router continues to send a DATA-MDT JOIN message every 60 seconds, as long as the multicast stream continues to exceed the defined threshold. If the stream falls below the threshold for more than 60 seconds, the sending PE router stops sending the DATA-MDT JOIN messages, and moves the stream back to the default MDT.
6. Receiving routers age out the cache information for the default MDT when they do not receive a DATA-MDT JOIN message for more than three minutes.

Data MDTs allow for high-bandwidth sources inside the VPN while still ensuring optimal traffic forwarding in the MPLS VPN core.

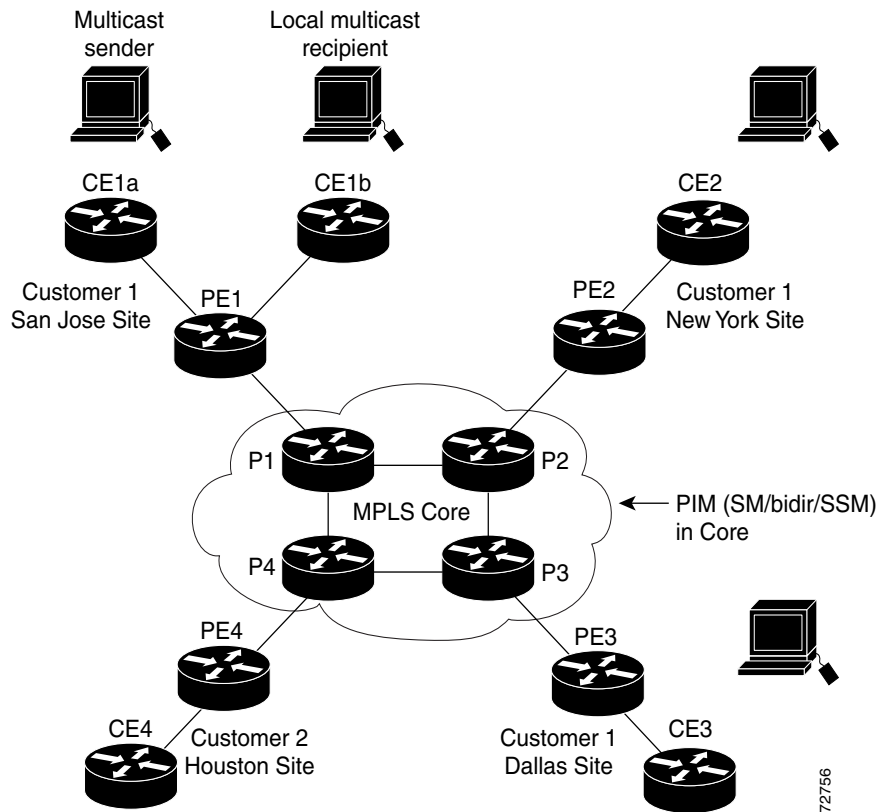
**Note**

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

In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. The San Jose site is transmitting a one-way multicast presentation. The service provider network supports all three sites associated with this customer, in addition to the Houston site of a different enterprise customer.

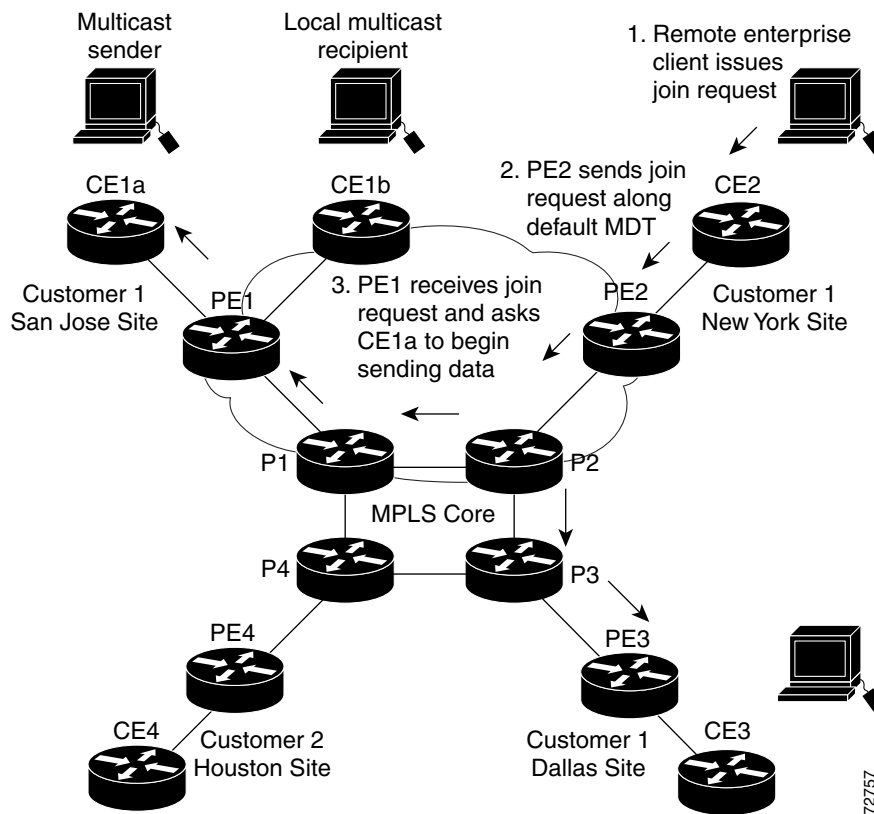
The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. Although PE4 is interconnected to these other routers in the MPLS core, PE4 is associated with a different customer and is therefore not part of the default MDT.

Figure 26-1 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 26-1 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 26-2](#) shows how the PE router forwards the request to the CE router associated with the multicast source (CE1a).

72756

Figure 26-2 *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

PFC3BXL or PFC3B mode 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, PFC3BXL or PFC3B mode hardware acceleration supports multicast traffic in PIM dense, PIM sparse, PIM bidirectional, and PIM Source Specific Multicast (SSM) modes.

In the service provider core, PFC3BXL or PFC3B mode hardware acceleration supports multicast traffic in PIM sparse, PIM bidirectional, and PIM SSM modes. In the service provider core, PFC3BXL or PFC3B mode hardware acceleration does not support multicast traffic in PIM dense mode.

MVPN Configuration Guidelines and Restrictions

When configuring MVPN, follow these guidelines and restrictions:

- Release 12.2(18)SXE and later releases support MVPN when the switch is operating in PFC3B mode or PFC3BXL mode. Supervisor Engine 2 does not support MVPN.
- All PE routers in the multicast domain need to be running a Cisco IOS software image that supports the MVPN feature. There is no requirement for MVPN support on the P and CE routers.
- Support for IPv4 multicast traffic must also be enabled on all backbone routers.
- The Border Gateway Protocol (BGP) routing protocol must be configured and operational on all routers supporting multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.
- Only ingress replication is supported when MVPN is configured. If the switch is currently configured for egress replication, it is forced into ingress replication when the first MVRP is configured.
- When the switch is acting as a PE, and receives a multicast packet from a customer router with a time-to-live (TTL) value of 2, it drops the packet instead of encapsulating it and forwarding it across the MVPN link. Because such packets would normally be dropped by the PE at the other end of the MVPN link, this does not affect traffic flow.
- If the core multicast routing uses SSM, then the data and default multicast distribution tree (MDT) groups must be configured within the SSM range of IPv4 addresses.
- The update source interface for the BGP peerings must be the same for all BGP peerings configured on the router in order for the default MDT to be configured properly. If you use a loopback address for BGP peering, then PIM sparse mode must be enabled on the loopback address.
- The **ip mroute-cache** command must be enabled on the loopback interface used as the BGP peering interface in order for distributed multicast switching to function on the platforms that support it. The **no ip mroute-cache** command must *not* be present on these interfaces.
- Data MDTs are not created for VRF PIM dense mode multicast streams because of the flood and prune nature of dense mode multicast flows and the resulting periodic bring-up and tear-down of such data MDTs.
- Data MDTs are not created for VRF PIM bidirectional mode because source information is not available.
- MVPN does not support multiple BGP peering update sources, and configuring them can break MVPN RPF checking. The source IPv4 address of the MVPN tunnels is determined by the highest IPv4 address used for the BGP peering update source. If this IPv4 address is not the IPv4 address used as the BGP peering address with the remote PE router, MVPN will not function properly.
- MDT tunnels do not carry unicast traffic.
- Although MVPN uses the infrastructure of MPLS VPN networks, you cannot apply MPLS tags or labels to multicast traffic over the VPNs.

- Each MVRF that is configured with a default MDT uses three hidden VLANs (one each for encapsulation, decapsulation, and interface), in addition to external, user-visible VLANs. This means that an absolute maximum of 1,000 MVRFs are supported on each router. (MVRFs without a configured MDT still use one internal VLAN, so unused MVRFs should be deleted to conserve VLAN allocation.)
- Because MVPN uses MPLS, MVPN supports only the RPR and RPR+ redundancy modes. In Release 12.2(18)SXD and rebuilds, MPLS can coexist with NSF with SSO redundancy mode, but there is no support for stateful MPLS switchover.
- If your MPLS VPN network already contains a network of VRFs, you do not need to delete them or recreate them to be able to support MVRF traffic. Instead, configure the **mdt default** and **mdt data** commands, as listed in the following procedure, to enable multicast traffic over the VRF.
- BGP should be already configured and operational on all routers that are sending or receiving multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.
- The same MVRF must be configured on each PE router that is to support a particular VPN connection.
- Each PE router that supports a particular MVRF must be configured with the same **mdt default** command.
- The switch supports only ingress replication when MVPN is configured. If a switch is currently configured for egress replication, it is forced into ingress replication when the first MVRF is configured. If a switch is currently configured for egress replication, we recommend performing this task only during scheduled maintenance periods, so that traffic disruption can be kept to a minimum.

Configuring MVPN

These sections describe how to configure MVPN:

- [Forcing Ingress Multicast Replication Mode \(Optional\), page 26-8](#)
- [Configuring a Multicast VPN Routing and Forwarding Instance, page 26-9](#)
- [Configuring Multicast VRF Routing, page 26-15](#)
- [Configuring Interfaces for Multicast Routing to Support MVPN, page 26-20](#)



Note

These configuration tasks assume that BGP is already configured and operational on all routers that are sending or receiving the multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.

Forcing Ingress Multicast Replication Mode (Optional)

The MVPN feature supports only ingress multicast replication mode. If the switch is currently configured for egress replication, it is forced into ingress replication when the first MVRF is configured. This change in replication mode automatically purges all forwarding entries in the hardware, temporarily forcing the switch into software switching until the table entries can be rebuilt.

To avoid disrupting customer traffic, we recommend verifying that the switch is already in ingress multicast replication mode before configuring any MVRFs.

This example shows how to verify the multicast replication mode:

```
Router# show mls ip multicast capability
```

```
Current mode of replication is Ingress
auto replication mode detection is ON
```

Slot	Multicast replication capability
2	Egress
5	Egress
6	Egress
8	Ingress
9	Ingress

```
Router#
```

If the current replication mode is egress or if any of the switching modules are capable of egress replication mode, configure ingress replication mode during a scheduled maintenance period to minimize the disruption of customer traffic.

To configure ingress multicast replication mode, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# mls ip multicast replication-mode ingress	Configures ingress multicast replication mode and disables automatic detection of the replication mode (enabled by default).
	Router(config)# no mls ip multicast replication-mode ingress	Enables automatic detection of the replication mode.
Step 3	Router(config)# do show mls ip multicast capability include Current	Verifies the configuration.

This example shows how to configure ingress multicast replication mode and verify the configuration:

```
Router(config)# mls ip multicast replication-mode ingress
Router(config)# do show mls ip multicast capability | include Current
Current mode of replication is Ingress
```

Configuring a Multicast VPN Routing and Forwarding Instance

These sections describe how to configure a multicast VPN routing and forwarding (MVRF) instance for each VPN connection on each PE router that is to handle the traffic for each particular VPN connection that is to transmit or receive multicast traffic:

- [Configuring a VRF Entry, page 26-10](#)
- [Configuring the Route Distinguisher, page 26-10](#)
- [Configuring the Route-Target Extended Community, page 26-11](#)
- [Configuring the Default MDT, page 26-11](#)
- [Configuring Data MDTs \(Optional\), page 26-12](#)
- [Enabling Data MDT Logging, page 26-12](#)

- [Sample Configuration, page 26-13](#)
- [Displaying VRF Information, page 26-13](#)

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] <i>route_target_ext_community</i>	Configures a route-target extended community for the VRF.
	Router(config-vrf)# no route-target [[import export both] <i>route_target_ext_community</i>]	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 <i>group_address</i>	Configures the default MDT.
Router(config-vrf)# no mdt default	Deletes the default MDT.

When configuring the default MDT, note the following information:

- The *group_address* is the multicast IPv4 address of the default MDT group. This address serves as an identifier for the MVRF community, because all provider-edge (PE) routers configured with this same group address become members of the group, which allows them to receive the PIM control messages and multicast traffic that are sent by other members of the group.
- This same default MDT must be configured on each PE router to enable the PE routers to receive multicast traffic for this particular MVRF.

This example shows how to configure 239.1.1.1 as the default MDT:

```
Router(config-vrf)# mdt default 239.1.1.1
```

Configuring Data MDTs (Optional)

To configure optional data MDTs, perform this task:

Command or Action	Purpose
Router(config-vrf)# mdt data <i>group_address</i> <i>wildcard_bits</i> [threshold <i>threshold_value</i>] [list <i>access_list</i>]	(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 switch, use the **show ip vrf** command:

```
Router# show ip vrf
```

Name	Default RD	Interfaces
green	1:52	GigabitEthernet6/1
red	200:1	GigabitEthernet1/1
		GigabitEthernet3/16
		Loopback2

```
Router#
```

To display information about the MDTs that are currently configured for all MVRFs, use the **show ip pim mdt** command. The following example shows typical output for this command:

```
Router# show ip pim mdt
```

MDT Group	Interface	Source	VRF
* 227.1.0.1	Tunnel1	Loopback0	BIDIR01
* 227.2.0.1	Tunnel2	Loopback0	BIDIR02
* 228.1.0.1	Tunnel3	Loopback0	SPARSE01
* 228.2.0.1	Tunnel4	Loopback0	SPARSE02

**Note**

To display information about a specific tunnel interface, use the **show interface tunnel** command. The IPv4 address for the tunnel interface is the multicast group address for the default MDT of the MVRF.

To display additional information about the MDTs, use the **show mls ip multicast mdt** command. The following example shows typical output for this command:

```
Router# show mls ip multicast mdt
```

```
State: H - Hardware Installed, I - Install Pending, D - Delete Pending,
      Z - Zombie
```

VRF	MMLS 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/stoppped, RP 10.10.10.12, flags: SJCF
  Incoming interface: Tunnel1, RPF nbr 10.10.10.12, Partial-SC
  Outgoing interface list:
    GigabitEthernet3/1.3001, Forward/Sparse-Dense, 00:16:25/00:02:49, H
(6.9.0.100, 228.1.0.1), 00:14:13/00:03:29, flags: FT
  Incoming interface: GigabitEthernet3/1.3001, RPF nbr 0.0.0.0, RPF-MFD
  Outgoing interface list:
    Tunnel1, Forward/Sparse-Dense, 00:14:13/00:02:46, H
```

Router#



Note

In this example, the **show ip mroute vrf** command shows that Tunnel1 is the MDT tunnel interface (MTI) being used by this VRF.

Configuring Multicast VRF Routing

These sections describe how to configure multicast routing to support MVPN:

- [Enabling IPv4 Multicast Routing Globally, page 26-16](#)
- [Enabling IPv4 Multicast VRF Routing, page 26-16](#)
- [Configuring a PIM VRF Register Message Source Address, page 26-16](#)
- [Specifying the PIM VRF Rendezvous Point \(RP\) Address, page 26-17](#)
- [Configuring a Multicast Source Discovery Protocol \(MSDP\) Peer, page 26-17](#)
- [Enabling IPv4 Multicast Header Storage, page 26-18](#)
- [Configuring the Maximum Number of Multicast Routes, page 26-18](#)
- [Sample Configuration, page 26-19](#)
- [Displaying IPv4 Multicast VRF Routing Information, page 26-20](#)



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 <i>vrf_name</i> [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 26-9.
- **distributed**—(Optional) Enables Multicast Distributed Switching (MDS).

This example show how to enable IPv4 multicast VRF routing:

```
Router# configure terminal
Router(config)# ip multicast-routing vrf blue
```

Configuring a PIM VRF Register Message Source Address

To configure a PIM VRF register message source address, perform this task:

	Command or Action	Purpose
	Router(config)# ip pim vrf <i>vrf_name</i> register-source <i>interface_type interface_number</i>	(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 <i>vrf_name</i> 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 <i>vrf_name</i> rp-address <i>rp_address</i> [<i>access_list</i>] [override] [bidir]	Specifies the PIM RP IPv4 address for a (required for sparse PIM networks):
Router(config)# no ip pim vrf <i>vrf_name</i> rp-address <i>rp_address</i>	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 <i>vrf_name</i> peer { <i>peer_name</i> <i>peer_address</i> } [connect-source <i>interface_type</i> <i>interface_number</i>] [remote-as <i>ASN</i>]	(Optional) Configures an MSDP peer.
Router(config)# no ip msdp vrf <i>vrf_name</i> peer { <i>peer_name</i> <i>peer_address</i> } [connect-source <i>interface_type</i> <i>interface_number</i>] [remote-as <i>ASN</i>]	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 <i>vrf_name</i> cache-headers [rtp]	(Optional) Enables a circular buffer to store IPv4 multicast packet headers.
Router(config)# no ip multicast vrf <i>vrf_name</i> 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 <i>vrf_name</i> route-limit <i>limit</i> [<i>threshold</i>]	(Optional) Configures the maximum number of multicast routes that can be added for multicast traffic.
Router(config)# no ip multicast vrf <i>vrf_name</i> route-limit <i>limit</i> [<i>threshold</i>]	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 mrimfo-filter <i>access_list</i>	(Optional) Configures IPV4 multicast route filtering with an access list. The <i>access_list</i> parameter can be the name or number of a access list.
Router(config)# no ip multicast mrimfo-filter	Clears the configured maximum number of routes.

This example show how to configure IPV4 multicast route filtering:

```
Router(config)# ip multicast mrimfo-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
40.60.0.11    Tunnel96        00:00:31/00:01:13 v2      1 / S
40.50.0.11    Tunnel96        00:00:54/00:00:50 v2      1 / S
```

```
Router#
```

Configuring Interfaces for Multicast Routing to Support MVPN

These sections describe how to configure interfaces for multicast routing to support MVPN:

- [Multicast Routing Configuration Overview, page 26-20](#)
- [Configuring PIM on an Interface, page 26-20](#)
- [Configuring an Interface for IPv4 VRF Forwarding, page 26-21](#)
- [Sample Configuration, page 26-22](#)

Multicast Routing Configuration Overview

Protocol Independent Multicast (PIM) must be configured on all interfaces that are being used for IPv4 multicast traffic. In a VPN multicast environment, you should enable PIM on at least all of the following interfaces:

- Physical interface on a provider edge (PE) router that is connected to the backbone.
- Loopback interface that is used for BGP peering.
- Loopback interface that is used as the source for the sparse PIM rendezvous point (RP) router address.

In addition, you must also associate MVRFs with those interfaces over which they are going to forward multicast traffic.

BGP should be already configured and operational on all routers that are sending or receiving multicast traffic. In addition, BGP extended communities must be enabled (using the **neighbor send-community both** or **neighbor send-community extended** command) to support the use of MDTs in the network.

Configuring PIM on an Interface

To configure PIM on an interface, perform this task

	Command 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.

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

When configuring PIM on an interface, note the following information:

- You can use one of these interface types:
 - A physical interface on a provider edge (PE) router that is connected to the backbone.
 - A loopback interface that is used for BGP peering.
 - A loopback interface that is used as the source for the sparse PIM network rendezvous point (RP) address.
- These are the PIM modes:
 - dense-mode**—Enables dense mode of operation.
 - sparse-mode**—Enables sparse mode of operation.
 - sparse-dense-mode**—Enables sparse mode if the multicast group has an RP router defined, or enables dense mode if an RP router is not defined.
- Use **sparse-mode** for the physical interfaces of all PE routers that are connected to the backbone, and on all loopback interfaces that are used for BGP peering or as the source for RP addressing.

This example shows how to configure PIM sparse mode on a physical interface:

```
Router# configure terminal
interface gigabitethernet 10/1
Router(config-if)# ip pim sparse-mode
```

This example shows how to configure PIM sparse mode on a loopback interface:

```
Router# configure terminal
Router(config)# interface loopback 2
Router(config-if)# ip pim sparse-mode
```

Configuring an Interface for IPv4 VRF Forwarding

To configure an interface for IPv4 VRF forwarding, perform this task:

Command 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, page 26-22](#)
- [MVPN Configuration with Default and Data MDTs, page 26-24](#)

MVPN Configuration with Default MDTs Only

The following excerpt from a configuration file shows the lines that are related to the MVPN configuration for three MVRFs. (The required BGP configuration is not shown.)

```
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service compress-config
!
hostname MVPN Router
!
boot system flash slot0:
logging snmp-authfail
!
ip subnet-zero
!
!
no ip domain-lookup
ip host tftp 223.255.254.238
!
```

```
ip vrf mvpn-cus1
 rd 200:1
 route-target export 200:1
 route-target import 200:1
 mdt default 239.1.1.1
!
ip vrf mvpn-cus2
 rd 200:2
 route-target export 200:2
 route-target import 200:2
 mdt default 239.1.1.2
!
ip vrf mvpn-cus3
 rd 200:3
 route-target export 200:3
 route-target import 200:3
 mdt default 239.1.1.3
!
ip multicast-routing
ip multicast-routing vrf mvpn-cus1
ip multicast-routing vrf mvpn-cus2
ip multicast-routing vrf mvpn-cus3
ip multicast multipath
frame-relay switching
mpls label range 4112 262143
mpls label protocol ldp
mpls ldp logging neighbor-changes
mpls ldp explicit-null
mpls traffic-eng tunnels
tag-switching tdp discovery directed-hello accept from 1
tag-switching tdp router-id Loopback0 force
mls ip multicast replication-mode ingress
mls ip multicast flow-stat-timer 9
mls ip multicast bidir gm-scan-interval 10
mls flow ip destination
no mls flow ipv6
mls rate-limit unicast cef glean 10 10
mls qos
mls cef error action freeze

...

vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
vlan 2001-2101,3501-3700,4001,4051-4080,4093
!
!
!
interface Loopback0
 ip address 201.252.1.14 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback1
 ip address 209.255.255.14 255.255.255.255
!
interface Loopback10
 ip vrf forwarding mvpn-cus1
 ip address 210.101.255.14 255.255.255.255
!
interface Loopback11
 ip vrf forwarding mvpn-cus1
 ip address 210.111.255.14 255.255.255.255
 ip pim sparse-dense-mode
```

```

!
interface Loopback12
 ip vrf forwarding mvpn-cus1
 ip address 210.112.255.14 255.255.255.255

...

!
interface GigabitEthernet3/3
 mtu 9216
 ip vrf forwarding mvpn-cus3
 ip address 172.10.14.1 255.255.255.0
 ip pim sparse-dense-mode
!

...

!
interface GigabitEthernet3/19
 ip vrf forwarding mvpn-cus2
 ip address 192.16.4.1 255.255.255.0
 ip pim sparse-dense-mode
 ip igmp static-group 229.1.1.1
 ip igmp static-group 229.1.1.2
 ip igmp static-group 229.1.1.4
!
interface GigabitEthernet3/20
 ip vrf forwarding mvpn-cus1
 ip address 192.16.1.1 255.255.255.0
 ip pim sparse-dense-mode
!

...

```

MVPN Configuration with Default and Data MDTs

The following sample configuration includes three MVRFs that have been configured for both default and data MDTs. Only the configuration that is relevant to the MVPN configuration is shown.

```

...
!
ip vrf v1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
 mdt default 226.1.1.1
 mdt data 226.1.1.128 0.0.0.7 threshold 1
!
ip vrf v2
 rd 2:2
 route-target export 2:2
 route-target import 2:2
 mdt default 226.2.2.1
 mdt data 226.2.2.128 0.0.0.7
!
ip vrf v3
 rd 3:3
 route-target export 3:3
 route-target import 3:3
 mdt default 226.3.3.1
 mdt data 226.3.3.128 0.0.0.7
!
ip vrf v4

```



```
rd 155.255.255.1:4
route-target export 155.255.255.1:4
route-target import 155.255.255.1:4
mdt default 226.4.4.1
mdt data 226.4.4.128 0.0.0.7
!
ip multicast-routing
ip multicast-routing vrf v1
ip multicast-routing vrf v2
ip multicast-routing vrf v3
ip multicast-routing vrf v4
mpls label protocol ldp
mpls ldp logging neighbor-changes
tag-switching tdp router-id Loopback1
mls ip multicast replication-mode ingress
mls ip multicast bidir gm-scan-interval 10
no mls flow ip
no mls flow ipv6
mls cef error action freeze
!
!
!
!
!
...

vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
!
interface Loopback1
 ip address 155.255.255.1 255.255.255.255
 ip pim sparse-mode
!
interface Loopback4
 ip vrf forwarding v4
 ip address 155.255.4.4 255.255.255.255
 ip pim sparse-mode
!
interface Loopback11
 ip vrf forwarding v1
 ip address 155.255.255.11 255.255.255.255
 ip pim sparse-dense-mode
!
interface Loopback22
 ip vrf forwarding v2
 ip address 155.255.255.22 255.255.255.255
 ip pim sparse-mode
!
interface Loopback33
 ip vrf forwarding v3
 ip address 155.255.255.33 255.255.255.255
 ip pim sparse-mode
!
interface Loopback44
 no ip address
!
interface Loopback111
 ip vrf forwarding v1
 ip address 1.1.1.1 255.255.255.252
 ip pim sparse-dense-mode
 ip ospf network point-to-point
!
```

```

interface GigabitEthernet1/1
  description Gi1/1 - 155.50.1.155 255.255.255.0 - peer dut50 - mpls
  mtu 9216
  ip address 155.50.1.155 255.255.255.0
  ip pim sparse-mode
  tag-switching ip
!
interface GigabitEthernet1/2
  ip vrf forwarding v1
  ip address 155.1.2.254 255.255.255.0
  ip pim sparse-mode
!
interface GigabitEthernet1/3
  description Gi1/3 - 185.155.1.155/24 - vrf v1 stub peer 185.Gi1/3
  ip vrf forwarding v1
  ip address 185.155.1.155 255.255.255.0
  ip pim sparse-mode
!
...
!
interface GigabitEthernet1/48
  ip vrf forwarding v1
  ip address 157.155.1.155 255.255.255.0
  ip pim bsr-border
  ip pim sparse-dense-mode
!
interface GigabitEthernet6/1
  no ip address
  shutdown
!
interface GigabitEthernet6/2
  ip address 9.1.10.155 255.255.255.0
  media-type rj45
!
interface Vlan1
  no ip address
  shutdown
!
router ospf 11 vrf v1
  router-id 155.255.255.11
  log-adjacency-changes
  redistribute connected subnets tag 155
  redistribute bgp 1 subnets tag 155
  network 1.1.1.0 0.0.0.3 area 155
  network 155.255.255.11 0.0.0.0 area 155
  network 155.0.0.0 0.255.255.255 area 155
  network 157.155.1.0 0.0.0.255 area 0
!
router ospf 22 vrf v2
  router-id 155.255.255.22
  log-adjacency-changes
  network 155.255.255.22 0.0.0.0 area 155
  network 155.0.0.0 0.255.255.255 area 155
  network 157.155.1.0 0.0.0.255 area 0
!
router ospf 33 vrf v3
  router-id 155.255.255.33
  log-adjacency-changes
  network 155.255.255.33 0.0.0.0 area 155
!
router ospf 1
  log-adjacency-changes

```

```

network 155.50.1.0 0.0.0.255 area 0
network 155.255.255.1 0.0.0.0 area 155
!
router bgp 1
  bgp router-id 155.255.255.1
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  neighbor 175.255.255.1 remote-as 1
  neighbor 175.255.255.1 update-source Loopback1
  neighbor 185.255.255.1 remote-as 1
  neighbor 185.255.255.1 update-source Loopback1
  !
  address-family vpnv4
    neighbor 175.255.255.1 activate
    neighbor 175.255.255.1 send-community extended
    neighbor 185.255.255.1 activate
    neighbor 185.255.255.1 send-community extended
  exit-address-family
  !
  address-family ipv4 vrf v4
    no auto-summary
    no synchronization
  exit-address-family
  !
  address-family ipv4 vrf v3
    redistribute ospf 33
    no auto-summary
    no synchronization
  exit-address-family
  !
  address-family ipv4 vrf v2
    redistribute ospf 22
    no auto-summary
    no synchronization
  exit-address-family
  !
  address-family ipv4 vrf v1
    redistribute ospf 11
    no auto-summary
    no synchronization
  exit-address-family
  !
ip classless
ip route 9.255.254.1 255.255.255.255 9.1.10.254
no ip http server
ip pim bidir-enable
ip pim rp-address 50.255.2.2 MCAST.MVPN.MDT.v2 override bidir
ip pim rp-address 50.255.3.3 MCAST.MVPN.MDT.v3 override bidir
ip pim rp-address 50.255.1.1 MCAST.MVPN.MDT.v1 override bidir
ip pim vrf v1 spt-threshold infinity
ip pim vrf v1 send-rp-announce Loopback11 scope 16 group-list MCAST.GROUP.BIDIR bidir
ip pim vrf v1 send-rp-discovery Loopback11 scope 16
ip pim vrf v1 bsr-candidate Loopback11 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
...
```



Configuring IP Unicast Layer 3 Switching

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



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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 Layer 3 Switching Works](#), page 27-1
- [Default Hardware Layer 3 Switching Configuration](#), page 27-4
- [Configuration Guidelines and Restrictions](#), page 27-4
- [Configuring Hardware Layer 3 Switching](#), page 27-4
- [Displaying Hardware Layer 3 Switching Statistics](#), page 27-5



Note

- Supervisor Engine 720 and Supervisor Engine 2 support IPX with fast switching 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 29, “Configuring IPv4 Multicast Layer 3 Switching.”](#)

Understanding How Layer 3 Switching Works

These sections describe Layer 3 switching:

- [Understanding Hardware Layer 3 Switching](#), page 27-2
- [Understanding Layer 3-Switched Packet Rewrite](#), page 27-2

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 PFC 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 Catalyst 6500 series switch.

Hardware Layer 3 switching generates flow statistics for Layer 3-switched traffic. Hardware Layer 3 flow statistics can be used for NetFlow Data Export (NDE). (See [Chapter 50, “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 Catalyst 6500 series switch 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 switch recognizes that the packet was sent to the Layer 2 (MAC) address of the MSFC.

To perform Layer 3 switching, the switch rewrites the Layer 2 frame header, changing the Layer 2 destination address to the Layer 2 address of Destination B and the Layer 2 source address to the Layer 2 address of the MSFC. The Layer 3 addresses remain the same.

In IP unicast and IP multicast traffic, the switch decrements the Layer 3 TTL value by 1 and recomputes the Layer 3 packet checksum. The switch recomputes the Layer 2 frame checksum and forwards (or, for multicast packets, replicates as necessary) the rewritten packet to Destination B's subnet.

A received IP unicast packet is formatted (conceptually) as follows:

Layer 2 Frame Header		Layer 3 IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
MSFC MAC	Source A MAC	Destination B IP	Source A IP	n	calculation1		

After the switch rewrites an IP unicast packet, it is formatted (conceptually) as follows:

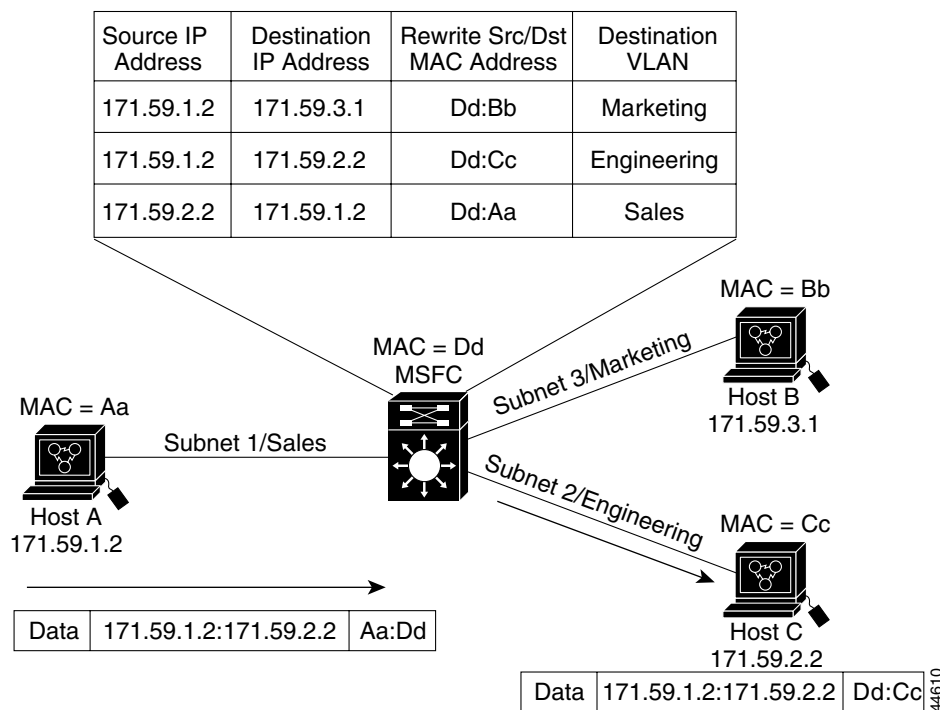
Layer 2 Frame Header		Layer 3 IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
Destination B MAC	MSFC MAC	Destination B IP	Source A IP	n-1	calculation2		

Hardware Layer 3 Switching Examples

Figure 27-1 on page 27-3 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 27-1 Hardware Layer 3 Switching Example Topology



Default Hardware Layer 3 Switching Configuration

Table 27-1 shows the default hardware Layer 3 switching configuration.

Table 27-1 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 23, “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.



Configuring IPv6 Multicast PFC3 and DFC3 Layer 3 Switching

With Release 12.2(18)SXE and later releases, the PFC3 and DFC3 provide hardware support for IPv6 multicast traffic. Use these publications to configure IPv6 multicast on Catalyst 6500 series switches:

- The *Cisco IOS IPv6 Configuration Library*, “Implementing IPv6 Multicast”:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/sa_mcast.htm
- The *Cisco IOS IPv6 Command Reference*:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_r/index.htm

These sections provide additional information about IPv6 multicast support on Catalyst 6500 series switches:

- [Features that Support IPv6 Multicast, page 28-1](#)
- [IPv6 Multicast Guidelines and Restrictions, page 28-2](#)
- [New or Changed IPv6 Multicast Commands, page 28-3](#)
- [Configuring IPv6 Multicast Layer 3 Switching, page 28-3](#)
- [Using show Commands to Verify IPv6 Multicast Layer 3 Switching, page 28-3](#)

Features that Support IPv6 Multicast

These features support IPv6 multicast:

- RPR and RPR+ redundancy mode—See [Chapter 9, “Configuring RPR and RPR+ Supervisor Engine Redundancy.”](#)
- Multicast Listener Discovery version 2 (MLDv2) snooping—See [Chapter 30, “Configuring MLDv2 Snooping for IPv6 Multicast Traffic.”](#)



Note MLDv1 snooping is not supported.

- IPv6 Multicast rate limiters—See [Chapter 37, “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 Catalyst 6500 series switches:

- With Release 12.2(18)SXE and later releases, the PFC3 and DFC3 provide hardware support for the following:
 - Completely switched IPv6 multicast flows
 - IPv6 PIM-Sparse Mode (PIM-SM) (S,G) forwarding
 - Multicast RPF check for IPv6 PIM-SM (S,G) traffic using the NetFlow table
 - Rate limiting of IPv6 PIM-SM (S,G) traffic that fails the multicast RPF check
 - Static IPv6 multicast routes
 - SSM Mapping for IPv6 (PIM-SSM)
 - IPv6 multicast forwarding information base (MFIB) using the NetFlow table
 - IPv6 distributed MFIB (dMFIB) using the NetFlow table
 - Link-local and link-global IPv6 multicast scopes
 - Egress multicast replication with the **ipv6 mfib hardware-switching** command
 - Ingress interface statistics for multicast routes (egress interface statistics not available)
 - RPR and RPR+ redundancy mode (see [Chapter 9, “Configuring RPR and RPR+ Supervisor Engine Redundancy”](#))
 - Ingress and egress PFC QoS (see [Chapter 42, “Configuring PFC QoS”](#))
 - Input and output Cisco access-control lists (ACLs)
- The PFC3 and DFC3 do not provide hardware support for the following:
 - Partially switched IPv6 multicast flows
 - PIM-SM (*,G) forwarding
 - Multicast RPF check for PIM-SM (*,G) traffic
 - Multicast helper maps
 - Site-local multicast scopes
 - Manually configured IPv6 over IPv4 tunnels
 - IPv6 multicast 6to4 tunnels
 - IPv6 multicast automatic tunnels
 - IPv6 over GRE tunnels
 - IPv6-in-IPv6 PIM register tunnels
 - IPv6 multicast basic ISATAP tunnels
 - ISATAP tunnels with embedded 6to4 tunnels

New or Changed IPv6 Multicast Commands

Refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX for information about these IPv6 multicast commands, which are new or changed in Release 12.2(18)SXE:

- **ipv6 mfib hardware-switching**
- **mls rate-limit multicast ipv6** (see [Chapter 37, “Configuring Denial of Service Protection”](#))
- **show ipv6 mfib**
- **show mls rate-limit** (see [Chapter 37, “Configuring Denial of Service Protection”](#))
- **show platform software ipv6-multicast**
- **show tcam interface**

Configuring IPv6 Multicast Layer 3 Switching

To configure IPv6 multicast Layer 3 switching, perform this task:

	Command	Purpose
Step 1	Router(config)# ipv6 unicast-routing	Enables unicast routing on all Layer 3 interfaces.
Step 2	Router(config)# ipv6 multicast-routing	Enables PIM-SM on all Layer 3 interfaces.
Step 3	Router(config)# ipv6 mfib hardware-switching	Enables MFIB hardware switching globally.

Using show Commands to Verify IPv6 Multicast Layer 3 Switching

These sections describe how to use **show** commands to verify IPv6 multicast Layer 3 switching:

- [Verifying MFIB Clients, page 28-4](#)
- [Displaying the Switching Capability, page 28-4](#)
- [Verifying the \(S,G\) Forwarding Capability, page 28-4](#)
- [Verifying the \(*,G\) Forwarding Capability, page 28-5](#)
- [Verifying the Subnet Entry Support Status, page 28-5](#)
- [Verifying the Current Replication Mode, page 28-5](#)
- [Displaying the Replication Mode Auto Detection Status, page 28-5](#)
- [Displaying the Replication Mode Capabilities, page 28-5](#)
- [Displaying Subnet Entries, page 28-6](#)
- [Displaying the IPv6 Multicast Summary, page 28-6](#)
- [Displaying the NetFlow Hardware Forwarding Count, page 28-6](#)
- [Displaying the FIB Hardware Bridging and Drop Counts, page 28-7](#)
- [Displaying the Shared and Well-Known Hardware Adjacency Counters, page 28-7](#)

**Note**

The show commands in the following sections are for a switch with a DFC3-equipped switching module in slot 1 and a Supervisor Engine 720 with a PFC3 in slot 6.

Verifying MFIB Clients

This example shows the complete output of the **show ipv6 mrib client** command:

```
Router# show ipv6 mrib client
IP MRIB client-connections
mfib ipv6:81      (connection id 0)
igmp:124          (connection id 1)
pim:281 (connection id 2)
slot 1  mfib ipv6 rp agent:15  (connection id 3)
slot 6  mfib ipv6 rp agent:15  (connection id 4)
```

This example shows how to display the MFIB client running on the MSFC:

```
Router# show ipv6 mrib client | include ^mfib ipv6
mfib ipv6:81      (connection id 0)
```

This example shows how to display the MFIB clients running on the PFC3 and any DFC3s:

```
Router# show ipv6 mrib client | include slot
slot 1  mfib ipv6 rp agent:15  (connection id 3)
slot 6  mfib ipv6 rp agent:15  (connection id 4)
```

Displaying the Switching Capability

This example displays the complete output of the **show platform software ipv6-multicast capability** command:

```
Router# show platform software ipv6-multicast capability

Hardware switching for IPv6 is enabled
(S,G) forwarding for IPv6 supported using Netflow
(*,G) bridging for IPv6 is supported using FIB
Directly-connected entries for IPv6 is supported using ACL-TCAM.

Current System HW Replication Mode : Ingress
Auto-detection of Replication Mode : ON

Slot Replication-Capability Replication-Mode
  1 Ingress                  Ingress
  2 Egress                   Ingress
  6 Egress                   Ingress
  8 Ingress                  Ingress
```

Verifying the (S,G) Forwarding Capability

This example shows how to verify the (S,G) forwarding:

```
Router# show platform software ipv6-multicast capability | include (S,G)
(S,G) forwarding for IPv6 supported using Netflow
```

Verifying the (*,G) Forwarding Capability

This example shows how to verify the (*,G) forwarding:

```
Router# show platform software ipv6-multicast capability | include (*,G)
(*,G) bridging for IPv6 is supported using FIB
```

Verifying the Subnet Entry Support Status

This example shows how to verify the subnet entry support status:

```
Router# show platform software ipv6-multicast capability | include entries
Directly-connected entries for IPv6 is supported using ACL-TCAM.
```

Verifying the Current Replication Mode

This example shows how to verify the current replication mode:

```
Router# show platform software ipv6-multicast capability | include Current
Current System HW Replication Mode : Ingress
```



Note

Enter the **no ipv6 mfib hardware-switching replication-mode ingress** command to enable replication mode auto detection.

Displaying the Replication Mode Auto Detection Status

This example shows how to display the replication mode auto detection status:

```
Router# show platform software ipv6-multicast capability | include detection
Auto-detection of Replication Mode : ON
```

Displaying the Replication Mode Capabilities

This example shows how to display the replication mode capabilities of the installed modules:

```
Router# show platform software ipv6-multicast capability | begin ^Slot
Slot Replication-Capability Replication-Mode
  1 Ingress                  Ingress
  2 Egress                   Ingress
  6 Egress                   Ingress
  8 Ingress                  Ingress
```

Displaying Subnet Entries

This example shows how to display subnet entries:

```
Router# show platform software ipv6-multicast connected
IPv6 Multicast Subnet entries
Flags : H - Installed in ACL-TCAM
        X - Not installed in ACL-TCAM due to
            label-full exception
Interface: Vlan20 [ H ]
           S:20::1 G:FF00::
Interface: Vlan10 [ H ]
           S:10::1 G:FF00::
```



Note In this example, there are subnet entries for VLAN 10 and VLAN 20.

Displaying the IPv6 Multicast Summary

This example shows how to display the IPv6 multicast summary:

```
Router# show platform software ipv6-multicast summary
IPv6 Multicast Netflow SC summary on Slot[1]:
Shortcut Type          Shortcut count
-----+-----
(S, G)                 100
(*, G)                 0
IPv6 Multicast FIB SC summary on Slot[1]:
Shortcut Type          Shortcut count
-----+-----
(*, G/128)             10
(*, G/m)               47

IPv6 Multicast Netflow SC summary on Slot[6]:
Shortcut Type          Shortcut count
-----+-----
(S, G)                 100
(*, G)                 0
IPv6 Multicast FIB SC summary on Slot[6]:
Shortcut Type          Shortcut count
-----+-----
(*, G/128)             10
(*, G/m)               47
```

Displaying the NetFlow Hardware Forwarding Count

This example shows how to display the NetFlow hardware forwarding count:

```
Router# show platform software ipv6-multicast summary
IPv6 Multicast Netflow SC summary on Slot[1]:
Shortcut Type          Shortcut count
-----+-----
(S, G)                 100
(*, G)                 0

<...Output deleted...>
```



```
IPv6 Multicast Netflow SC summary on Slot[6]:
Shortcut Type                Shortcut count
-----+-----
(S, G)                      100
(*, G)                      0
<...Output truncated...>
```

**Note**

The Netflow (*, G) count is always zero because PIM-SM (*,G) forwarding is supported in software on the MSFC3.

Displaying the FIB Hardware Bridging and Drop Counts

This example shows how to display the FIB hardware bridging and drop hardware counts:

```
Router# show platform software ipv6-multicast summary | begin FIB
IPv6 Multicast FIB SC summary on Slot[1]:
Shortcut Type                Shortcut count
-----+-----
(*, G/128)                  10
(*, G/m)                    47
<...Output deleted...>

IPv6 Multicast FIB SC summary on Slot[6]:
Shortcut Type                Shortcut count
-----+-----
(*, G/128)                  10
(*, G/m)                    47
```

**Note**

- The (*, G/128) value is a hardware bridge entry count.
- The (*, G/m) value is a hardware bridge/drop entry count.

Displaying the Shared and Well-Known Hardware Adjacency Counters

The `show platform software ipv6-multicast shared-adjacencies` command displays the shared and well-known hardware adjacency counters used for IPv6 multicast by entries in FIB and ACL-TCAM.

```
Router# show platform software ipv6-multicast shared-adjacencies
```

```
---- SLOT [1] ----
```

Shared IPv6 Mcast Adjacencies	Index	Packets	Bytes
Subnet bridge adjacency	0x7F802	0	0
Control bridge adjacency	0x7	0	0
StarG_M bridge adjacency	0x8	0	0
S_G bridge adjacency	0x9	0	0
Default drop adjacency	0xA	0	0
StarG (spt == INF) adjacency	0xB	0	0
StarG (spt != INF) adjacency	0xC	0	0

Using show Commands to Verify IPv6 Multicast Layer 3 Switching

```
---- SLOT [6] ----
```

Shared IPv6 Mcast Adjacencies	Index	Packets	Bytes
Subnet bridge adjacency	0x7F802	0	0
Control bridge adjacency	0x7	0	0
StarG_M bridge adjacency	0x8	0	0
S_G bridge adjacency	0x9	0	0
Default drop adjacency	0xA	28237	3146058
StarG (spt == INF) adjacency	0xB	0	0
StarG (spt != INF) adjacency	0xC	0	0



Configuring IPv4 Multicast Layer 3 Switching

This chapter describes how to configure IPv4 multicast Layer 3 switching on the Catalyst 6500 series switches.



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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](#), page 29-1
- [Understanding How IPv4 Bidirectional PIM Works](#), page 29-6
- [Default IPv4 Multicast Layer 3 Switching Configuration](#), page 29-7
- [IPv4 Multicast Layer 3 Switching Configuration Guidelines and Restrictions](#), page 29-7
- [Configuring IPv4 Multicast Layer 3 Switching](#), page 29-8
- [Configuring IPv4 Bidirectional PIM](#), page 29-22

Understanding How IPv4 Multicast Layer 3 Switching Works

These sections describe how IPv4 multicast Layer 3 switching works:

- [IPv4 Multicast Layer 3 Switching Overview](#), page 29-2
- [Multicast Layer 3 Switching Cache](#), page 29-2
- [Layer 3-Switched Multicast Packet Rewrite](#), page 29-3
- [Partially and Completely Switched Flows](#), page 29-3
- [Non-RPF Traffic Processing](#), page 29-5
- [Understanding How IPv4 Bidirectional PIM Works](#), page 29-6

IPv4 Multicast Layer 3 Switching Overview

The Policy Feature Card (PFC) provides Layer 3 switching for IP multicast flows using the hardware replication table and hardware Cisco Express Forwarding (CEF), which uses the forwarding information base (FIB) and the adjacency table on the PFC. In systems with Distributed Forwarding Cards (DFCs), IP multicast flows are Layer 3 switched locally using Multicast Distributed Hardware Switching (MDHS). MDHS uses local hardware CEF and replication tables on each DFC to perform Layer 3 switching and rate limiting of reverse path forwarding (RPF) failures locally on each DFC-equipped switching module.

The PFC and the DFCs support hardware switching of (*,G) state flows. The PFC and the DFCs support rate limiting of non-RPF traffic.

Multicast Layer 3 switching forwards IP multicast data packet flows between IP subnets using advanced application-specific integrated circuit (ASIC) switching hardware, which offloads processor-intensive multicast forwarding and replication from network routers.

Layer 3 flows that cannot be hardware switched are still forwarded in the software by routers. Protocol Independent Multicast (PIM) is used for route determination.

The PFC and the DFCs all use the Layer 2 multicast forwarding table to determine on which ports Layer 2 multicast traffic should be forwarded (if any). The multicast forwarding table entries are populated in conjunction with Internet Group Management Protocol (IGMP) snooping (see [Chapter 31, “Configuring IGMP Snooping for IPv4 Multicast Traffic”](#)).

Multicast Layer 3 Switching Cache

This section describes how the PFC and the DFCs maintain Layer 3 switching information in hardware tables.

The PFC and DFC populate the (S,G) or (*,G) flows in the hardware FIB table with the appropriate masks; for example, (S/32, G/32) and (*/0, G/32). The RPF interface and the adjacency pointer information is also stored in each entry. The adjacency table contains the rewrite and a pointer to the replication entries. If a flow matches a FIB entry, the RPF check compares the incoming interface/VLAN with the entry. A mismatch is an RPF failure, which can be rate limited if this feature is enabled.

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.
- When you disable multicast Layer 3 switching on an individual interface basis using the **no mls ipmulticast** command, flows that use this interface as the RPF interface are routed only by the MSFC in software.

Layer 3-Switched Multicast Packet Rewrite

When a multicast packet is Layer 3 switched from a multicast source to a destination multicast group, the PFC and the DFCs perform a packet rewrite that is based on information learned from the 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 switch also bridges the packet in the source VLAN).

When the PFC receives the multicast packet, it is (conceptually) formatted as follows:

Layer 2 Frame Header		Layer 3 IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
<i>Group G1 MAC¹</i>	<i>Source A MAC</i>	<i>Group G1 IP</i>	<i>Source A IP</i>	<i>n</i>	<i>calculation1</i>		

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. This MAC address can be displayed using the **show mls multicast statistics** command.)
- Decrements the IP header Time to Live (TTL) by one and recalculates the IP header checksum

The result is a rewritten IP multicast packet that appears to have been routed. The PFC replicates the rewritten packet onto the appropriate destination VLANs, where it is forwarded to members of IP multicast group G1.

After the PFC performs the packet rewrite, the packet is (conceptually) formatted as follows:

Frame Header		IP Header				Data	FCS
Destination	Source	Destination	Source	TTL	Checksum		
<i>Group G1 MAC</i>	<i>MSFC MAC</i>	<i>Group G1 IP</i>	<i>Source A IP</i>	<i>n-1</i>	<i>calculation2</i>		

Partially and Completely Switched Flows

When at least one outgoing Layer 3 interface for a given flow is multilayer switched and at least one outgoing interface is not multilayer switched, that flow is considered partially switched. When a partially switched flow is created, all multicast traffic belonging to that flow still reaches the MSFC and is forwarded by software on those outgoing interfaces that are not multilayer switched.

These sections describe partially and completely switched flow:

- [Partially Switched Flows, page 29-4](#)
- [Completely Switched Flows, page 29-4](#)

Partially Switched Flows

A flow might be partially switched instead of completely switched in these situations:

- If the switch is configured as a member of the IP multicast group on the RPF interface of the multicast source (using the **ip igmp join-group** command).
- During the registering state, if the switch is the first-hop router to the source in PIM sparse mode (in this case, the switch must send PIM-register messages to the rendezvous point [RP]).
- If the multicast TTL threshold is configured on an outgoing interface for the flow (using the **ip multicast ttl-threshold** command).
- If the multicast helper is configured on the RPF interface for the flow, and multicast to broadcast translation is required.
- With a PFC2, if the outgoing interface is a generic routing encapsulation (GRE) tunnel interface.
- With a PFC3 and releases earlier than Release 12.2(18)SXE, if the outgoing interface is a generic routing encapsulation (GRE) tunnel interface.
- If the outgoing interface is a Distance Vector Multicast Routing Protocol (DVMRP) tunnel interface.
- If Network Address Translation (NAT) is configured on an interface and source address translation is required for the outgoing interface.
- Flows are partially switched if any of the outgoing interfaces for a given flow are not Layer 3 switched.

(S,G) flows are partially switched instead of completely switched in these situations:

- (S,G) flows are partially switched if the (S,G) entry has the RPT-bit (R bit) set.
- (S,G) flows are partially switched if the (S,G) entry does not have the SPT bit (T flag) set and the Prune bit (P flag) set.

(* ,G) flows are partially switched instead of completely switched in these situations:

- (* ,G) flows are partially switched on the last-hop leaf router if the shared-tree to shortest-path-tree (SPT) threshold is not equal to infinity. This allows the flow to transition from the SPT.
- (* ,G) flows are partially switched if at least one (S,G) entry has the same RPF as a (* ,g) entry but any of these is true:
 - The RPT flag (R bit) is not set.
 - The SPT flag(T bit) is not set.
 - The Prune-flag (P bit) is not set.
- (* ,G) flows are partially switched if a DVMRP neighbor is detected on the input interface of a (* ,G) entry.
- (* ,G) flows are partially switched if the interface and mask entry is not installed for the RPF-interface of a (* ,G) entry and the RPF interface is not a point-to-point interface.

Completely Switched Flows

When all the outgoing interfaces for a given flow are Layer 3 switched, and none of the above situations apply to the flow, that flow is considered completely switched. When a completely switched flow is created, the 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:

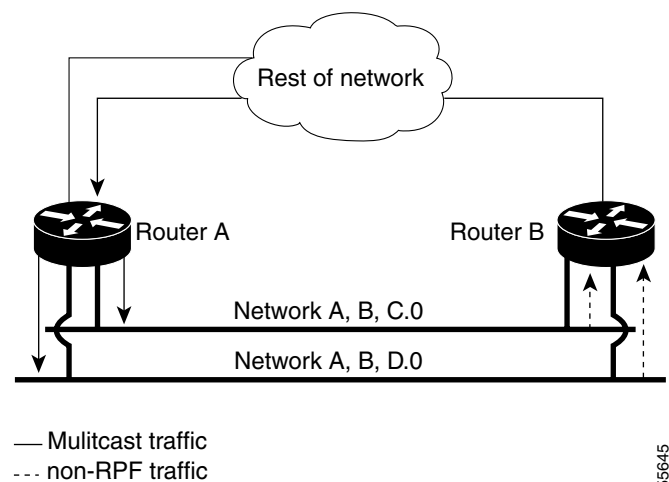
- [Non-RPF Traffic Overview, page 29-5](#)
- [Filtering of RPF Failures for Stub Networks, page 29-6](#)
- [Rate Limiting of RPF Failure Traffic, page 29-6](#)

Non-RPF Traffic Overview

In a redundant configuration where multiple routers connect to the same LAN segment, only one router forwards the multicast traffic from the source to the receivers on the outgoing interfaces (see [Figure 29-1](#)). In this kind of topology, only the PIM designated router (PIM DR) forwards the data in the common VLAN, but the non-PIM DR receives the forwarded multicast traffic. The redundant router (non-PIM DR) must drop this traffic because it has arrived on the wrong interface and fails the RPF check. Traffic that fails the RPF check is called non-RPF traffic.

The Catalyst 6500 series switch processes non-RPF traffic in hardware on the PFC by filtering (dropping) or rate limiting the non-RPF traffic.

Figure 29-1 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 29-16](#).

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

Supervisor Engine 720 supports hardware forwarding of IPv4 bidirectional PIM groups. To support IPv4 bidirectional PIM groups, the Supervisor Engine 720 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 29-22](#).

Default IPv4 Multicast Layer 3 Switching Configuration

[Table 29-1](#) shows the default IP multicast Layer 3 switching configuration.

Table 29-1 Default IP Multicast Layer 3 Switching Configuration

Feature	Default Value
ACL for stub networks	Disabled on all interfaces
Installing of directly connected subnet entries	Enabled globally
Multicast routing	Disabled globally
PIM routing	Disabled on all interfaces
IP multicast Layer 3 switching	Enabled when multicast routing is enabled and PIM is enabled on the interface
Shortcut consistency checking	Enabled

Internet Group Management Protocol (IGMP) snooping is enabled by default on all VLAN interfaces. If you disable IGMP snooping on an interface, multicast Layer 3 flows are still switched by the hardware. Bridging of the flow on an interface with IGMP snooping disabled causes flooding to all forwarding interfaces of the VLAN. For details on configuring IGMP snooping, see [Chapter 31, “Configuring IGMP Snooping for IPv4 Multicast Traffic.”](#)

IPv4 Multicast Layer 3 Switching Configuration Guidelines and Restrictions

These sections describe IP Multicast Layer 3 switching configuration restrictions:

- [Restrictions, page 29-7](#)
- [Unsupported Features, page 29-8](#)

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 switch is the RP for the group.
- For IPv4 bidirectional PIM entries when a DF interface or RPF interface is a tunnel.
- With a PFC2, NAT translations for multicast packets is handled in software.
- With a PFC2, GRE tunnel encapsulation and de-encapsulation for multicast packets is handled in software.
- With a PFC3 and releases earlier than Release 12.2(18)SXE, GRE tunnel encapsulation and de-encapsulation for multicast packets is handled in software.
- Supervisor Engine 32 does not support egress multicast replication and cannot detect the multicast replication mode.

Unsupported Features

If you enable IP multicast Layer 3 switching, IP accounting for Layer 3 interfaces does not report accurate values. The **show ip accounting** command is not supported.

Configuring IPv4 Multicast Layer 3 Switching

These sections describe how to configure IP multicast Layer 3 switching:

- [Source-Specific Multicast with IGMPv3, IGMP v3lite, and URD, page 29-9](#)
- [Enabling IPv4 Multicast Routing Globally, page 29-9](#)
- [Enabling IPv4 PIM on Layer 3 Interfaces, page 29-10](#)
- [Enabling IP Multicast Layer 3 Switching on Layer 3 Interfaces, page 29-10](#)
- [Configuring the Replication Mode, page 29-11](#)
- [Enabling Local Egress Replication, page 29-12](#)
- [Specifying the Maximum Number of Multicast Routes, page 29-14](#)
- [Configuring the Layer 3 Switching Global Threshold, page 29-14](#)
- [Enabling Installation of Directly Connected Subnets, page 29-15](#)
- [Specifying the Flow Statistics Message Interval, page 29-15](#)
- [Configuring IPv4 Bidirectional PIM, page 29-22](#)
- [Setting the IPv4 Bidirectional PIM Scan Interval, page 29-24](#)
- [Enabling Shortcut-Consistency Checking, page 29-15](#)
- [Configuring ACL-Based Filtering of RPF Failures, page 29-16](#)

- [Displaying RPF Failure Rate-Limiting Information](#), page 29-16
- [Displaying IPv4 Multicast Layer 3 Hardware Switching Summary](#), page 29-17
- [Displaying the IPv4 Multicast Routing Table](#), page 29-20
- [Displaying IPv4 Multicast Layer 3 Switching Statistics](#), page 29-21
- [Displaying IPv4 Bidirectional PIM Information](#), page 29-24
- [Using IPv4 Debug Commands](#), page 29-26
- [Clearing IPv4 Multicast Layer 3 Switching Statistics](#), page 29-27

**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/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcpt2/index.htm
- *Cisco IOS IP and IP Routing Command Reference*, Release 12.1, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipras_r/index.htm

To enable IP multicast routing globally, perform this task:

Command	Purpose
Router(config)# ip multicast-routing	Enables IP multicast routing globally.
Router(config)# no ip multicast-routing	Disables IP multicast routing globally.

This example shows how to enable multicast routing globally:

```
Router(config)# ip multicast-routing
Router(config)#
```

Enabling IPv4 PIM on Layer 3 Interfaces

You must enable PIM on the Layer 3 interfaces before IP multicast Layer 3 switching functions on those interfaces.

To enable IP PIM on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port}}	Selects an interface to configure.
Step 2	Router(config-if)# ip pim {dense-mode sparse-mode sparse-dense-mode}	Enables IP PIM on a Layer 3 interface.
	Router(config-if)# no ip pim [dense-mode sparse-mode sparse-dense-mode]	Disables IP PIM on a Layer 3 interface.

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

This example shows how to enable PIM on an interface using the default mode (**sparse-dense-mode**):

```
Router(config-if)# ip pim
Router(config-if)#
```

This example shows how to enable PIM sparse mode on an interface:

```
Router(config-if)# ip pim sparse-mode
Router(config-if)#
```

Enabling IP Multicast Layer 3 Switching Globally

To enable hardware switching of multicast routes globally on your system, perform this task:

	Command	Purpose
Step 1	Router(config)# mls ip multicast	Globally enables hardware switching of multicast routes.
Step 2	Router# show mls ip multicast	Displays MLS IP multicast configuration.

This example shows how to globally enable hardware switching of multicast routes:

```
Router(config)# mls ip multicast
Router(config)#
```

Enabling IP Multicast Layer 3 Switching on Layer 3 Interfaces

IP multicast Layer 3 switching is enabled by default on the Layer 3 interface when you enable PIM on the interface. Perform this task only if you disabled IP multicast Layer 3 switching on the interface and you want to reenabling 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 29-10.

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.
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

Release 12.2(18)SXF and later releases support the **egress** keyword. Support for the **egress** keyword is called “Multicast Enhancement - Replication Mode Detection” in the release notes and Feature Navigator.

By default, 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 **mls ip multicast 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 **mls ip multicast replication-mode ingress** command in global configuration mode. This command forces the system to operate in ingress-replication mode.

The **no** form of the **mls ip multicast replication-mode ingress** command restores the system to automatic detection mode.

To enable IP multicast Layer 3 switching, perform this task:

	Command	Purpose
Step 1	Router(config)# mls ip multicast replication-mode [egress ingress]	Specifies the replication mode. Note Release 12.2(18)SXF and later releases support the egress keyword.
Step 2	Router# show mls ip multicast capability	Displays the configured replication mode.
Step 3	Router# show mls ip multicast summary	Displays the replication mode and if automatic detection is enabled or disabled.

This example shows how to enable the replication mode:

```
Router (config)# mls ip multicast replication-mode egress
Router# show mls ip multicast capability
Current mode of replication is Ingress
Configured replication mode is Egress
```

```
Slot          Multicast replication capability
 2              Egress
 3              Egress
 4              Ingress
 5              Egress
 6              Egress
```

```
Router# show mls ip multicast summary
4 MMLS entries using 656 bytes of memory
Number of partial hardware-switched flows:2
Number of complete hardware-switched flows:2
```

```
Directly connected subnet entry install is enabled
Current mode of replication is Ingress
Auto-detection of replication mode is enabled
Consistency checker is enabled
Router (config)#
```

Enabling Local Egress Replication

Release 12.2(18)SXF and later releases allow you to 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.

With Release 12.2(18)SXF and later releases, 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 mls ip multicast capability Router# show mls cef ip multicast detail	Displays the configured replication mode.

This example shows how to enable local egress replication:

```
Router (config)# mls ip multicast egress local
Router (config)# exit
Router # reload
Router # show mls ip multicast capability
Current mode of replication is Ingress
Configured replication mode is Egress
Egress Local is Enabled

Slot Multicast replication capability Egress Local
2 Egress No
3 Egress Yes
4 Ingress No
5 Egress No
6 Egress No
```

Specifying the Maximum Number of Multicast Routes

By default, a Supervisor Engine 720 supports 32,000 multicast routes in sparse mode. The following default settings apply for maximum number of multicast routes:

- 32,000 for PIM-SM/DM/SSM for ingress- or egress-replication mode
- 32,000 for IPv4 bidirectional PIM ingress-replication mode
- 10,700 for IPv4 bidirectional PIM egress-replication mode

By entering the **mls ip multicast max-routes** command, you can increase the maximum number of multicast routes to 64,000 for PIM-SM/DM/SSM with either ingress- and egress-replication mode.

**Note**

Rate limiting of directly connected sources is not available if you increase the maximum number of multicast routes above the default values.

To change the maximum number of multicast routes supported for PIM-SM/DM/SSM, perform this task:

	Command	Purpose
Step 1	Router(config)# mls ip multicast max-routes	Specifies the maximum number of multicast routes.
Step 1	Router# show mls ip multicast	Displays the multicast route configuration.

Configuring the Layer 3 Switching Global Threshold

You can configure a global multicast rate threshold (specified in packets per second) below which all multicast traffic is routed by the 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 <i>ppsec</i>	Configures the IP MMLS threshold.
Router(config)# no mls ip multicast threshold	Reverts to the default IP MMLS threshold.

This example shows how to configure the Layer 3 switching threshold to 10 packets per second:

```
Router(config)# mls ip multicast threshold 10
Router(config)#
```


Enabling Installation of Directly Connected Subnets

In PIM sparse mode, a first-hop router that is the designated router for the interface may need to encapsulate the source traffic in a PIM register message and unicast it to the rendezvous point. To prevent new sources for the group from being learned in the routing table, the (*,G) flows should remain as completely hardware-switched flows. When (subnet/mask, 224/4) entries are installed in the hardware, the FIB allows both (*,G) flows to remain completely hardware-switched flows, and new, directly connected sources to be learned correctly. The installation of directly connected subnets is enabled globally by default. One (subnet/mask, 224/4) is installed per PIM-enabled interface.

To view FIB entries, enter the **show mls ip multicast connected** command.

To enable installation of directly connected subnets, perform this task:

Command	Purpose
Router(config)# mls ip multicast connected	Enables installation of directly connected subnets.
Router(config)# no mls ip multicast connected	Disables installation of directly connected subnets.

This example shows how to enable installation of directly connected subnets:

```
Router(config)# mls ip multicast connected
Router(config)#
```

Specifying the Flow Statistics Message Interval

By default, the supervisor engine forwards flow statistics messages to the 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)#
```

Enabling Shortcut-Consistency Checking

When you enable the shortcut-consistency checking feature, the multicast route table and the multicast-hardware entries are checked for consistency, and any inconsistencies are corrected. You can view inconsistencies by entering the **show mls ip multicast consistency-check** command.

If consistency checking is enabled, the multicast route table will be scanned every two seconds and a full scan is completed within 4 minutes.

To enable shortcut-consistency checking, perform this task:

Command	Purpose
Router(config)# mls ip multicast consistency-check	Enables shortcut-consistency checking.
Router(config)# no mls ip multicast consistency-check <i>num</i>	Restores the default.

This example shows how to enable the hardware shortcut-consistency checker:

```
Router (config)# mls ip multicast consistency-check
Router (config)#
```

Configuring ACL-Based Filtering of RPF Failures

When you configure ACL-based filtering of RPF failures, ACLs that filter RPF failures in hardware are downloaded to the hardware-based ACL engine and applied on the interface you specify.

To enable ACL-based filtering of RPF failures on an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }}	Selects an interface to configure.
Step 2	Router(config-if)# mls ip multicast stub	Enables ACL-based filtering of RPF failures on an interface.
	Router(config-if)# no mls ip multicast stub	Disables ACL-based filtering of RPF failures on an interface.

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

Displaying RPF Failure Rate-Limiting Information

To display RPF failure rate-limiting information, perform this task:

Command	Purpose
Router# show mls ip multicast summary	Displays RPF failure rate-limiting information.

This example shows how to display RPF failure rate-limiting information:

```
Router# show mls ip multicast summary
10004 MMLS entries using 1280464 bytes of memory
Number of partial hardware-switched flows:4
Number of complete hardware-switched flows:10000
Router#
```

Displaying IPv4 Multicast Layer 3 Hardware Switching Summary



Note

The **show interface statistics** command does not display hardware-switched packets, only packets switched by software.

The **show ip pim interface count** command displays the IP multicast Layer 3 switching enable state on IP PIM interfaces and the number of packets received and sent on the interface.

To display IP multicast Layer 3 switching information for an IP PIM Layer 3 interface, perform one of these tasks:

Command	Purpose
Router# show ip pim interface [{vlan vlan_ID} {type ¹ slot/port} {port-channel number}] count	Displays IP multicast Layer 3 switching enable state information for all 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:

This example shows how to display the IP multicast Layer 3 switching configuration of Gigabit Ethernet interface 1/2:

Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide, Release 12.2SX

```
0 output buffer failures, 0 output buffers swapped out
Cat6k-B#
```

Displaying the IPv4 Multicast Routing Table

The **show ip mroute** command displays the IP multicast routing table.

To display the IP multicast routing table, perform this task:

Command	Purpose
Router# show ip mroute partical-sc [hostname group_number]	Displays the IP multicast routing table and the hardware-switched interfaces.

This example shows how to display the IP multicast routing table:

```
Router# show ip mroute 230.13.13.1
IP Multicast Routing Table
Flags:D - Dense, S - Sparse, s - SSM Group, C - Connected, L - Local,
      P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
      J - Join SPT, M - MSDP created entry, X - Proxy Join Timer Running
      A - Advertised via MSDP, U - URD, I - Received Source Specific Host
      Report
Outgoing interface flags:H - Hardware switched
Timers:Uptime/Expires
Interface state:Interface, Next-Hop or VCD, State/Mode

(*, 230.13.13.1), 00:16:41/00:00:00, RP 10.15.1.20, flags:SJC
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20
  Outgoing interface list:
    GigabitEthernet4/9, Forward/Sparse-Dense, 00:16:41/00:00:00, H

(*, 230.13.13.2), 00:16:41/00:00:00, RP 10.15.1.20, flags:SJC
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/9, Forward/Sparse-Dense, 00:16:41/00:00:00, H

(10.20.1.15, 230.13.13.1), 00:14:31/00:01:40, flags:CJT
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/9, Forward/Sparse-Dense, 00:14:31/00:00:00, H
(132.206.72.28, 224.2.136.89), 00:14:31/00:01:40, flags:CJT
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
  Outgoing interface list:Null
Router#
```



Note

The RPF-MFD flag indicates that the flow is completely switched by the hardware. The H flag indicates the flow is switched by the hardware on the outgoing interface.

Displaying IPv4 Multicast Layer 3 Switching Statistics

The **show mls ip multicast** command displays detailed information about IP multicast Layer 3 switching.

To display detailed IP multicast Layer 3 switching information, perform one of these tasks:

Command	Purpose
Router# show mls ip multicast group <i>ip_address</i> [interface <i>type slot/port</i> statistics]	Displays IP multicast Layer 3 switching group information.
Router# show mls ip multicast interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }} [statistics summary]	Displays IP multicast Layer 3 switching details for all interfaces.
Router# show mls ip multicast source <i>ip_address</i> [interface {{ vlan <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }} statistics]	Displays IP multicast Layer 3 switching source information.
Router# show mls ip multicast summary	Displays a summary of IP multicast Layer 3 switching information.
Router# show mls ip multicast statistics	Displays IP multicast Layer 3 switching statistics.

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

This example shows how to display information on a specific IP multicast Layer 3 switching entry:

```
Router# show mls ip multicast group 10.1.0.11
Multicast hardware switched flows:
Total shortcut installed: 0
```

This example shows how to display IP multicast group information:

```
Router# show mls ip multicast group 230.13.13.1 source 10.20.1.15
Multicast hardware switched flows:
(10.20.1.15, 230.13.13.1) Incoming interface:Gi4/8, Packets switched:0
Hardware switched outgoing interfaces:Gi4/9
RPF-MFD installed
```

```
Total hardware switched flows :1
Router#
```

This example shows how to display IP multicast Layer 3 switching information for VLAN 10:

```
Router# show mls ip multicast interface vlan 10
Multicast hardware switched flows:
(10.1.0.15, 224.2.2.15) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
MFD installed: Vlan10

(10.1.0.19, 224.2.2.19) Incoming interface: Vlan10, Packets switched: 1970
Hardware switched outgoing interfaces:
MFD installed: Vlan10

(10.1.0.11, 224.2.2.11) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
MFD installed: Vlan10

(10.1.0.10, 224.2.2.10) Incoming interface: Vlan10, Packets switched: 2744
Hardware switched outgoing interfaces:
MFD installed: Vlan10
```

```
(10.1.0.17, 224.2.2.17) Incoming interface: Vlan10, Packets switched: 3340
Hardware switched outgoing interfaces:
MFD installed: Vlan10
```

```
(10.1.0.13, 224.2.2.13) Incoming interface: Vlan10, Packets switched: 0
Hardware switched outgoing interfaces:
```

This example shows how to display the IP multicast Layer 3 switching statistics:

```
Router# show mls ip multicast statistics

MLS Multicast Operation Status:
MLS Multicast configuration and state:
  Router Mac: 00e0.b0ff.7b00, Router IP: 33.0.33.24
  MLS multicast operating state: ACTIVE
  Shortcut Request Queue size 4
  Maximum number of allowed outstanding messages: 1
  Maximum size reached from feQ: 3096
  Feature Notification sent: 1
  Feature Notification Ack received: 1
  Unsolicited Feature Notification received: 0
  MSM sent: 205170
  MSM ACK received: 205170
  Delete notifications received: 0
  Flow Statistics messages received: 35211

MLS Multicast statistics:
  Flow install Ack: 996508
  Flow install Nack: 1
  Flow update Ack: 1415959
  Flow update Nack: 0
  Flow delete Ack: 774953
  Complete flow install Ack: 958469

Router#
```

Configuring IPv4 Bidirectional PIM

These sections describe how to configure IPv4 bidirectional protocol independent multicast (PIM):

- [Enabling IPv4 Bidirectional PIM Globally, page 29-22](#)
- [Configuring the Rendezvous Point for IPv4 Bidirectional PIM Groups, page 29-23](#)
- [Setting the IPv4 Bidirectional PIM Scan Interval, page 29-24](#)
- [Displaying IPv4 Bidirectional PIM Information, page 29-24](#)

Enabling IPv4 Bidirectional PIM Globally

To enable IPv4 bidirectional PIM, perform this task:

Command	Purpose
Router(config)# ip pim bidir-enable	Enables IPv4 bidirectional PIM globally on the switch.
Router(config)# no ip pim bidir-enable	Disables IPv4 bidirectional PIM globally on the switch.

This example shows how to enable IPv4 bidirectional PIM on the switch:

```
Router(config)# ip pim bidir-enable
Router(config)#
```

Configuring the Rendezvous Point for IPv4 Bidirectional PIM Groups

To statically configure the rendezvous point for an IPv4 bidirectional PIM group, perform this task:

	Command	Purpose
Step 1	Router(config)# ip pim rp-address <i>ip_address</i> <i>access_list</i> [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 <i>access-list</i> permit deny <i>ip_address</i>	Configures an access list.
Step 3	Router(config)# ip pim send-rp-announce <i>type</i> <i>number</i> scope <i>tvl_value</i> [group-list <i>access-list</i>] [interval <i>seconds</i>] [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 <i>access-list-name</i> permit deny <i>ip_address</i>	Configures a standard IP access list.
Step 5	Router(config)# mls ip multicast	Enables MLS IP multicast.

This example shows how to configure a static rendezvous point for an IPv4 bidirectional PIM group:

```
Router(config)# ip pim rp-address 10.0.0.1 10 bidir override
Router(config)# access-list 10 permit 224.1.0.0 0.0.255.255
Router(config)# ip pim send-rp-announce Loopback0 scope 16 group-list c21-rp-list-0 bidir
Router(config)# ip access-list standard c21-rp-list-0 permit 230.31.31.1 0.0.255.255
```

Setting the IPv4 Bidirectional PIM Scan Interval

You can specify the interval between the IPv4 bidirectional PIM RP Reverse Path Forwarding (RPF) scans.

To set the IPv4 bidirectional PIM RP RPF scan interval, perform this task:

Command	Purpose
Router(config)# mls ip multicast bidir gm-scan-interval <i>interval</i>	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 task:

Command	Purpose
Router# show ip pim rp mapping [<i>in-use</i>]	Displays mappings between PIM groups and rendezvous points and shows learned rendezvous points in use.
Router# show mls ip multicast rp-mapping [<i>rp_address</i>]	Displays PIM group to active rendezvous points mappings.
Router# show mls ip multicast rp-mapping gm-cache	Displays information based on the group/mask ranges in the RP mapping cache.
Router# show mls ip multicast rp-mapping df-cache	Displays information based on the DF list in RP mapping cache.
Router# show mls ip multicast bidir	Displays IPv4 bidirectional PIM information.
Router# show ip mroute	Displays information about the multicast routing table.

This example shows how to display information about the PIM group and rendezvous point mappings:

```
Router# show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent
Group(s) 230.31.0.0/16
  RP 60.0.0.60 (?), v2v1, bidir
    Info source:60.0.0.60 (?), elected via Auto-RP
      Uptime:00:03:47, expires:00:02:11
  RP 50.0.0.50 (?), v2v1, bidir
    Info source:50.0.0.50 (?), via Auto-RP
      Uptime:00:03:04, expires:00:02:55
  RP 40.0.0.40 (?), v2v1, bidir
    Info source:40.0.0.40 (?), via Auto-RP
      Uptime:00:04:19, expires:00:02:38
```

This example shows how to display information in the IP multicast routing table that is related to IPv4 bidirectional PIM:

```
Router# show ip mroute bidirectional
(*, 225.1.3.0), 00:00:02/00:02:57, RP 3.3.3.3, flags:BC
  Bidir-Upstream:GigabitEthernet2/1, RPF nbr 10.53.1.7, RPF-MFD
  Outgoing interface list:
    GigabitEthernet2/1, Bidir-Upstream/Sparse-Dense, 00:00:02/00:00:00,H
    Vlan30, Forward/Sparse-Dense, 00:00:02/00:02:57, H

(*, 225.1.2.0), 00:00:04/00:02:55, RP 3.3.3.3, flags:BC
  Bidir-Upstream:GigabitEthernet2/1, RPF nbr 10.53.1.7, RPF-MFD
  Outgoing interface list:
    GigabitEthernet2/1, Bidir-Upstream/Sparse-Dense, 00:00:04/00:00:00,H
    Vlan30, Forward/Sparse-Dense, 00:00:04/00:02:55, H

(*, 225.1.4.1), 00:00:00/00:02:59, RP 3.3.3.3, flags:BC
  Bidir-Upstream:GigabitEthernet2/1, RPF nbr 10.53.1.7, RPF-MFD
  Outgoing interface list:
    GigabitEthernet2/1, Bidir-Upstream/Sparse-Dense, 00:00:00/00:00:00,H
    Vlan30, Forward/Sparse-Dense, 00:00:00/00:02:59, H
```

This example shows how to display information related to a specific multicast route. In the output below, the arrow in the margin points to information about a particular short cut:

```
Router# show ip mroute 239.1.1.2 4.4.4.4
IP Multicast Routing Table
Flags:D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
       Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags:H - Hardware switched
Timers:Uptime/Expires
Interface state:Interface, Next-Hop or VCD, State/Mode

(4.4.4.4, 239.1.1.2), 1d02h/00:03:20, flags:FTZ
  Incoming interface:Loopback0, RPF nbr 0.0.0.0, Partial-SC
  Outgoing interface list:
    Vlan10, Forward/Sparse-Dense, 1d02h/00:02:39 (ttl-threshold 5)
```

This example shows how to display the entries for a specific multicast group address:

```
Router# show mls ip multicast group 230.31.31.1
Multicast hardware switched flows:
(*, 230.31.31.1) Incoming interface:Vlan611, Packets switched:1778
Hardware switched outgoing interfaces:Vlan131 Vlan151 Vlan415 Gi4/16 Vlan611
RPF-MFD installed
```

This example shows how to display PIM group to active rendezvous points mappings:

```
Router# show mls ip multicast rp-mapping
State:H - Hardware Switched, I - Install Pending, D - Delete Pending, Z - Zombie

RP Address      State      RPF      DF-count  GM-count
60.0.0.60       H          V1611    4         1
```

This example shows how to display information based on the group/mask ranges in the RP mapping cache:

```
Router# show mls ip multicast rp-mapping gm-cache
State:H - Hardware Switched, I - Install Pending, D - Delete Pending,
      Z - Zombie
```

RP Address	State	Group	Mask	State	Packet/Byte-count
60.0.0.60	H	230.31.0.0	255.255.0.0	H	100/6400

This example shows how to display information about specific MLS IP multicasting groups:

```
Router# show mls ip multicast rp-mapping df-cache
```

State:H - Hardware Switched, I - Install Pending, D - Delete Pending, Z - Zombie

RP Address	State	DF	State
60.0.0.60	H	Vl131	H
60.0.0.60	H	Vl151	H
60.0.0.60	H	Vl415	H
60.0.0.60	H	Gi4/16	H

Using IPv4 Debug Commands

[Table 29-2](#) describes IPv4 multicast Layer 3 switching debug commands that you can use to troubleshoot IP multicast Layer 3 switching problems.

Table 29-2 IP Multicast Layer 3 Switching Debug Commands

Command	Description
[no] debug mls ip multicast events	Displays IP multicast Layer 3 switching events.
[no] debug mls ip multicast errors	Turns on debug messages for multicast MLS-related errors.
[no] debug mls ip multicast group <i>group_id</i> <i>group_mask</i>	Turns on debugging for a subset of flows.
[no] debug mls ip multicast messages	Displays IP multicast Layer 3 switching messages from and to hardware switching engine.
[no] debug mls ip multicast all	Turns on all IP multicast Layer 3 switching messages.
[no] debug mdss errors	Turns on MDSS ¹ error messages.
[no] debug mdss events	Displays MDSS-related events for debugging.
[no] debug mdss events mroute-bidir	Displays IPv4 bidirectional PIM MDSS events for debugging.
[no] debug mdss all	Displays all MDSS messages.
[no] debug ip pim df <i>ip_address</i>	Displays the DF election for a given rendezvous point for debug purposes.

1. MDSS = Multicast Distributed Switching Services

Clearing IPv4 Multicast Layer 3 Switching Statistics

To clear IP multicast Layer 3 switching statistics, perform this task:

Command	Purpose
Router# clear mls ip multicast statistics	Clears IP multicast Layer 3 switching statistics.

This example shows how to clear IP multicast Layer 3 switching statistics:

```
Router# clear mls ip multicast statistics
```

The **show mls multicast statistics** command displays a variety of information about the multicast flows being handled by the PFC. You can display entries based on any combination of the participating MSFC, the VLAN, the multicast group address, or the multicast traffic source. For an example of the **show mls ip multicast statistics** command, see the [“Displaying IPv4 Multicast Layer 3 Switching Statistics” section on page 29-21](#).



Configuring MLDv2 Snooping for IPv6 Multicast Traffic

This chapter describes how to configure Multicast Listener Discovery version 2 (MLDv2) snooping for IPv6 multicast traffic on the Catalyst 6500 series switches. Release 12.2(18)SX and later releases support MLDv2 snooping on the Supervisor Engine 720 with all versions of the PFC3.



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL: <http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- To constrain IPv4 Multicast traffic, see [Chapter 31, “Configuring IGMP Snooping for IPv4 Multicast Traffic.”](#)

This chapter consists of these sections:

- [Understanding How MLDv2 Snooping Works, page 30-1](#)
- [Default MLDv2 Snooping Configuration, page 30-7](#)
- [MLDv2 Snooping Configuration Guidelines and Restrictions, page 30-7](#)
- [MLDv2 Snooping Querier Configuration Guidelines and Restrictions, page 30-8](#)
- [Enabling the MLDv2 Snooping Querier, page 30-8](#)
- [Configuring MLDv2 Snooping, page 30-9](#)

Understanding How MLDv2 Snooping Works

These sections describe MLDv2 snooping:

- [MLDv2 Snooping Overview, page 30-2](#)
- [MLDv2 Messages, page 30-2](#)
- [Source-Based Filtering, page 30-3](#)
- [Explicit Host Tracking, page 30-3](#)
- [MLDv2 Snooping Proxy Reporting, page 30-3](#)
- [Joining an IPv6 Multicast Group, page 30-4](#)

- [Leaving a Multicast Group](#), page 30-6
- [Understanding the MLDv2 Snooping Querier](#), page 30-7

MLDv2 Snooping Overview

MLDv2 snooping allows Catalyst 6500 series switches to examine MLDv2 packets and make forwarding decisions based on their content.

You can configure the switch to use MLDv2 snooping in subnets that receive MLDv2 queries from either MLDv2 or the MLDv2 snooping querier. MLDv2 snooping constrains IPv6 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv6 multicast traffic only to those ports that want to receive it.

MLDv2, which runs at Layer 3 on a multicast router, generates Layer 3 MLDv2 queries in subnets where the multicast traffic needs to be routed. For information about MLDv2, see this publication:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/sa_mcast.htm

You can configure the MLDv2 snooping querier on the switch to support MLDv2 snooping in subnets that do not have any multicast router interfaces. For more information about the MLDv2 snooping querier, see the [“Enabling the MLDv2 Snooping Querier”](#) section on page 30-8.

MLDv2 (on a multicast router) or the MLDv2 snooping querier (on the Supervisor Engine 720) sends out periodic general MLDv2 queries that the switch forwards through all ports in the VLAN, and to which hosts respond. MLDv2 snooping monitors the Layer 3 MLDv2 traffic.



Note

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 switch is in report-suppression mode, the multicast router might not be able to track all the hosts accessed through a VLAN interface.

MLDv2 Snooping Proxy Reporting

Because MLDv2 does not have report suppression, all the hosts send their complete multicast group membership information to the multicast router in response to queries. The switch snoops these responses, updates the database and forwards the reports to the multicast router. To prevent the multicast router from becoming overloaded with reports, MLDv2 snooping does proxy reporting.

Proxy reporting forwards only the first report for a multicast group to the router and suppresses all other reports for the same multicast group.

Proxy reporting processes solicited and unsolicited reports. Proxy reporting is enabled and cannot be disabled.

**Note**

Disabling explicit host tracking disables fast-leave processing and proxy reporting.

Joining an IPv6 Multicast Group

Hosts join IPv6 multicast groups either by sending an unsolicited MLDv2 report or by sending an MLDv2 report in response to a general query from an IPv6 multicast router (the switch forwards general queries from IPv6 multicast routers to all ports in a VLAN). The switch snoops these reports.

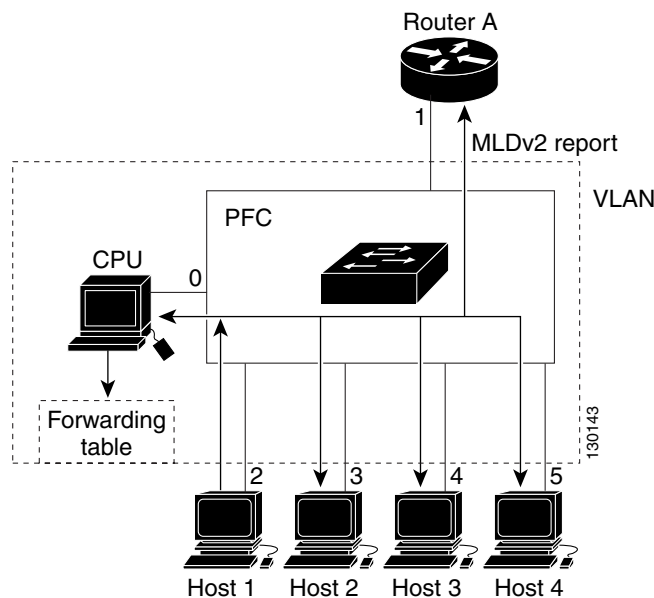
In response to a snooped MLDv2 report, the switch creates an entry in its Layer 2 forwarding table for the VLAN on which the report was received. When other hosts that are interested in this multicast traffic send MLDv2 reports, the switch snoops their reports and adds them to the existing Layer 2 forwarding table entry. The switch creates only one entry per VLAN in the Layer 2 forwarding table for each multicast group for which it snoops an MLDv2 report.

MLDv2 snooping suppresses all but one of the host reports per multicast group and forwards this one report to the IPv6 multicast router.

The switch forwards multicast traffic for the multicast group specified in the report to the interfaces where reports were received (see [Figure 30-1](#)).

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 30-1 Initial MLDv2 Listener Report



Multicast router A sends an MLDv2 general query to the switch, which forwards the query to ports 2 through 5 (all members of the same VLAN). Host 1 wants to join an IPv6 multicast group and multicasts an MLDv2 report to the group with the equivalent MAC destination address of 0x0100.5E01.0203.

When the switch snoops the MLDv2 report multicast by Host 1, the switch uses the information in the MLDv2 report to create a forwarding-table entry, as shown in Table 30-1, that includes the port numbers of Host 1, the multicast router, and the switch.

Table 30-1 MLDv2 Snooping Forwarding Table

Destination MAC Address	Type of Packet	Ports
0100.5exx.xxxx	MLDv2	0
0100.5e01.0203	!MLDv2	1, 2

The switch hardware can distinguish MLDv2 information packets from other packets for the multicast group. The first entry in the table tells the switch to send only MLDv2 packets to the CPU. This prevents the switch from becoming overloaded with multicast frames. The second entry tells the switch 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 30-2), the switch snoops 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 MLDv2 messages only to the switch, the message is not flooded to other ports. Any known multicast traffic is forwarded to the group and not to the switch.

Figure 30-2 Second Host Joining a Multicast Group

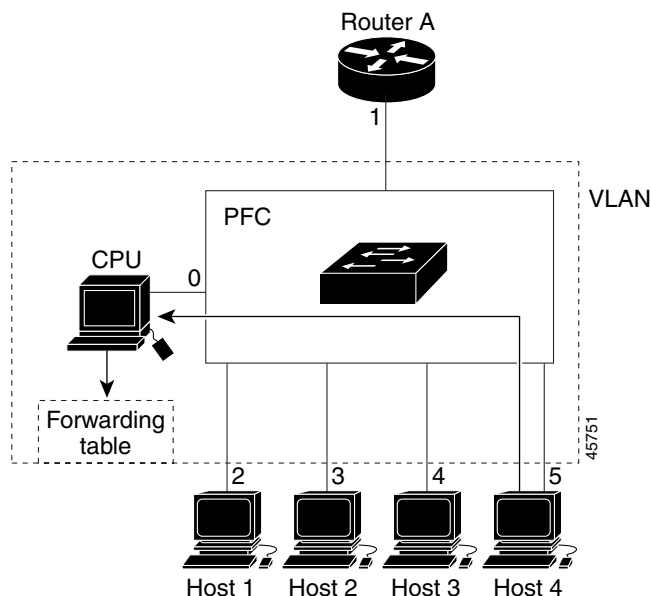


Table 30-2 Updated MLDv2 Snooping Forwarding Table

Destination MAC Address	Type of Packet	Ports
0100.5exx.xxxx	MLDv2	0
0100.5e01.0203	!MLDv2	1, 2, 5

Leaving a Multicast Group

These sections describe leaving a multicast group:

- [Normal Leave Processing, page 30-6](#)
- [Fast-Leave Processing, page 30-6](#)

Normal Leave Processing

Interested hosts must continue to respond to the periodic MLDv2 general queries. As long as at least one host in the VLAN responds to the periodic MLDv2 general queries, the multicast router continues forwarding the multicast traffic to the VLAN. When hosts want to leave a multicast group, they can either ignore the periodic MLDv2 general queries (called a “silent leave”), or they can send an MLDv2 filter mode change record.

When MLDv2 snooping receives a filter mode change record from a host that configures the EXCLUDE mode for a group, MLDv2 snooping sends out a MAC-addressed general query to determine if any other hosts connected to that interface are interested in traffic for the specified multicast group.

If MLDv2 snooping does not receive an MLDv2 report in response to the general query, MLDv2 snooping assumes that no other hosts connected to the interface are interested in receiving traffic for the specified multicast group, and MLDv2 snooping removes the interface from its Layer 2 forwarding table entry for the specified multicast group.

If the filter mode change record was from the only remaining interface with hosts interested in the group, and MLDv2 snooping does not receive an MLDv2 report in response to the general query, MLDv2 snooping removes the group entry and relays the MLDv2 filter mode change record to the multicast router. If the multicast router receives no reports from a VLAN, the multicast router removes the group for the VLAN from its MLDv2 cache.

The interval for which the switch waits before updating the table entry is called the “last member query interval.” To configure the interval, enter the **ipv6 mld snooping last-member-query-interval** *interval* command.

Fast-Leave Processing

Fast-leave processing is enabled by default. To disable fast-leave processing, turn off explicit-host tracking.

Fast-leave processing is implemented by maintaining source-group based membership information in software while also allocating LTL indexes on a MAC GDA basis.

When fast-leave processing is enabled, hosts send BLOCK_OLD_SOURCES{src-list} messages for a specific group when they no longer want to receive traffic from that source. When the switch receives such a message from a host, it parses the list of sources for that host for the given group. If this source list is exactly the same as the source list received in the leave message, the switch removes the host from the LTL index and stops forwarding this multicast group traffic to this host.

If the source lists do not match, the switch does not remove the host from the LTL index until the host is no longer interested in receiving traffic from any source.

**Note**

Disabling explicit host tracking disables fast-leave processing and proxy reporting.

Understanding the MLDv2 Snooping Querier

Use the MLDv2 snooping querier to support MLDv2 snooping in a VLAN where PIM and MLDv2 are not configured because the multicast traffic does not need to be routed.

In a network where IP multicast routing is configured, the IP multicast router acts as the MLDv2 querier. If the IP-multicast traffic in a VLAN only needs to be Layer 2 switched, an IP-multicast router is not required, but without an IP-multicast router on the VLAN, you must configure another switch as the MLDv2 querier so that it can send queries.

When enabled, the MLDv2 snooping querier sends out periodic MLDv2 queries that trigger MLDv2 report messages from the switch that wants to receive IP multicast traffic. MLDv2 snooping listens to these MLDv2 reports to establish appropriate forwarding.

You can enable the MLDv2 snooping querier on all the Catalyst 6500 series switches in the VLAN, but for each VLAN that is connected to switches that use MLDv2 to report interest in IP multicast traffic, you must configure at least one switch as the MLDv2 snooping querier.

You can configure a switch to generate MLDv2 queries on a VLAN regardless of whether or not IP multicast routing is enabled.

Default MLDv2 Snooping Configuration

Table 30-3 shows the default MLDv2 snooping configuration.

Table 30-3 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 15, “Configuring VLANs”](#)).
- Configure an IPv6 address on the VLAN interface (see [Chapter 23, “Configuring Layer 3 Interfaces”](#)). When enabled, the MLDv2 snooping querier uses the IPv6 address as the query source address.
- If there is no IPv6 address configured on the VLAN interface, the MLDv2 snooping querier does not start. The MLDv2 snooping querier disables itself if the IPv6 address is cleared. When enabled, the MLDv2 snooping querier restarts if you configure an IPv6 address.
- When enabled, the MLDv2 snooping querier does not start if it detects MLDv2 traffic from an IPv6 multicast router.
- When enabled, the MLDv2 snooping querier starts after 60 seconds with no MLDv2 traffic detected from an IPv6 multicast router.
- When enabled, the MLDv2 snooping querier disables itself if it detects MLDv2 traffic from an IPv6 multicast router.
- QoS does not support MLDv2 packets when MLDv2 snooping is enabled.
- You can enable the MLDv2 snooping querier on all the Catalyst 6500 series switches in the VLAN that support it. One switch is elected as the querier.

Enabling the MLDv2 Snooping Querier

Use the MLDv2 snooping querier to support MLDv2 snooping in a VLAN where PIM and MLDv2 are not configured because the multicast traffic does not need to be routed.

To enable the MLDv2 snooping querier in a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects the VLAN interface.
Step 2	Router(config-if)# ipv6 address <i>prefix/prefix_length</i>	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 <i>vlan_ID</i> 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 30-8).

These sections describe how to configure MLDv2 snooping:

- [Enabling MLDv2 Snooping, page 30-9](#)
- [Configuring a Static Connection to a Multicast Receiver, page 30-10](#)
- [Enabling Fast-Leave Processing, page 30-12](#)
- [Configuring Explicit Host Tracking, page 30-13](#)
- [Configuring Report Suppression, page 30-13](#)
- [Displaying MLDv6 Snooping Information, page 30-14](#)



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 <i>vlan_ID</i> 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 <i>vlan_ID</i>	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 <i>vlan_ID</i> 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 <i>mac_addr</i> vlan <i>vlan_id</i> interface <i>type</i> ¹ <i>slot/port</i> [disable-snooping]	Configures a static connection to a multicast receiver.
	Router(config)# no mac-address-table static <i>mac_addr</i> vlan <i>vlan_id</i>	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 <i>mac_addr</i>	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 <i>vlan_ID</i>	Selects the VLAN interface.
Step 2	Router(config-if)# ipv6 mld snooping mrouter interface <i>type</i> ¹ <i>slot/port</i>	Configures a static connection to a multicast router.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show ipv6 mld snooping mrouter	Verifies the configuration.

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

The interface to the router must be in the VLAN where you are entering the command, the interface must be administratively up, and the line protocol must be up.

This example shows how to configure a static connection to a multicast router:

```
Router(config-if)# ipv6 mld snooping mrouter interface fastethernet 5/6
Router(config-if)#
```

Configuring the MLD Snooping Query Interval

You can configure the interval for which the switch waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group.



Note

When both MLD snooping fast-leave processing and the MLD snooping query interval are configured, fast-leave processing takes precedence.

To configure the interval for the MLD snooping queries sent by the switch, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects a VLAN interface.
Step 2	Router(config-if)# ipv6 mld snooping last-member-query-interval <i>interval</i>	Configures the interval for the IGMP queries sent by the switch. Default is 1 second. Valid range is 1000 to 9990 milliseconds.
	Router(config-if)# no ipv6 mld snooping last-member-query-interval	Reverts to the default value.
Step 3	Router# show ipv6 mld interface vlan <i>vlan_ID</i> 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 <i>vlan_ID</i>	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 <i>vlan_ID</i> 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 <i>vlan_ID</i>	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 <i>vlan_ID</i>	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 <i>vlan_ID</i>	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 <i>vlan_ID</i>	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 <i>vlan_ID</i> include report-suppression	Displays the status of report suppression.

This example shows how to enable explicit host tracking:

```
Router(config)# interface vlan 25
Router(config-if)# ipv6 mld snooping report-suppression
Router(config-if)# end
Router# Router# show ipv6 mld interface vlan 25 | include report-suppression
MLD snooping report-suppression is enabled
```

Displaying MLDv6 Snooping Information

These sections describe displaying MLDv6 snooping information:

- [Displaying Multicast Router Interfaces, page 30-14](#)
- [Displaying MAC Address Multicast Entries, page 30-14](#)
- [Displaying MLDv2 Snooping Information for a VLAN Interface, page 30-15](#)

Displaying Multicast Router Interfaces

When you enable IGMP snooping, the switch automatically learns to which interface the multicast routers are connected.

To display multicast router interfaces, perform this task:

Command	Purpose
Router# show ipv6 mld snooping mrouter <i>vlan_ID</i>	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 <i>vlan_ID</i> [<i>count</i>]	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 <i>vlan_ID</i> } { mrouter [vlan <i>vlan_ID</i>]} } { report-suppression vlan <i>vlan_ID</i> } { statistics vlan <i>vlan_ID</i> }	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 statictics 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    -
```




Configuring IGMP Snooping for IPv4 Multicast Traffic

This chapter describes how to configure Internet Group Management Protocol (IGMP) snooping for IPv4 multicast traffic on the Catalyst 6500 series switches.



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- To constrain IPv6 Multicast traffic, see [Chapter 30, “Configuring MLDv2 Snooping for IPv6 Multicast Traffic.”](#)

This chapter consists of these sections:

- [Understanding How IGMP Snooping Works, page 31-1](#)
- [Default IGMP Snooping Configuration, page 31-7](#)
- [IGMP Snooping Configuration Guidelines and Restrictions, page 31-8](#)
- [IGMP Snooping Querier Configuration Guidelines and Restrictions, page 31-8](#)
- [Enabling the IGMP Snooping Querier, page 31-9](#)
- [Configuring IGMP Snooping, page 31-9](#)

Understanding How IGMP Snooping Works

These sections describe IGMP snooping:

- [IGMP Snooping Overview, page 31-2](#)
- [Joining a Multicast Group, page 31-2](#)
- [Leaving a Multicast Group, page 31-4](#)
- [Understanding the IGMP Snooping Querier, page 31-5](#)
- [Understanding IGMP Version 3 Support, page 31-5](#)

IGMP Snooping Overview

You can configure the switch to use IGMP snooping in subnets that receive IGMP queries from either IGMP or the IGMP snooping querier. IGMP snooping constrains IPv4 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv4 multicast traffic only to those ports that want to receive it.

IGMP, which runs at Layer 3 on a multicast router, generates Layer 3 IGMP queries in subnets where the multicast traffic needs to be routed. For information about IGMP, see [Chapter 29, “Configuring IPv4 Multicast Layer 3 Switching.”](#)

You can configure the IGMP snooping querier on the switch to support IGMP snooping in subnets that do not have any multicast router interfaces. For more information about the IGMP snooping querier, see the [“Enabling the IGMP Snooping Querier”](#) section on page 31-9.

IGMP (on a multicast router) or the IGMP snooping querier (on the supervisor engine) sends out periodic general IGMP queries that the switch forwards through all ports in the VLAN and to which hosts respond. IGMP snooping monitors the Layer 3 IGMP traffic.

**Note**

If a multicast group has only sources and no receivers in a VLAN, IGMP snooping constrains the multicast traffic to only the multicast router ports.

Joining a Multicast Group

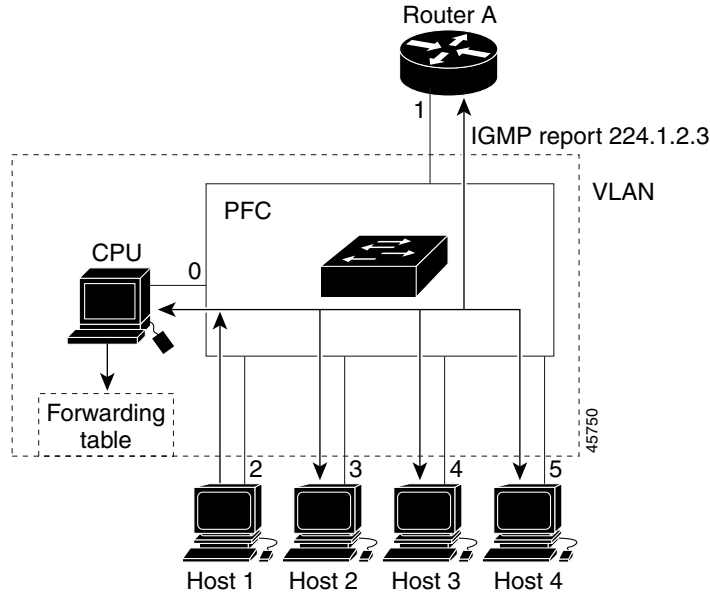
Hosts join multicast groups either by sending an unsolicited IGMP join message or by sending an IGMP join message in response to a general query from a multicast router (the switch forwards general queries from multicast routers to all ports in a VLAN).

In response to an IGMP join request, the switch creates an entry in its Layer 2 forwarding table for the VLAN on which the join request was received. When other hosts that are interested in this multicast traffic send IGMP join requests, the switch adds them to the existing Layer 2 forwarding table entry. The switch creates only one entry per VLAN in the Layer 2 forwarding table for each multicast group for which it receives an IGMP join request.

IGMP snooping suppresses all but one of the host join messages per multicast group and forwards this one join message to the multicast router.

The switch forwards multicast traffic for the multicast group specified in the join message to the interfaces where join messages were received (see [Figure 31-1](#)).

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 31-1 Initial IGMP Join Message

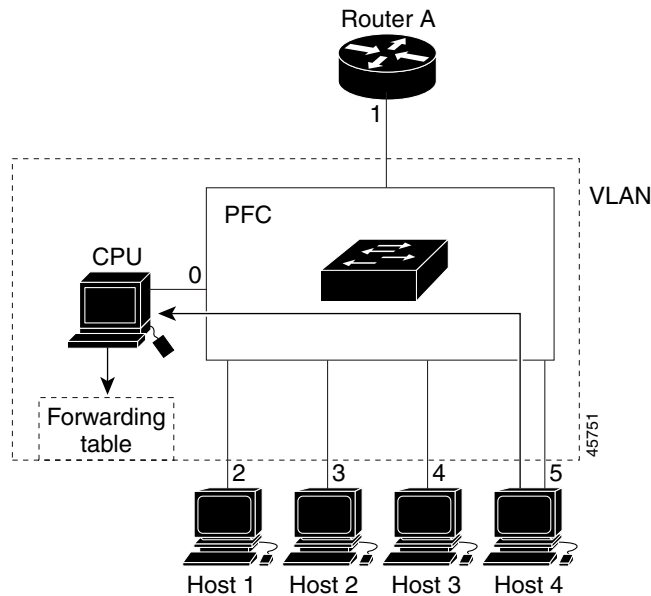
Multicast router A sends a general query to the switch, which forwards the query to ports 2 through 5 (all members of the same VLAN). Host 1 wants to join multicast group 224.1.2.3 and multicasts an IGMP membership report (IGMP join message) to the group with the equivalent MAC destination address of 0x0100.5E01.0203. When the CPU receives the IGMP report multicast by Host 1, the CPU uses the information in the IGMP report to set up a forwarding-table entry, as shown in [Table 31-1](#), that includes the port numbers of Host 1, the multicast router, and the switch internal CPU.

Table 31-1 IGMP Snooping Forwarding Table

Destination Address	Type of Packet	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1, 2

The switch hardware can distinguish IGMP information packets from other packets for the multicast group. The first entry in the table tells the switching engine to send only IGMP packets to the CPU. This prevents the CPU from becoming overloaded with multicast frames. The second entry tells the switching engine to send frames addressed to the 0x0100.5E01.0203 multicast MAC address that are not IGMP packets (!IGMP) to the multicast router and to the host that has joined the group.

If another host (for example, Host 4) sends an unsolicited IGMP join message for the same group ([Figure 31-2](#)), the CPU receives that message and adds the port number of Host 4 to the forwarding table as shown in [Table 31-2](#). 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 31-2 Second Host Joining a Multicast Group**Table 31-2 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 31-4](#)
- [Fast-Leave Processing, page 31-5](#)

Normal Leave Processing

Interested hosts must continue to respond to the periodic general IGMP queries. As long as at least one host in the VLAN responds to the periodic general IGMP queries, the multicast router continues forwarding the multicast traffic to the VLAN. When hosts want to leave a multicast group, they can either ignore the periodic general IGMP queries (called a “silent leave”), or they can send a group-specific IGMPv2 leave message.

When IGMP snooping receives a group-specific IGMPv2 leave message from a host, it sends out a MAC-based general query to determine if any other devices connected to that interface are interested in traffic for the specific multicast group. If IGMP snooping does not receive an IGMP Join message in response to the general query, it assumes that no other devices connected to the interface are interested in receiving traffic for this multicast group, and it removes the interface from its Layer 2 forwarding table entry for that multicast group. If the leave message was from the only remaining interface with hosts interested in the group and IGMP snooping does not receive an IGMP Join in response to the general

query, it removes the group entry and relays the IGMP leave to the multicast router. If the multicast router receives no reports from a VLAN, the multicast router removes the group for the VLAN from its IGMP cache.

The interval for which the switch waits before updating the table entry is called the “last member query interval.” To configure the interval, enter the **ip igmp snooping last-member-query-interval** *interval* command.

Fast-Leave Processing

IGMP snooping fast-leave processing allows IGMP snooping to remove a Layer 2 LAN interface from the forwarding-table entry without first sending out IGMP group-specific queries to the interface. Upon receiving a group-specific IGMPv2 leave message, IGMP snooping immediately removes the interface from the Layer 2 forwarding table entry for that multicast group, unless a multicast router was learned on the port. Fast-leave processing improves bandwidth management for all hosts on a switched network.



Note

Use fast-leave processing only on VLANs where only one host is connected to each Layer 2 LAN port. If fast-leave is enabled in VLANs where more than one host is connected to a Layer 2 LAN port, some hosts might be dropped inadvertently. Fast-leave processing is supported only with IGMP version 2 hosts.

Understanding the IGMP Snooping Querier

Use the IGMP snooping querier to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not need to be routed.

In a network where IP multicast routing is configured, the IP multicast router acts as the IGMP querier. If the IP-multicast traffic in a VLAN only needs to be Layer 2 switched, an IP-multicast router is not required, but without an IP-multicast router on the VLAN, you must configure another switch as the IGMP querier so that it can send queries.

When enabled, the IGMP snooping querier sends out periodic IGMP queries that trigger IGMP report messages from the switch that wants to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

You can enable the IGMP snooping querier on all the Catalyst 6500 series switches in the VLAN, but for each VLAN that is connected to switches that use IGMP to report interest in IP multicast traffic, you must configure at least one switch as the IGMP snooping querier.

You can configure a switch to generate IGMP queries on a VLAN regardless of whether or not IP multicast routing is enabled.

Understanding IGMP Version 3 Support

IGMP snooping supports IGMP version 3. IGMP version 3 uses source-based filtering, which enables hosts and routers to specify which source addresses should be allowed or blocked for a specific multicast group. When you enable IGMP version 3 snooping on a Catalyst 6500 series switch, the system maintains IGMP version 3 states based on messages it receives for a particular group in a particular VLAN and either allows or blocks traffic based on the following information in these messages:

- Source lists
- Allow (include) or block (exclude) filtering options

Because the Layer 2 table is (MAC-group, VLAN) based, with IGMPv3 hosts it is preferable to have only a single multicast source per MAC-group.

**Note**

Source-based filtering for IGMP version 3 reports is not supported in hardware. The states are maintained only in software and used for explicit host tracking and statistics collection.

IGMPv3 Fast-Leave Processing

IGMP version 3 fast-leave processing is enabled by default. To disable IGMP version 3 fast-leave processing you must turn off explicit-host tracking.

Fast-leave processing with IGMPv3 is implemented by maintaining source-group based membership information in software while also allocating LTL indexes on a MAC GDA basis.

When fast-leave processing is enabled, hosts send `BLOCK_OLD_SOURCES{src-list}` messages for a specific group when they no longer want to receive traffic from that source. When the switch receives such a message from a host, it parses the list of sources for that host for the given group. If this source list is exactly the same as the source list received in the leave message, the switch removes the host from the LTL index and stops forwarding this multicast group traffic to this host.

If the source lists do not match, the switch does not remove the host from the LTL index until the host is no longer interested in receiving traffic from any source.

Proxy Reporting

Because IGMPv3 does not have report suppression, all the hosts send their complete membership information to the router in response to queries. The switch receives these responses, updates the database and forwards the reports to the router. To prevent the router from becoming overloaded with reports, you can configure the switch for proxy-reporting mode. In proxy reporting mode, the switch forwards only the first report for a channel to the router and suppresses all other reports for the same channel.

With IGMPv3 proxy reporting, the switch does proxy reporting for unsolicited reports, as well as for reports received in the general query interval. Proxy reporting is turned on by default. When you disable proxy reporting, the switch works in transparent mode and updates the IGMP snooping database as it receives reports and forwards this information to the upstream router, which can then explicitly track all reporting hosts.

To support a mix of IGMPv2 and IGMPv3 hosts, the switch converts the IGMPv2 report into a `EXCLUDE` mode report. You must configure the switch to support both IGMPv2 and IGMPv3 hosts.

**Note**

- Source-based filtering for IGMP version 3 reports is not supported in hardware. The states are maintained only in software and used for explicit host tracking and statistics collection.
- Turning off explicit host tracking disables fast-leave processing and proxy reporting.

Explicit Host Tracking

IGMPv3 supports explicit tracking of membership information on any port. The explicit-tracking database is used for fast-leave processing for IGMPv3 hosts, proxy reporting, and statistics collection. When explicit tracking is enabled on a VLAN, the IGMP snooping software processes the IGMPv3 report it receives from a host and builds an explicit-tracking database that contains the following information:

- The port connected to the host
- The channels reported by the host
- The filter mode for each group reported by the host
- The list of sources for each group reported by the hosts
- The router filter mode of each group
- For each group, the list of hosts requesting the source



Note

- Turning off explicit host tracking disables fast-leave processing and proxy reporting.
- When explicit tracking is enabled and the switch is working in proxy-reporting mode, the router may not be able to track all the hosts behind a VLAN interface.

Default IGMP Snooping Configuration

Table 31-3 shows the default IGMP snooping configuration.

Table 31-3 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; deprecated in Release 12.2(18)SXE and later releases

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 15, “Configuring VLANs”](#)).
- Configure an IP address on the VLAN interface (see [Chapter 23, “Configuring Layer 3 Interfaces”](#)). When enabled, the IGMP snooping querier uses the IP address as the query source address.
- If there is no IP address configured on the VLAN interface, the IGMP snooping querier does not start. The IGMP snooping querier disables itself if the IP address is cleared. When enabled, the IGMP snooping querier restarts if you configure an IP address.
- The IGMP snooping querier supports IGMP version 2.
- When enabled, the IGMP snooping querier does not start if it detects IGMP traffic from a multicast router.
- When enabled, the IGMP snooping querier starts after 60 seconds with no IGMP traffic detected from a multicast router.
- When enabled, the IGMP snooping querier disables itself if it detects IGMP traffic from a multicast router.
- QoS does not support IGMP packets when IGMP snooping is enabled.
- You can enable the IGMP snooping querier on all the Catalyst 6500 series switches in the VLAN. One switch is elected as the querier.

**Note**

When you are in configuration mode you can enter EXEC mode commands by entering the **do** keyword before the EXEC mode command.

Enabling the IGMP Snooping Querier

Use the IGMP snooping querier to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not need to be routed.

To enable the IGMP snooping querier in a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects the VLAN interface.
Step 2	Router(config-if)# ip address <i>ip_address</i> <i>subnet_mask</i>	Configures the IP address and IP subnet.
Step 3	Router(config-if)# ip igmp snooping querier	Enables the IGMP snooping querier.
	Router(config-if)# no ip igmp snooping querier	Disables the IGMP snooping querier.
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show ip igmp interface vlan <i>vlan_ID</i> 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 29, “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 31-9).

IGMP snooping allows Catalyst 6500 series switches to examine IGMP packets and make forwarding decisions based on their content.

These sections describe how to configure IGMP snooping:

- [Enabling IGMP Snooping, page 31-10](#)
- [Configuring a Static Connection to a Multicast Receiver, page 31-11](#)
- [Configuring a Multicast Router Port Statically, page 31-11](#)
- [Configuring the IGMP Snooping Query Interval, page 31-11](#)
- [Enabling IGMP Fast-Leave Processing, page 31-12](#)
- [Configuring Source Specific Multicast \(SSM\) Mapping, page 31-12](#)
- [Enabling SSM Safe Reporting, page 31-13](#)

- [Configuring IGMPv3 Explicit Host Tracking, page 31-13](#)
- [Displaying IGMP Snooping Information, page 31-14](#)

**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 <i>mac_addr</i> vlan <i>vlan_id</i> interface <i>type</i> ¹ <i>slot/port</i> [disable-snooping]	Configures a static connection to a multicast receiver.
	Router(config)# no mac-address-table static <i>mac_addr</i> vlan <i>vlan_id</i>	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 <i>mac_addr</i>	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 <i>type</i> ¹ <i>slot/port</i>	Configures a static connection to a multicast router.
Step 2	Router(config-if)# end	Exits configuration mode.
Step 3	Router# show ip igmp snooping mrouter	Verifies the configuration.

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

The interface to the router must be in the VLAN where you are entering the command, the interface must be administratively up, and the line protocol must be up.

This example shows how to configure a static connection to a multicast router:

```
Router(config-if)# ip igmp snooping mrouter interface fastethernet 5/6
Router(config-if)#
```

Configuring the IGMP Snooping Query Interval

You can configure the interval for which the switch waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group.



Note

When both IGMP fast-leave processing and the IGMP query interval are configured, fast-leave processing takes precedence.

To configure the interval for the IGMP snooping queries sent by the switch, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	Selects a VLAN interface.
Step 2	Router(config-if)# ip igmp snooping last-member-query-interval <i>interval</i>	Configures the interval for the IGMP snooping queries sent by the switch. Default is 1 second. Valid range is 100 to 999 milliseconds.
	Router(config-if)# no ip igmp snooping last	Reverts to the default value.

This example shows how to configure the IGMP snooping query interval:

```
Router(config-if)# ip igmp snooping last-member-query-interval 200
Router(config-if)# exit
Router# show ip igmp interface vlan 200 | include last
IGMP snooping last member query interval on this interface is 200 ms
```

Enabling IGMP Fast-Leave Processing

To enable IGMP fast-leave processing in a VLAN, perform this task:

	Command	Purpose
Step 1	Router(config)# interface vlan <i>vlan_ID</i>	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

- Release 12.2(18)SXD3 and later releases support SSM mapping.
- 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 deprecated in Release 12.2(18)SXE and later releases.

When you configure SSM safe reporting, the group mode is IGMPv3 even in the presence of IGMPv1 and IGMPv2 hosts.

To make sure the switch 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 <i>vlan_ID</i>	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 switch 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 <i>vlan_ID</i>	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	V125:1/2	16.27.2.3	INCLUDE
10.2.2.2/226.2.2.2	V125:1/2	16.27.2.3	INCLUDE

Displaying IGMP Snooping Information

These sections describe displaying IGMP snooping information:

- [Displaying Multicast Router Interfaces, page 31-14](#)
- [Displaying MAC Address Multicast Entries, page 31-14](#)
- [Displaying IGMP Snooping Information for a VLAN Interface, page 31-15](#)
- [Displaying IGMP Snooping Statistics, page 31-15](#)

Displaying Multicast Router Interfaces

When you enable IGMP snooping, the switch automatically learns to which interface the multicast routers are connected.

To display multicast router interfaces, perform this task:

Command	Purpose
Router# show ip igmp snooping mrouter <i>vlan_ID</i>	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 <i>vlan_ID</i> [<i>count</i>]	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 <i>vlan_ID</i>	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 <i>vlan_ID</i>	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#
```



Configuring PIM Snooping

This chapter describes how to configure protocol independent multicast (PIM) snooping on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How PIM Snooping Works, page 32-1](#)
- [Default PIM Snooping Configuration, page 32-4](#)
- [PIM Snooping Configuration Guidelines and Restrictions, page 32-4](#)
- [Configuring PIM Snooping, page 32-4](#)

Understanding How PIM Snooping Works

In networks where a Layer 2 switch interconnects several routers, such as an Internet exchange point (IXP), the switch floods IP multicast packets on all multicast router ports by default, even if there are no multicast receivers downstream. With PIM snooping enabled, the switch restricts multicast packets for each IP multicast group to only those multicast router ports that have downstream receivers joined to that group. When you enable PIM snooping, the switch learns which multicast router ports need to receive the multicast traffic within a specific VLAN by listening to the PIM hello messages, PIM join and prune messages, and bidirectional PIM designated forwarder-election messages.



Note

To use PIM snooping, you must enable IGMP snooping on the Catalyst 6500 series switch. IGMP snooping restricts multicast traffic that exits through the LAN ports to which hosts are connected. IGMP snooping does not restrict traffic that exits through the LAN ports to which one or more multicast routers are connected.

The following illustrations show the flow of traffic and flooding that results in networks without PIM snooping enabled and the flow of traffic and traffic restriction when PIM snooping is enabled.

[Figure 32-1](#) 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.

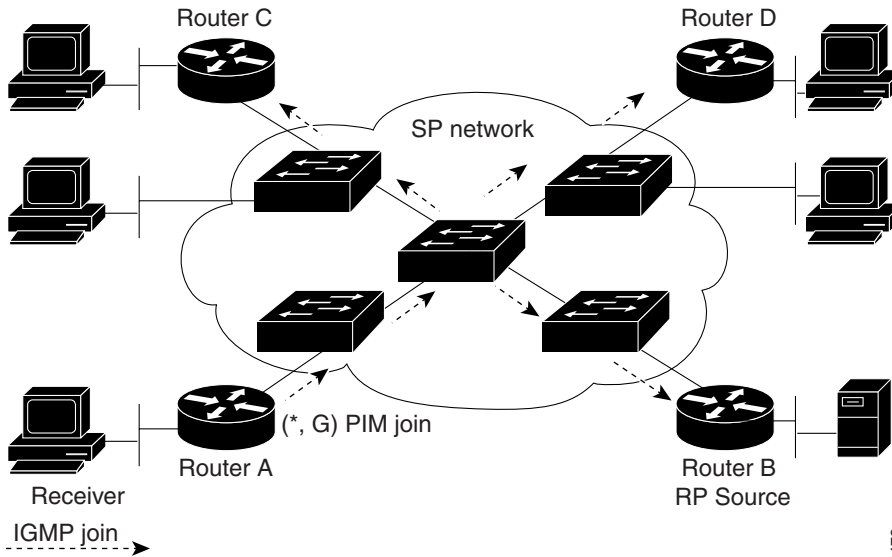
Figure 32-1 PIM Join Message Flow without PIM Snooping

Figure 32-2 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).

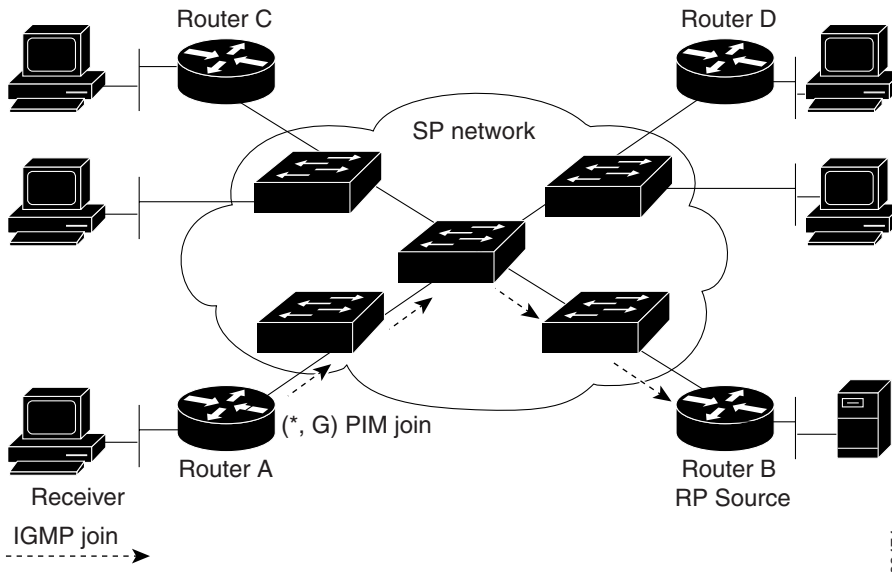
Figure 32-2 PIM Join Message Flow with PIM Snooping

Figure 32-3 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 32-3 Data Traffic Flow without PIM Snooping

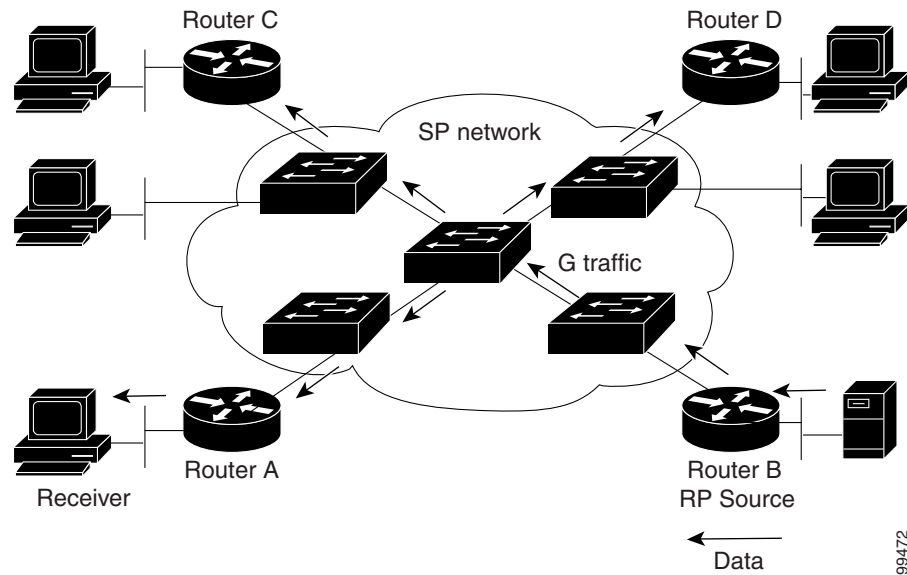
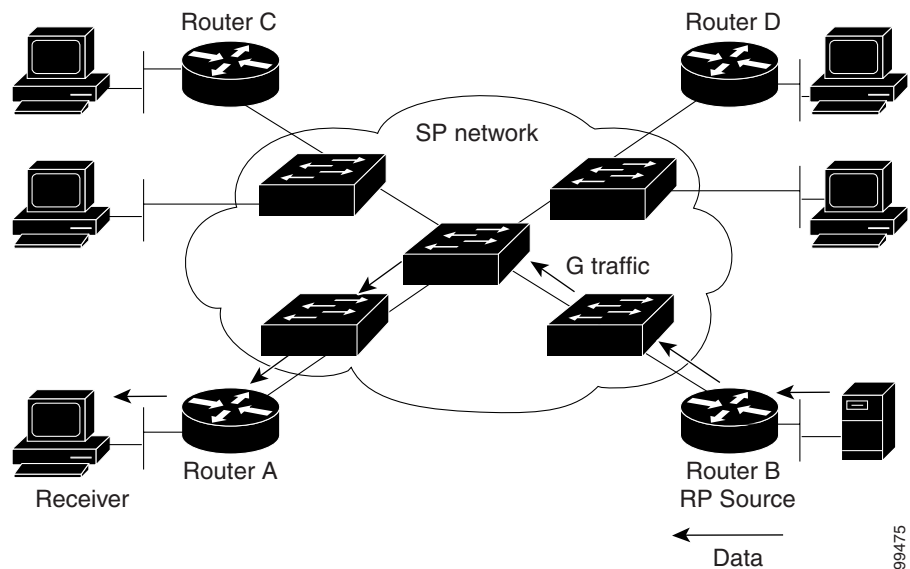


Figure 32-4 shows the flow of data traffic with PIM snooping enabled. In the figure, the switches forward the data traffic only to the router that needs to receive it (Router A).

Figure 32-4 Data Traffic Flow with PIM Snooping



Default PIM Snooping Configuration

PIM snooping is disabled by default.

PIM Snooping Configuration Guidelines and Restrictions

When configuring PIM snooping, follow these guidelines and restrictions:

- When you use the PIM-sparse mode (PIM-SM) feature, downstream routers only see traffic if they previously indicated interest through a PIM join or prune message. An upstream router only sees traffic if it was used as an upstream router during the PIM join or prune process.
- Join or prune messages are not flooded on all router ports but are sent only to the port corresponding to the upstream router mentioned in the payload of the join or prune message.
- Directly connected sources are supported for bidirectional PIM groups. Traffic from directly connected sources is forwarded to the designated router and designated forwarder for a VLAN. In some cases, a nondesignated router (NDR) can receive a downstream (S, G) join. For source-only networks, the initial unknown traffic is flooded only to the designated routers and designated forwarders.
- Dense group mode traffic is seen as unknown traffic and is dropped.
- The AUTO-RP groups (224.0.1.39 and 224.0.1.40) are always flooded.
- The switch snoops on designated forwarder election and maintains a list of all designated forwarder routers for various RPs for the VLAN. All traffic is sent to all designated forwarders which ensures that bidirectional functionality works properly.
- PIM snooping and IGMP snooping can be enabled at the same time in a VLAN. Either RGMP or PIM snooping can be enabled in a VLAN but not both.
- Any non-PIMv2 multicast router will receive all traffic.
- You can enable or disable PIM snooping on a per-VLAN basis.
- All mroute and router information is timed out based on the hold-time indicated in the PIM hello and join/prune control packets. All mroute state and neighbor information is maintained per VLAN.

Configuring PIM Snooping

These sections describe how to configure PIM snooping:

- [Enabling PIM Snooping Globally, page 32-5](#)
- [Enabling PIM Snooping in a VLAN, page 32-5](#)
- [Disabling PIM Snooping Designated-Router Flooding, page 32-6](#)

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 <i>vlan_ID</i>	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

By default, switches that have PIM snooping enabled will flood multicast traffic to the designated router (DR). This method of operation can send unnecessary multicast packets to the designated router. The network must carry the unnecessary traffic, and the designated router must process and drop the unnecessary traffic.

With Release 12.2(18)SXF and later releases, you can disable designated-router flooding to reduce the traffic sent over the network to the designated router. 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.

**Note**

Do not disable designated-router flooding on switches in a Layer 2 broadcast domain that supports multicast sources.

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
```



Configuring RGMP

This chapter supplements the information and procedures about Router-Port Group Management Protocol (RGMP) in the Release 12.2 publication at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcpt3/1cfrgmp.htm

This chapter consists of these sections:

- [Understanding How RGMP Works, page 33-1](#)
- [Default RGMP Configuration, page 33-2](#)
- [RGMP Configuration Guidelines and Restrictions, page 33-2](#)
- [Enabling RGMP on Layer 3 Interfaces, page 33-3](#)

Understanding How RGMP Works

RGMP constrains multicast traffic that exits the Catalyst 6500 series switch through ports to which only disinterested multicast routers are connected. RGMP reduces network congestion by forwarding multicast traffic to only those routers that are configured to receive it.



Note

To use RGMP, you must enable IGMP snooping on the Catalyst 6500 series switch. IGMP snooping constrains multicast traffic that exits through LAN ports to which hosts are connected. IGMP snooping does not constrain traffic that exits through LAN ports to which one or more multicast routers are connected.



Note

You must enable Protocol Independent Multicast (PIM) on all routers and switches for RGMP to work. Only PIM sparse mode is currently supported.

All routers on the network must be RGMP-capable. RGMP-capable routers send RGMP hello messages periodically. The RGMP hello message tells the Catalyst 6500 series switch not to send multicast data to the router unless an RGMP join message has also been sent to the Catalyst 6500 series switch from that router. When an RGMP join message is sent, the router is able to receive multicast data.

To stop receiving multicast data, a router must send an RGMP leave message to the Catalyst 6500 series switch. To disable RGMP on a router, the router must send an RGMP bye message to the Catalyst 6500 series switch.

[Table 33-1](#) provides a summary of the RGMP packet types.

Table 33-1 RGMP Packet Types

Description	Action
Hello	When RGMP is enabled on the router, no multicast data traffic is sent to the router by the Catalyst 6500 series switch unless an RGMP join is specifically sent for a group.
Bye	When RGMP is disabled on the router, all multicast data traffic is sent to the router by the Catalyst 6500 series switch.
Join	Multicast data traffic for a multicast MAC address from the Layer 3 group address G is sent to the router. These packets have group G in the Group Address field of the RGMP packet.
Leave	Multicast data traffic for the group G is not sent to the router. These packets have group G in the group address field of the RGMP packet.

Default RGMP Configuration

RGMP is permanently enabled on Layer 2 LAN ports. RGMP is disabled by default on Layer 3 interfaces.

RGMP Configuration Guidelines and Restrictions

When configuring RGMP, follow these guidelines and restrictions:

- Either RGMP or PIM snooping can be enabled in a VLAN but not both.
- RGMP supports PIM sparse mode. RGMP does not support PIM dense mode. RGMP explicitly supports the two AutoRP groups in dense mode by not restricting traffic to those groups but by flooding it to all router ports. For this reason, you should configure PIM sparse-dense mode. If you configure groups other than the AutoRP groups for dense mode, their traffic will not be correctly forwarded through router ports that have been enabled for RGMP.
- To effectively constrain multicast traffic with RGMP, connect RGMP-enabled routers to separate ports on RGMP-enabled Catalyst 6500 series switches. (VLAN interfaces satisfy this restriction.)
- RGMP only constrains traffic that exits through LAN ports on which it detects an RGMP-enabled router. If a non-RGMP enabled router is detected on a LAN port, that port receives all multicast traffic.
- RGMP does not support directly connected multicast sources in the network. A directly connected multicast source will send multicast traffic into the network without signaling through RGMP or PIM. This multicast traffic will not be received by an RGMP-enabled router unless the router already requested receipt of that multicast group through RGMP. This restriction applies to hosts and to functions in routers that source multicast traffic, such as the **ping** and **mtrace** commands and multicast applications that source multicast traffic, such as UDPTN.
- RGMP supports directly connected receivers in the network. Traffic to these receivers will be constrained by IGMP snooping, or if the receiver is a router itself, by PIM and RGMP.
- CGMP is not supported in networks where RGMP is enabled on routers. You cannot enable both RGMP and CGMP on a Layer 3 interface. If RGMP is enabled on a Layer 3 interface, CGMP is silently disabled and vice versa.

- The following properties of RGMP are the same as for IGMP snooping:
 - RGMP constrains traffic based on the multicast group, not on the sender's IP address.
 - If spanning tree topology changes occur in the network, the state is not flushed as it is with Cisco Group Management Protocol (CGMP).
 - RGMP does not constrain traffic for multicast groups 224.0.0.x (x = 0...255), which allows use of PIMv2 bootstrap router (BSR) in an RGMP-controlled network.
 - RGMP in Cisco network devices operates on MAC addresses, not on IP multicast addresses. Because multiple IP multicast addresses can map to one MAC address (see RFC 1112), RGMP cannot differentiate between the IP multicast groups that might map to a MAC address.
 - The capability of the Catalyst 6500 series switch to constrain traffic is limited by its content-addressable memory (CAM) table capacity.

Enabling RGMP on Layer 3 Interfaces

To enable RGMP on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port} {port-channel number}}	Selects an interface to configure.
Step 2	Router(config-if)# ip rgmp	Enables RGMP on the Layer 3 interface.
	Router(config-if)# no ip rgmp	Disables RGMP on the Layer 3 interface.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# debug ip rgmp [name_or_group_address]	(Optional) Monitors RGMP.

1. *type* = ethernet, fastethernet, gigabitethernet, tengigabitethernet, or ge-wan

This example shows how to configure RGMP on FastEthernet port 3/3:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 3/3
Router(config-if)# ip rgmp
Router(config-if)# end
Router#
```




Configuring Network Security

This chapter contains network security information unique to the Catalyst 6500 series switches, which supplements the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/index.htm
- *Cisco IOS Security Command Reference*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_r/index.htm



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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:

- [Configuring MAC Address-Based Traffic Blocking, page 34-1](#)
- [Configuring TCP Intercept, page 34-2](#)
- [Configuring Unicast Reverse Path Forwarding Check, page 34-2](#)

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 <i>mac_address</i> vlan <i>vlan_ID</i> drop	Blocks all traffic to or from the configured MAC address in the specified VLAN.
Router(config)# no mac-address-table static <i>mac_address</i> vlan <i>vlan_ID</i>	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/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftafwl/scfdenl.htm

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, page 34-2](#)
- [Understanding PFC2 Unicast RPF Check Support, page 34-3](#)
- [Unicast RPF Check Guidelines and Restrictions, page 34-3](#)
- [Configuring Unicast RPF Check, page 34-3](#)

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/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fothersf/scfrpf.htm

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.

Understanding PFC2 Unicast RPF Check Support

The PFC2 supports Unicast RPF check with hardware processing for packets that have a single return path. The MSFC2 processes traffic in software that has multiple return paths (for example, load sharing).

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 34-3](#)
- [Configuring the Multiple-Path Unicast RPF Check Mode on a PFC3, page 34-5](#)
- [Enabling Self-Pinging, page 34-6](#)

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 <i>vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>number</i> }}	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] [<i>list</i>] 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 switch.

This example shows how to enable Unicast RPF exist-only check mode on Gigabit Ethernet port 4/1:

```
Router(config)# interface gigabitethernet 4/1
Router(config-if)# ip verify unicast source reachable-via any
Router(config-if)# end
Router#
```

This example shows how to enable Unicast RPF strict check mode on Gigabit Ethernet port 4/2:

```
Router(config)# interface gigabitethernet 4/2
Router(config-if)# ip verify unicast source reachable-via rx
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show running-config interface gigabitethernet 4/2
Building configuration...
Current configuration : 114 bytes
!
interface GigabitEthernet4/2
ip address 42.0.0.1 255.0.0.0
```

```

ip verify unicast reverse-path
no cdp enable
end
Router# show running-config interface gigabitethernet 4/1
Building configuration...
Current configuration : 114 bytes
!
interface GigabitEthernet4/1
ip address 41.0.0.1 255.0.0.0
→ ip verify unicast reverse-path (RPF mode on g4/1 also changed to strict-check RPF mode)
no cdp enable
end
Router#

```

Configuring the Multiple-Path Unicast RPF Check Mode on a 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] <i>interface1</i> [<i>interface2</i> [<i>interface3</i> [<i>interface4</i>]]]	Configures a multiple path RPF interface group on a PFC3.
Step 2	Router(config)# mls ip cef rpf interface-group <i>group_number</i>	Removes an interface group.
Step 3	Router(config)# end	Exits configuration mode.
Step 4	Router# show mls cef ip rpf	Verifies the configuration.

This example shows how to configure interface group 2:

```
Router(config)# mls ip cef rpf interface-group 2 fastethernet 3/3 fastethernet 3/4
fastethernet 3/5 fastethernet 3/6
```

Enabling Self-Pinging

With Unicast RPF check enabled, by default the switch cannot ping itself.

To enable self-pinging, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{ <i>vlan vlan_ID</i> } { <i>type</i> ¹ <i>slot/port</i> } { <i>port-channel number</i> }}	Selects the interface to configure.
Step 2	Router(config-if)# ip verify unicast source reachable-via any allow-self-ping Router(config-if)# no ip verify unicast source reachable-via any allow-self-ping	Enables the switch to ping itself or a secondary address. 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
```



Understanding Cisco IOS ACL Support

This chapter describes Cisco IOS ACL support on the Catalyst 6500 series switches:

- [Cisco IOS ACL Configuration Guidelines and Restrictions, page 35-1](#)
- [Hardware and Software ACL Support, page 35-2](#)
- [Optimized ACL Logging on Supervisor Engine 720, page 35-3](#)
- [Guidelines and Restrictions for Using Layer 4 Operators in ACLs, page 35-5](#)

For complete information about configuring 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/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/trafwl/index.htm

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 36, “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 unreachable** 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 unreachable** 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:

- ACL flows that match a “deny” statement in standard and extended ACLs (input and output) are dropped in hardware if “ip unreachable” 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 the hardware; however, idle timeout is processed in software.
- 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 on Supervisor Engine 720”](#) section on page 35-3).
- 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 on Supervisor Engine 720

**Note**

Supervisor Engine 2 does not support optimized ACL logging (OAL).

Release 12.2(17d)SXB and later releases support OAL on the Supervisor Engine 720. These sections describe OAL:

- [Understanding OAL, page 35-3](#)
- [OAL Guidelines and Restrictions, page 35-3](#)
- [Configuring OAL, page 35-4](#)

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 MSFC3. 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 switch. With OAL configured, use SPAN to capture traffic.
- OAL is supported only on the Supervisor Engine 720.
- 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:

- [Configuring OAL Global Parameters, page 35-4](#)
- [Configuring OAL on an Interface, page 35-5](#)
- [Displaying OAL Information, page 35-5](#)
- [Clearing Cached OAL Entries, page 35-5](#)



Note

- For complete syntax and usage information for the commands used in this section, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX.
- 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 <i>number_of_entries</i> } { interval <i>seconds</i> } { rate-limit <i>number_of_packets</i> } { threshold <i>number_of_packets</i> }}	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 35-6](#)
- [Determining Logical Operation Unit Usage, page 35-6](#)

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 35-6](#) 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)



Configuring VLAN ACLs

This chapter describes how to configure VLAN ACLs (VACLs) on Catalyst 6500 series switches.



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- With a Supervisor Engine 720 and releases earlier than Release 12.2(17d)SXB, VACL capture is supported only for use with the WS-SVC-IDSM2-K9 Intrusion Detection System Module 2 and the WS-SVC-NAM-2 and WS-SVC-NAM-1 network analysis modules. This restriction is removed in Release 12.2(17d)SXB and later releases.
- OAL and VACL capture are incompatible. Do not configure both features on the switch. With OAL configured (see the “[Optimized ACL Logging on Supervisor Engine 720](#)” section on page 35-3), use SPAN to capture traffic.

This chapter consists of these sections:

- [Understanding VACLs, page 36-1](#)
- [Configuring VACLs, page 36-4](#)
- [Configuring VACL Logging, page 36-11](#)

Understanding VACLs

These sections describe VACLs:

- [VACL Overview, page 36-2](#)
- [Bridged Packets, page 36-2](#)
- [Routed Packets, page 36-3](#)
- [Multicast Packets, page 36-4](#)

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.

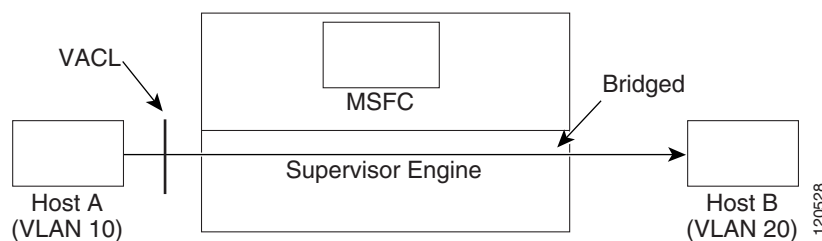
**Note**

- 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 36-1 shows a VACL applied on bridged packets.

Figure 36-1 Applying VACLs on Bridged Packets

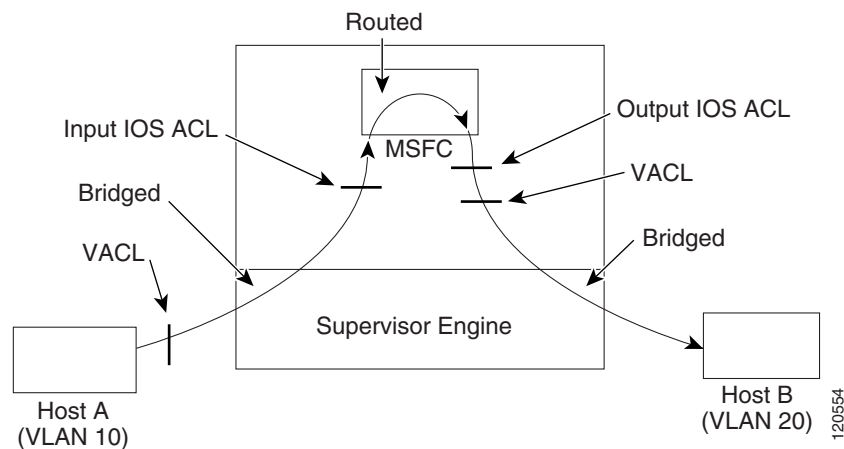


Routed Packets

Figure 36-2 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 36-2 Applying VACLs on Routed Packets

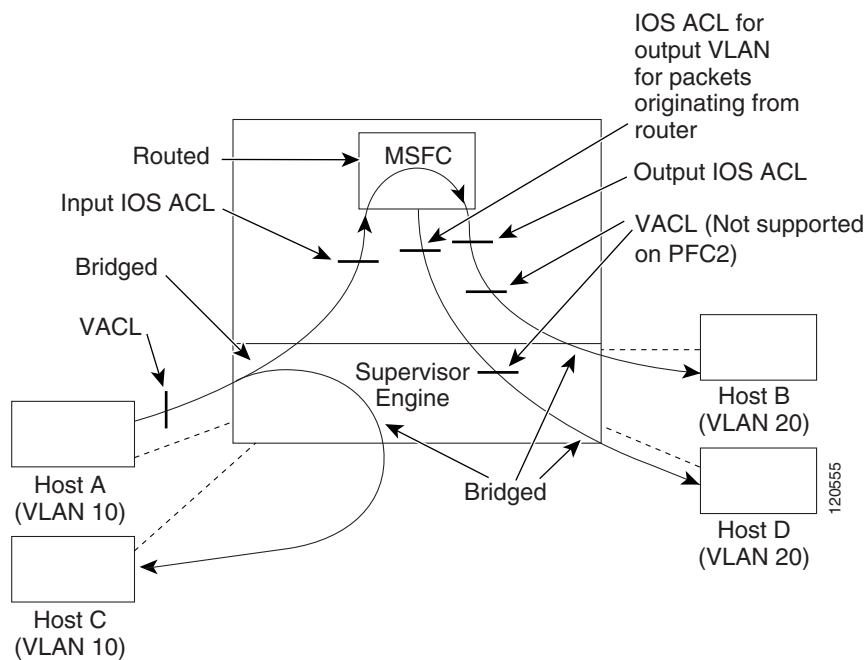


Multicast Packets

Figure 36-3 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 36-3 Applying VACLs on Multicast Packets



Configuring VACLs

These sections describe how to configure VACLs:

- [VACL Configuration Overview, page 36-5](#)
- [Defining a VLAN Access Map, page 36-5](#)
- [Configuring a Match Clause in a VLAN Access Map Sequence, page 36-6](#)
- [Configuring an Action Clause in a VLAN Access Map Sequence, page 36-7](#)
- [Applying a VLAN Access Map, page 36-8](#)

- [Verifying VLAN Access Map Configuration, page 36-8](#)
- [VLAN Access Map Configuration and Verification Examples, page 36-9](#)
- [Configuring a Capture Port, page 36-9](#)

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 42-50](#)) 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 <i>map_name</i> [0-65535]	Defines the VLAN access map. Optionally, you can specify the VLAN access map sequence number.
Router(config)# no vlan access-map <i>map_name</i> 0-65535	Deletes a map sequence from the VLAN access map.
Router(config)# no vlan access-map <i>map_name</i>	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 36-9.

Configuring a Match Clause in a VLAN Access Map Sequence

To configure a match clause in a VLAN access map sequence, perform this task:

Command	Purpose
Router(config-access-map)# match { ip address {1-199 1300-2699 <i>acl_name</i> } ipx address {800-999 <i>acl_name</i> } mac address <i>acl_name</i> }	Configures the match clause in a VLAN access map sequence.
Router(config-access-map)# no match { ip address {1-199 1300-2699 <i>acl_name</i> } ipx address {800-999 <i>acl_name</i> } mac address <i>acl_name</i> }	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 42-50.
- 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/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/index.htm

See the “VLAN Access Map Configuration and Verification Examples” section on page 36-9.

Configuring an Action Clause in a VLAN Access Map Sequence

To configure an action clause in a VLAN access map sequence, perform this task:

Command	Purpose
Router(config-access-map)# action { drop [log]} { forward [capture]} { redirect {{ ethernet fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> } { port-channel <i>channel_id</i> }}	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 } <i>slot/port</i> } { port-channel <i>channel_id</i> }}	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 36-9](#).
- 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 Supervisor Engine 720, if a VACL is redirecting traffic to an egress SPAN source port, SPAN does not copy the VACL-redirection traffic.
- With a Supervisor Engine 2, if a VACL is redirecting traffic to an egress SPAN source port, SPAN copies the VACL-redirection traffic.
- SPAN and RSPAN destination ports transmit VACL-redirection 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 36-9](#).

Applying a VLAN Access Map

To apply a VLAN access map, perform this task:

Command	Purpose
Router(config)# vlan filter <i>map_name</i> { vlan-list <i>vlan_list</i> interface <i>type</i> ¹ <i>number</i> ² }	Applies the VLAN access map to the specified VLANs or WAN interfaces.
Router(config)# no vlan filter <i>map_name</i> [vlan-list <i>vlan_list</i> interface <i>type</i> ¹ <i>number</i> 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. VACLs applied to VLANs without a Layer 3 VLAN interface are inactive. 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. If creation of the Layer 3 VLAN interface fails, the VACL is inactive.
- 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 36-9.

Verifying VLAN Access Map Configuration

To verify VLAN access map configuration, perform this task:

Command	Purpose
Router# show vlan access-map [<i>map_name</i>]	Verifies VLAN access map configuration by displaying the content of a VLAN access map.
Router# show vlan filter [access-map <i>map_name</i> vlan <i>vlan_id</i> interface <i>type</i> ¹ <i>number</i> 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



Note

A port configured to capture VACL-filtered traffic is called a capture port.

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 11-8 and the [“Configuring the Layer 2 Trunk Not to Use DTP”](#) section on page 11-9).

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} <i>vlan_list</i>	(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 11-7) 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 11-14) before you enter the **switchport capture** command.
- The capture port supports only egress traffic. No traffic can enter the switch through a capture port.

This example shows how to configure a Fast Ethernet interface 5/1 as a capture port:

```
Router(config)# interface gigabitEthernet 5/1
Router(config-if)# switchport capture
Router(config-if)# end
```

This example shows how to display VLAN access map information:

```
Router# show vlan access-map mordred
Vlan access-map "mordred" 10
    match: ip address net_10
    action: forward capture
Router#
```

This example shows how to display mappings between VACLs and VLANs. For each VACL map, there is information about the VLANs that the map is configured on and the VLANs that the map is active on. A VACL is not active if the VLAN does not have an interface.

```
Router# show vlan filter
VLAN Map mordred:
    Configured on VLANs: 2,4-6
    Active on VLANs: 2,4-6
Router#
```


Configuring VACL Logging

When you configure VACL logging, IP packets that are denied generate log messages in these situations:

- When the first matching packet is received
- For any matching packets received during the last 5-minute interval
- If the threshold is reached before the 5-minute interval

Log messages are generated on a per-flow basis. A flow is defined as packets with the same IP addresses and Layer 4 (UDP or TCP) port numbers. When a log message is generated, the timer and packet count is reset.

These restrictions apply to VACL logging:

- Because of the rate-limiting function for redirected packets, VACL logging counters may not be accurate.
- Only denied IP packets are logged.

To configure VACL logging, use the **action drop log** command action in VLAN access map submode (see the “[Configuring VACLs](#)” section on page 36-4 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 <i>max_number</i>	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 <i>pps</i>	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 <i>pkt_count</i>	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 { <i>{src_addr src_mask} any {host {hostname host_ip}}</i> } { <i>{dst_addr dst_mask} any {host {hostname host_ip}}</i> } [vlan <i>vlan_id</i>]	(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
```




Configuring Denial of Service Protection

This chapter contains information on how to protect your Catalyst 6500 series switch against Denial of Service (DoS) attacks. The information covered in this chapter is unique to the Catalyst 6500 series switches, and it supplements the network security information and procedures in the “[Configuring Network Security](#)” chapter in this publication as well as the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/index.htm
- *Cisco IOS Security Command Reference*, Release 12.2, at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_r/index.htm



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- 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 DoS Protection Works](#), page 37-2
- [DoS Protection Default Configuration](#), page 37-22
- [DoS Protection Configuration Guidelines and Restrictions](#), page 37-22
- [Understanding How Control Plane Policing Works](#), page 37-29
- [CoPP Default Configuration](#), page 37-29
- [CoPP Configuration Guidelines and Restrictions](#), page 37-30
- [Configuring CoPP](#), page 37-31
- [Monitoring CoPP](#), page 37-32
- [Defining Traffic Classification](#), page 37-33

Understanding How DoS Protection Works

The following sections contain an overview of the DoS protection on the Catalyst 6500 series switch and describe some types of DoS attack scenarios:

- [DoS Protection for the Supervisor Engine 2, page 37-2](#)
- [DoS Protection on the Supervisor Engine 720, page 37-10](#)

DoS Protection for the Supervisor Engine 2

This section contains information about the available methods to counteract DoS attacks on the Supervisor Engine 2 and includes configuration examples. The following sections describe these protection methods:



Note

The Supervisor Engine 2 CPU rate limiters are off by default.

- [Security ACLs, page 37-2](#)
- [Security ACLs, page 37-2](#)
- [QoS ACLs, page 37-3](#)
- [FIB Rate Limiting, page 37-4](#)
- [ARP Throttling, page 37-5](#)
- [uRPF Check, page 37-5](#)
- [TCP Intercept, page 37-6](#)

Security ACLs

The Catalyst 6500 series switch can deny DoS packets in hardware using security access control lists (ACLs). Security ACLs are applied in hardware using the TCAM to traffic that can be easily identified using Layer 3 or Layer 4 data. You can apply security ACLs preventively before a DoS attack occurs or after an attack has been identified.

This example shows how a security ACL is used to drop DoS packets:

```
Router# clear mls ip mod 9
Router# show mls ip mod 9
Displaying Netflow entries in module 9
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
Pkts           Bytes         Age   LastSeen  Attributes
-----
192.168.0.0    192.168.1.0  0    :0        :0        0    : 0
1843           84778         2     02:30:17  L3 - Dynamic
192.168.1.0    192.168.0.0  0     :0        :0        0    : 0
→ 2742416      126151136    2     02:30:17  L3 - Dynamic  <===== Note: traffic flow
identified
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# no access-list 199
Router(config)# access-list 199 deny ip host 192.168.0.0 any
Router(config)# access-list 199 permit ip any any
Router(config)# interface g9/1
```

```

→ Router(config-if)# ip access 199 in
   applied
Router(config-if)# end
Router#
1w6d: %SYS-5-CONFIG_I: Configured from console by console
Router# clear mls ip mod 9
Router# show mls ip mod 9
Displaying Netflow entries in module 9
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
Pkts           Bytes           Age    LastSeen  Attributes
-----
192.168.0.0    192.168.1.0 0    :0        :0        : 0
1542          70932         2    02:31:56  L3 - Dynamic
192.168.1.0    192.168.0.0 0    :0        :0        0 : 0
→ 0            0             2    02:31:56  L3 - Dynamic
→
Extended IP access list 199
  deny ip host 192.168.0.0 any (100 matches)
  permit ip any any
Router# show access-list 199
Extended IP access list 199
→   deny ip host 192.168.0.0 any (103 matches)
   permit ip any any
Router #

```

<===== Note: security ACL

<===== Note: hardware-forwarded

<===== Note: traffic stopped

Security VACLs

Security virtual access lists (VACLs) are security-enforcement tools based on Layer 2, Layer 3, and Layer 4 information. The result of a security VACL lookup against a packet can be a permit, a deny, a permit and capture, or a redirect. When you associate a security VACL with a particular VLAN, all traffic must be permitted by the security VACL before the traffic is allowed into the VLAN. Security VACLs are enforced in hardware, so there is no performance penalty for applying security VACLs to a VLAN on the Catalyst 6500 series switches.

QoS ACLs

Unlike security ACLs, QoS ACLs can be used to limit the rate of traffic without denying access to all the traffic in a flow.

This example shows how to use a QoS ACL to prevent a ping attack on a switch. A QoS ACL is configured and applied on all interfaces to limit the rate of incoming ICMP echo packets.

```

Router# show ip ospf neighbors

Neighbor ID      Pri   State           Dead Time   Address        Interface
6.6.6.122        1    FULL/BDR        00:00:30    6.6.6.122     Vlan46
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                Interface    Hold Uptime   SRTT   RTO   Q   Seq Type
                               (sec)          (ms)      Cnt Num
0   4.4.4.122                V144        11 00:06:07   4      200   0   6555
→ Router#
   starts
Router# show proc cpu | include CPU utilization
CPU utilization for five seconds: 99%/90%; one minute: 48%; five minutes: 25%
Router#
2w0d: %OSPF-5-ADJCHG: Process 100, Nbr 6.6.6.122 on Vlan46 from FULL to DOWN, Neighbor
Down: Dead timer expired
Router# show ip eigrp neighbors

```

<===== Note: ping attack

```

IP-EIGRP neighbors for process 200
Router#
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# access-list 199 permit icmp any any echo
Router(config)# class-map match-any icmp
Router(config-cmap)# match access-group 199
Router(config-cmap)# exit
Router(config)# policy-map icmp
Router(config-pmap)# class icmp
Router(config-pmap-c)# police 96000 16000 16000 conform-action transmit exceed-action drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface range g4/1 - 9
→ Router(config-if-range)# service-policy input icmp                                <===== Note: policy
applied
Router(config-if-range)# end
2w0d: %SYS-5-CONFIG_I: Configured from console by console
2w0d: %OSPF-5-ADJCHG: Process 100, Nbr 6.6.6.122 on Vlan46 from LOADING to FULL, Loading
Done
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                      Interface    Hold Uptime    SRTT    RTO  Q  Seq Type
                               (sec)        (ms)        Cnt  Num
0   4.4.4.122                    V144        13 00:00:48    8       200  0  6565
Router#

```

FIB Rate Limiting

The forwarding information base (FIB) rate-limiting feature allows all packets that require software processing to be rate limited.

This example shows traffic destined for a nonexistent host address on a locally connected subnet. Normally, the ARP request would result in an ARP reply and the installation of a FIB adjacency for this traffic. However, the adjacency in the FIB for the destination subnet would continue to receive traffic that would be forwarded for software processing. By applying rate-limiting to this traffic, the rate of traffic forwarded for software processing can be limited to a manageable amount.

```

Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                      Interface    Hold Uptime    SRTT    RTO  Q  Seq Type
                               (sec)        (ms)        Cnt  Num
0   4.4.4.122                    V144        11 00:00:26    8       200  0  6534
Router# show ip ospf neighbors

Neighbor ID      Pri   State           Dead Time   Address        Interface
6.6.6.122        1    FULL/BDR        00:00:36   6.6.6.122     Vlan46
→ Router#                                           <===== Note: attack
starts
Router# show arp | include 199.2.250.250
Internet 199.2.250.250      0    Incomplete     ARPA
Router#
1w6d: %OSPF-5-ADJCHG: Process 100, Nbr 6.6.6.122 on Vlan46 from FULL to DOWN, Neighbor
Down: Dead timer expired
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200
Router#
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
→ Router(config)# mls rate-limit unicast cef receive 1000                        <===== Note: traffic
rate limited to 1000 pps
Router(config)# end

```

```

Router#
1w6d: %SYS-5-CONFIG_I: Configured from console by console
Router#
1w6d: %OSPF-5-ADJCHG: Process 100, Nbr 6.6.6.122 on Vlan46 from LOADING to FULL, Loading Done
Router# show ip eigrp neighbors
IP-EIGRP neighbors for process 200

```

H	Address	Interface	Hold Uptime (sec)	SRTT (ms)	RTO	Q Cnt	Seq Num	Type
0	4.4.4.122	Vl44	12 00:00:07	12	200	0	6536	

```

Router#

```

Traffic Storm Control

A traffic storm occurs when packets flood the LAN, which creates excessive traffic and degrades network performance. The traffic storm control feature prevents LAN ports from being disrupted by a broadcast, multicast, or unicast traffic storm on physical interfaces from either mistakes in network configurations or from users issuing a DoS attack. Traffic storm control (also called traffic suppression) monitors incoming traffic levels over a 1-second traffic storm control interval. During the interval, traffic storm control compares the traffic level with the configured traffic storm control level. The traffic storm control level is a percentage of the total available bandwidth of the port. Each port has a single traffic storm control level that is used for all types of traffic (broadcast, multicast, and unicast).

Traffic storm control is configured on an interface and is disabled by default. The configuration example here enables broadcast address storm control on interface FastEthernet 2/3 to a level of 20 percent. When the broadcast traffic exceeds the configured level of 20 percent of the total available bandwidth of the port within a 1-second traffic-storm-control interval, traffic storm control will drop all broadcast traffic until the end of the traffic-storm-control interval.

```
Router(config-if)# storm-control broadcast level 20
```

The Catalyst 6500 series switch supports broadcast storm control on all LAN ports and multicast and unicast storm control on Gigabit Ethernet ports.

When two or three suppression modes are configured simultaneously, they share the same level settings. If broadcast suppression is enabled, and if multicast suppression is also enabled and configured at a 70-percent threshold, the broadcast suppression will also have a setting for 70 percent.

For more information about configuring traffic storm control see [Chapter 40, “Configuring Traffic Storm Control.”](#)

ARP Throttling

ARP throttling can be used to automatically install hardware-based FIB and adjacency entries to drop packets during ARP resolution. Most of these packets are dropped, but a small number are sent to the MSFC (rate limited).

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 in the Supervisor Engine 2 is supported in hardware but with only one return path. 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)

TCP Intercept

TCP intercept protects recipients of TCP traffic from TCP SYN-flooding DoS attacks. A normal TCP connection starts with a three-way handshake. Host A sends a SYN request to Host B requesting the start of a new TCP session. Host B responds with a SYN ACK acknowledging the receipt of the SYN request. Host A then returns an ACK for Host B's SYN ACK, and the session commences. A SYN-flooding attack occurs when hackers flood servers with requests for connections that have unreachable return addresses. The three-way handshake is never completed, and the connections cannot be established. The amount of session requests to which the server host is responding can overwhelm the server host and prevent legitimate users from connecting to legitimate services, such as web sites and email servers.

TCP intercept prevents the SYN flooding by intercepting and validating TCP requests. TCP intercept supports the following modes:

- **Intercept mode**—The TCP intercept software intercepts TCP synchronization (SYN) packets from clients to servers that match an extended access list. The software establishes a connection with the client on behalf of the destination server, and if successful, establishes the connection with the server on behalf of the client and connects the two half-connections together transparently. Connection attempts from unreachable hosts will never reach the server. The software continues to intercept and forward packets throughout the duration of the connection.

In the case of illegitimate requests from potential hackers, the software's aggressive timeouts on half-open connections and its thresholds on TCP connection requests protect destination servers while still allowing valid requests. When establishing the network security policy using TCP intercept, you can choose to intercept all requests or only those coming from specific networks or destined for specific servers. You also can configure the connection rate and threshold of outstanding connections.

- **Watch mode**—The software passively watches the connection requests flowing through the switch. If a connection fails to get established in a configurable interval, the software intervenes and terminates the connection attempt.

Because TCP intercept can operate in either active intercept mode or passive watch mode, it is important to decide which mode is suitable for the network, and to configure your network accordingly. TCP intercept is hardware-assisted on the PFC2 and PFC3 (all types). Configuring many sources and destinations for active intercept mode may overrun the CPU, so it is recommended that only critical servers be protected with active intercept mode.

The default mode of operation is intercept mode. In intercept mode, the software actively intercepts each incoming connection request (SYN) and responds on behalf of the server with the SYN-ACK, and then waits for the ACK from the client. After the preparation is complete, the original SYN is sent to the server, and the software performs the three-way handshake with the server. The two halves are connected together.

In Watch mode, the connection requests pass through the switch to the server, but are watched until they become established. If they fail to become established within 30 seconds (this value is configurable), the software sends a reset to the server to clear up its state. Configuring switches for watch mode has less CPU impact than intercept mode. In watch mode, the CPU is not performing checks and connects on both halves of the connection. The CPU is passively monitoring the connection and acting on failed connections after the fact.

TCP intercept is configured globally by first creating the extended access list for the traffic to be intercepted, and then creating the TCP intercept list. The type of traffic to be intercepted must be one of the following:

- All requests
- Only the requests that come from specific networks
- Only the requests that are destined for specific servers

This example defines the source in the access list as any; it does not attempt to filter the source address because it is difficult to know exactly who to intercept packets from. The destination, is specified to protect the destination servers from the TCP SYN-flood attack. If an access list match is not found, traffic is permitted to pass without further action.

```
Router(config)# access-list 101 permit tcp any 10.1.1.1 0.0.0.255
Router(config)# ip tcp intercept list 101
```

Table 37-1 lists the command used to configure the TCP intercept.

Table 37-1 TCP Intercept Configuration

Command	Purpose
Router(config)# access-list <i>access-list-number {deny permit} tcp any</i> <i>destination destination-wildcard</i>	Defines an IP extended access list.
Router(config)# ip tcp intercept list <i>access-list-number</i>	Enables TCP intercept.
Router(config)# ip tcp intercept mode { intercept watch }	Sets the TCP intercept mode.
Router(config)# ip tcp intercept drop-mode { oldest random }	Sets the drop mode.
Router(config)# ip tcp intercept watch-timeout <i>seconds</i>	Changes the time allowed to reach established state; valid values are from 1 to 2147483 seconds.
Router(config)# ip tcp intercept finrst-timeout <i>seconds</i>	Changes the time between receipt of a reset or FIN-exchange and dropping the connection; valid values are from 1 to 2147483 seconds.
Router(config)# ip tcp intercept connection-timeout <i>seconds</i>	Changes the time the software will manage a connection after no activity; valid values are from 1 to 2147483 seconds.

Table 37-1 TCP Intercept Configuration (continued)

Command	Purpose
Router(config)# ip tcp intercept max-incomplete low <i>number</i>	Defines the number of incomplete connections below which the software leaves aggressive mode; valid values are from 1 to 2147483647 connections.
Router(config)# ip tcp intercept max-incomplete high <i>number</i>	Defines the maximum number of incomplete connections allowed before the software enters aggressive mode; valid values are from 1 to 2147483647 connections.
Router(config)# ip tcp intercept one-minute low <i>number</i>	Defines the number of connection requests below which the software leaves aggressive mode; valid values are from 1 to 2147483647 connections.
Router(config)# ip tcp intercept one-minute high <i>number</i>	Defines the number of connection requests received in the last one-minutes sample period before the software enters aggressive mode; valid values are from 1 to 2147483647 connections.
Router# show tcp intercept connections	Displays incomplete connections and established connections.
Router# show tcp intercept statistics	Displays TCP intercept statistics.

Hardware-Based Rate Limiters on the Supervisor Engine 2

The Supervisor Engine 2 for the Catalyst 6500 series switches supports additional hardware-based rate limiters. The forwarding engine on the Supervisor Engine 2 provides four rate-limiter registers for the new rate limiters, which are configured globally on the switch. These rate-limiter registers are present in the Layer 3 forwarding engine (PFC) and are responsible for containing rate-limiting information for result packets that match the various available configured rate limiters.

Because four rate-limiter registers are present on the Layer 3 forwarding engine only, 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 Supervisor Engine 2 are as follows:

- Ingress and egress ACL bridged packets
- FIB receive and FIB glean cases
- VACL log
- Layer 3 features

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 switch accomplishes this by altering existing and new ACL TCAM entries with a TCAM bridge result to a Layer 3 redirect result pointing to the 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.

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# sh mls rate-limit
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0% Time source is NTP,
10:32:15.584 PDT Fri Aug 5 2005
```

Rate Limiter Type	Status	Packets/s
-----	-----	-----
ACL BRIDGE IN	Off	-
ACL BRIDGE OUT	Off	-
L3_SEC_FEATURES	Off	-
VACL LOG	Off	-
FIB RECEIVE	Off	-
FIB GLEAN	Off	-

FIB (CEF) Receive and FIB Glean 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
```

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
```

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 switch, IP packets that are denied in the VACL generate log messages.

This example shows how to rate limit logging requests to 5000 pps (the range for this rate limiter is from 10 to 5000 pps):

```
Router(config)# mls rate-limit unicast acl vACL-log 5000
```

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 switch to allow the users through after authentication. These ACLs are stored and processed in software, and if there are many users utilizing auth-proxy, the 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
```

DoS Protection on the Supervisor Engine 720

This section contains information about the available methods to counteract DoS attacks on the Supervisor Engine 720 and includes configuration examples. The Supervisor Engine 720 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 37-29](#).

These sections describe DoS protection on the Supervisor Engine 720:

- [Security ACLs and VACLs, page 37-11](#)
- [QoS Rate Limiting, page 37-12](#)
- [uRPF Check, page 37-12](#)
- [Traffic Storm Control, page 37-13](#)
- [Network Under SYN Attack, page 37-13](#)
- [ARP Policing, page 37-14](#)
- [Recommended Rate-Limiter Configuration, page 37-15](#)

- [Hardware-Based Rate Limiters on the Supervisor Engine 720, page 37-15](#)
 - [Ingress-Egress ACL Bridged Packets \(Unicast Only\), page 37-16](#)
 - [uRPF Check Failure, page 37-16](#)
 - [TTL Failure, page 37-17](#)
 - [ICMP Unreachable \(Unicast Only\), page 37-17](#)
 - [FIB \(CEF\) Receive Cases \(Unicast Only\), page 37-17](#)
 - [FIB Glean \(Unicast Only\), page 37-17](#)
 - [Layer 3 Security Features \(Unicast Only\), page 37-18](#)
 - [ICMP Redirect \(Unicast Only\), page 37-18](#)
 - [VACL Log \(Unicast Only\), page 37-18](#)
 - [MTU Failure, page 37-19](#)
 - [Layer 2 PDU, page 37-19](#)
 - [Layer 2 Protocol Tunneling, page 37-19](#)
 - [IP Errors, page 37-20](#)
 - [Layer 2 Multicast IGMP Snooping, page 37-19](#)
 - [IPv4 Multicast, page 37-20](#)
 - [IPv6 Multicast, page 37-21](#)

Security ACLs and VACLs

If the network is under a DoS attack, ACLs can be an efficient method for dropping the DoS packets before they reach the intended target. Use security ACLs if an attack is detected from a particular host. In this example, the host 10.1.1.10 and all traffic from that host is denied:

```
Router(config)# access-list 101 deny ip host 10.1.1.10 any
Router(config)# access-list 101 permit ip any any
```

Security ACLs also protect against the spoofing of addresses. For example, assume that a source address A is on the inside of a network and a switch interface that is pointing to the Internet. You can apply an inbound ACL on the switch Internet interface that denies all addresses with a source of A (the inside address). This action stops attacks where the attackers spoof inside source addresses. When the packet arrives at the switch interface, it matches on that ACL and drops the packet before it causes damage.

When the Catalyst 6500 series switch is used with a Cisco Intrusion Detection Module (CIDM), you can dynamically install the security ACL as a response to the detection of the attack by the sensing engine.

VACLs are a security enforcement tool based on Layer 2, Layer 3, and Layer 4 information. The result of a VACL lookup against a packet can be a permit, a deny, a permit and capture, or a redirect. When you associate a VACL with a particular VLAN, all traffic must be permitted by the VACL before the traffic is allowed into the VLAN. VACLs are enforced in hardware, so there is no performance penalty for applying VACLs to a VLAN on the Catalyst 6500 series switches.

QoS Rate Limiting

QoS ACLs limit the amount of a particular type of traffic that is processed by the 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 Supervisor Engine 720 performs QoS in hardware, which offers an efficient means of limiting DoS traffic (once that traffic has been identified) to protect the switch from impacting the 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 in the Supervisor Engine 720 is supported in hardware; it is also supported in hardware on the Supervisor Engine 2, but with only one return path. 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 Catalyst 6500 series switch supports broadcast storm control on all LAN ports and multicast and unicast storm control on Gigabit Ethernet ports.

When two or three suppression modes are configured simultaneously, they share the same level settings. If broadcast suppression is enabled, and if multicast suppression is also enabled and configured at a 70-percent threshold, the broadcast suppression will also have a setting for 70 percent.

Network Under SYN Attack

A network under a SYN attack is easily recognized. The target host becomes unusually slow, crashes, or suspends operation. Traffic returned from the target host can also cause trouble on the 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.

Another method for identifying a SYN attack is to examine the hits in the existing ACL. For example, suppose you have the following ACEs defined:

```
Router(config)# access-list 101 permit tcp any any established
Router(config)# access-list 101 permit tcp any any
```

The first entry matches any TCP packet with the ACK bit set, or any packet that is not a TCP SYN. The second ACE matches only TCP SYNs. By looking at the counters on these entries, in a normal operating environment the non-SYN TCP packets should outnumber the SYN TCP packets by at least a factor of three. If under a SYN attack, the SYN TCP packets will outnumber the non-SYN TCP packets by a factor of ten.

When the network is under a SYN attack, the TCP intercept feature becomes aggressively defensive. Two factors determine when aggressive behavior on the switch begins and ends:

- The total incomplete connections
- Connection requests during the last one-minute sample period

Both factors are configured with low and high values.

If the number of incomplete connections exceed 1,100, or the number of connections arriving in the last one-minute period exceed 1,100, each new arriving connection causes the oldest partial connection (or a random connection) to be deleted. These are the default values, which can be altered. When either of the thresholds is exceeded, the TCP intercept assumes the server is under attack and goes into aggressive mode with the following reactions:

- Each new arriving connection causes the oldest partial (or random partial) to be deleted.
- The initial retransmission timeout is reduced by half to 0.5 seconds, and so the total time trying to establish the connection is cut in half.
- In watch mode, the watch timeout is reduced by half.



Note When both thresholds fall below the configured low value, the aggressive behavior ceases (default value is 900 in both factors). See [Table 37-1](#) for information about TCP intercept configuration.

TCP flows are hardware assisted on both the PFC2 and PFC3 (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 configurable with the **mls qos protocol** command. The routing protocols supported include RIP, BGP, LDP, OSPF, IS-IS, IGRP, and EIGRP. For example, the command **mls qos protocol arp police 32000** rate limits ARP packets in hardware at 32,000 bps. Although this policing mechanism effectively protects the MSFC CPU against attacks such as line-rate ARP attacks, it does not only police routing protocols and ARP packets to the switch but also polices traffic through the box with less granularity than CoPP.

The policing mechanism shares the root configuration with a policing-avoidance mechanism. The policing-avoidance mechanism lets the routing protocol and ARP packets flow through the network when they reach a QoS policer. This mechanism can be configured using the **mls qos protocol protocol pass-through** command.

This example shows how to display the available protocols to use with ARP policing.

```
Router(config)# mls qos protocol ?
isis
eigrp
ldp
ospf
rip
bgp
ospfv3
bgpv2
ripng
neigh-discover
wlccp
arp
```

This example shows how to display the available keywords to use with the **mls qos protocol arp** command:

```
Router(config)# mls qos protocol arp ?
pass-through  pass-through keyword
police        police keyword
precedence    change ip-precedence(used to map the dscp to cos value)
```


Recommended Rate-Limiter Configuration

The recommended rate-limiter configuration is as follows:

- Enable the rate limiters for the traffic types most likely to be used in a DoS attack.
- Do not use a rate limiter on VACL logging unless you configure VACL logging.
- Disable redirects because a platform that supports hardware forwarding, such as the Catalyst 6500 series switch, reduces the need for redirects.
- Disable unreachable because a platform that supports hardware unreachables, such as the Catalyst 6500 series switch, reduces the need for unreachables.
- Do not enable the MTU rate limiter if all interfaces have the same MTU.
- When configuring the Layer 2 PDU rate limiter, note the following information:
 - Calculate the expected or possible number of valid PDUs and double or triple the number.
 - PDUs include BPDUs, DTP, VTP, PAgP, LACP, UDLD, etc.
 - Rate limiters do not discriminate between good frames or bad frames.

Hardware-Based Rate Limiters on the Supervisor Engine 720

The Supervisor Engine 720 for the Catalyst 6500 series switches supports additional hardware-based rate limiters. The forwarding engine on the Supervisor Engine 720 provides eight rate-limiter registers for the new rate limiters, which are configured globally on the switch. These rate-limiter registers are present in the Layer 3 forwarding engine (PFC) and are responsible for containing rate-limiting information for result packets that match the various available configured rate limiters.

Because eight rate-limiter registers are present on the Layer 3 forwarding engine only, 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 Supervisor Engine 720 are as follows:

- Ingress and egress ACL bridged packets
- uRPF check failures
- FIB receive cases
- FIB glean cases
- Layer 3 security features
- ICMP redirects
- ICMP unreachable (ACL drop)
- No-route (FIB miss)
- VACL log
- TTL failure
- MTU failure
- Multicast IPv4
- Multicast IPv6

Ingress-Egress ACL Bridged Packets (Unicast Only)

This rate limiter rate limits packets sent to the MSFC because of an ingress/egress ACL bridge result. The switch accomplishes this by altering existing and new ACL TCAM entries with a TCAM bridge result to a Layer 3 redirect result pointing to the 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 switch can accumulate up to 50 tokens and absorb a burst of 50 packets.

This example shows how to rate limit the unicast packets from an ingress ACL bridge result to 50000 packets per second, and 50 packets in burst:

```
Router(config)# mls rate-limit unicast acl input 50000 50
```

This example shows how to rate limit the unicast packets from an ingress ACL bridge result to the same rate (50000 pps and 50 packets in burst) for egress ACL bridge results:

```
Router(config)# mls rate-limit unicast acl output 50000 50
```

If the values of the rate limiter are altered on either the ingress or the egress when both are enabled, both values are changed to that new value. In the following example, the output rate is changed to 40000 pps:

```
Router(config)# mls rate-limit unicast acl output 40000 50
```

When you enter the **show mls rate-limit** command, both the ACL bridged in and the ACL bridged out display the new value of 40000 pps:

```
Router# show mls rate-limit
```

Rate Limiter Type	Status	Packets/s	Burst
MCAST NON RPF	Off	-	-
MCAST DFLT ADJ	On	100000	100
MCAST DIRECT CON	Off	-	-
ACL BRIDGED IN	On	40000	50
ACL BRIDGED OUT	On	40000	50
IP FEATURES	Off		

...

uRPF Check Failure

The uRPF check failure rate limiter allows you to configure a rate for the packets that need to be sent to the 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 switch to allow the users through after authentication. These ACLs are stored and processed in software, and if there are many users utilizing auth-proxy, the 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 switch, 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 switch, IP packets that are denied in the VACL generate log messages.

This example shows how to rate limit logging requests to 5000 pps (the range for this rate limiter is from 10 to 5000 pps):

```
Router(config)# mls rate-limit unicast acl vac1-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 Catalyst 6500 series switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.

This example shows how to rate limit IGMP-snooping traffic:

```
Router(config)# mls rate-limit multicast ipv4 igmp 20000 40
```

Layer 2 PDU

The Layer 2 protocol data unit (PDU) rate limiter allows you to limit the number of Layer 2 PDU protocol packets (including BPDUs, DTP, PAP, 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 Catalyst 6500 series switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.

This example shows how to rate limit Layer 2 PDUs to 20000 pps with a burst of 20 packets.

```
Router(config)# mls rate-limit layer2 pdu 20000 20
```

Layer 2 Protocol Tunneling

This rate limiter limits the Layer 2 protocol tunneling packets, which include control PDUs, CDP, STP, and VTP packets destined for the supervisor engine. These packets are encapsulated in software (rewriting the destination MAC address in the PDU), and then forwarded to a proprietary multicast address (01-00-0c-cd-cd-d0). You cannot enable the Layer 2 PDU rate limiter if the Catalyst 6500 series switch is operating in truncated mode. The switch uses truncated mode for traffic between fabric-enabled modules when there are both fabric-enabled and nonfabric-enabled modules installed. In this mode, the switch sends a truncated version of the traffic (the first 64 bytes of the frame) over the switch fabric channel.

This example shows how to rate limit Layer 2 protocol tunneling packets to 10000 pps with a burst of 10 packets:

```
Router(config)# mls rate-limit layer2 l2pt 10000 10
```

IP Errors

This rate limiter limits the packets with IP checksum and length errors. When a packet reaches the 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 in PFC3B or PFC3BXL mode 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 37-2](#) lists the IPv6 rate limiters and the class of traffic that each rate limiter serves.

Table 37-2 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-cntl rate limiter:

```
Router(config)# mls rate-limit multicast ipv6 route-cntl share auto
```

This example shows how to set the rate limiters for the IPv6 multicast packets from a directly connected source:

```
Router(config)# mls rate-limit multicast ipv6 connected 1500 20
```

This example shows how to configure a direct association of the rate limiters for a traffic class:

```
Router(config)# mls rate-limit multicast ipv6 default-drop 1000 20
```

This example shows how to configure the static sharing of a rate limiter with another preconfigured rate limiter:

```
Router(config)# mls rate-limit multicast ipv6 route-cntl share default-drop
```

This example shows how to enable dynamic sharing for the route control rate limiter:

```
Router(config)# mls rate-limit multicast ipv6 route-cntl share auto
```

DoS Protection Default Configuration

Table 37-3 shows the DoS protection default configuration for the Supervisor Engine 720 hardware-based rate limiters.

Table 37-3 Supervisor Engine 720 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 <i>packets-in-burst</i> 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:

- [Supervisor Engine 2, page 37-23](#)

- [Supervisor Engine 720, page 37-23](#)

Supervisor Engine 2

When configuring DoS protection on systems configured with a Supervisor Engine 2, follow these guidelines and restrictions:

- When using security ACLs to drop DoS packets, note the following information:
 - The security ACL must specify the traffic flow to be dropped.
 - Security ACLs need to be configured on all external interfaces that require protection. Use the **interface range** command to configure a security ACL on multiple interfaces.
- When using QoS ACLs to limit the rate of packets, note the following information:
 - The QoS ACL must specify the traffic flow to be rate limited.
 - When adding a QoS ACL to limit the rate of packets to an interface that already has a QoS ACL configured, you can perform one of the following:
 - * Merge the rate-limiting ACL with the existing QoS ACL.
 - * Define a separate class that matches the DoS ACL and tie the class to the policy map.
 - QoS ACLs need to be configured on all external interfaces that require protection. Use the **interface range** command to configure an ACL on multiple interfaces.
- The CPU rate limiters limit the traffic in aggregate only and do not distinguish between good and bad packets.
- The following FIB rate-limiting usage guidelines apply:
 - FIB rate limiting does not limit the rate of broadcast or some multicast traffic in hardware.
The Supervisor Engine 720 has separate multicast rate limiters. The Supervisor Engine 2 does not have separate multicast rate limiters.
 - FIB rate limiting does not differentiate between legitimate and illegitimate traffic (for example, tunnels, Telnet).
 - FIB rate limiting applies aggregate rate limiting and not per-flow rate limiting.

Supervisor Engine 720

When configuring DoS protection on systems configured with a Supervisor Engine 720, 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 37-30](#).

- 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 in PFC3B or PFC3BXL mode:
 - Unicast IP options
 - Multicast IP options
- These are Layer 2 rate limiters:
 - Layer 2 PDUs
 - Layer 2 protocol tunneling
 - Layer 2 Multicast IGMP
- There are eight Layer 3 registers and two Layer 2 registers that can be used as CPU rate limiters.
- Do not use the CEF receive limiter if CoPP is being used. The CEF receive limiter will override the CoPP traffic.
- Rate limiters override the CoPP traffic.
- Configured rate limits is applied to each forwarding engine (except for the Layer 2 hardware rate limiter which is applied globally).
- Layer 2 rate limiters are not supported in truncated mode.
- The following restrictions apply when using the ingress and egress ACL-bridged packet rate limiters:
 - The ingress and egress ACL-bridged packet rate limiter is available for unicast traffic only.
 - The ingress and egress ACL-bridged packet rate limiters share a single rate-limiter register. If you enable the ACL-bridge ingress and egress rate limiters, both the ingress and the egress ACLs must share the same rate-limiter value.
- Use the **mls rate-limit unicast** command to rate limit unicast traffic.
- Use the **mls rate-limit multicast** command to rate limit multicast traffic.
- Use the **mls rate-limit multicast layer 2** command to rate limit Layer 2 multicast traffic.

Monitoring Packet Drop Statistics

Because the rate-limiting method allows a certain number of packets to be forwarded for software processing, you can view the packet drop statistics by entering NetFlow **show** commands from the CLI. You can also capture the incoming or outgoing traffic on an interface and send a copy of this traffic to an external interface for monitoring by a traffic analyzer. To capture traffic and forward it to an external interface, use the **monitor session** command.

When capturing traffic, these restrictions apply:

- The incoming captured traffic is not filtered.
- The incoming captured traffic is not rate limited to the capture destination.

Monitoring Dropped Packets Using NetFlow Commands

The following NetFlow commands display flows that are destined to the switch MAC that are either hardware switched or forwarded to the MSFC.

Statistics that are based on source or flow can be displayed only if the MLS NetFlow flow mask is set to a value that is greater than destination-only.

```

Router# show mls ip
Displaying Netflow entries in Supervisor Earl
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
200.2.5.3      0.0.0.0          0 :0 :0              0 : 0

Pkts          Bytes          Age   LastSeen  Attributes
-----
0             0              1     01:52:25  L3 - Dynamic

Router# show mls netflow flowmask
current ip flowmask for unicast: destination only
current ipx flowmask for unicast: destination only

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mls flow ip destination-source
Router(config)# exit
1w6d: %SYS-5-CONFIG_I: Configured from console by console
Router# show mls ip
Displaying Netflow entries in Supervisor Earl
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
200.2.5.3      223.255.254.226 0 :0 :0              0 : 0

Pkts          Bytes          Age   LastSeen  Attributes
-----
0             0              2     01:54:05  L3 - Dynamic

Router#

```

When you use the **show mls ip** command to display information about flows for a specific source or destination address, the command accepts 32-bit prefixes only. When you use the output modifiers, you might see all flows from a specific subnet.

```

Router# show mls ip source 9.9.9.2 mod 4
Displaying Netflow entries in module 4
-----
DstIP          SrcIP          Prot:SrcPort:DstPort  Src i/f:AdjPtr
-----
9.9.9.177      9.9.9.2        0 :0 :0              0 : 0

Pkts          Bytes          Age   LastSeen  Attributes
-----
0             0              28    01:56:59  L3 - Dynamic

Router# show mls ip mod 4 | include 9.9.9
9.9.9.177      9.9.9.2        0 :0 :0              0 : 0
9.9.9.177      9.9.9.1        0 :0 :0              0 : 0

```

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

PFC3B and PFC3BXL mode supports ACL hit counters in hardware. You can use the **show tcam interface** command to display each entry in the ACL TCAM.

This example shows how to use the **show tcam interface** command to display the number of times the entry was hit:

```
Router# show tcam interface fa5/2 acl in ip detail
```

```
-----
DPort - Destination Port   SPort - Source Port       TCP-F - U -URG Pro   - Protocol
I      - Inverted LOU      TOS   - TOS Value         - A -ACK rtr   - Router
MRFM   - M -MPLS Packet   TN     - T -Tcp Control       - P -PSH COD   - C -Bank Care Flag
        - R -Recirc. Flag   - N -Non-cachable     - R -RST       - I -OrdIndep. Flag
        - F -Fragment Flag CAP - Capture Flag         - S -SYN       - D -Dynamic Flag
        - M -More Fragments F-P - FlowMask-Prior.     - F -FIN T     - V(Value)/M(Mask)/R(Result)
X      - XTAG              (*) - Bank Priority
-----
```

```
Interface: 1018   label: 1   lookup_type: 0
protocol: IP   packet-type: 0
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|T|Index|  Dest Ip Addr | Source Ip Addr|   DPort   |   SPort   | TCP-F|Pro|MRFM|X|TOS|TN|COD|F-P|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
V 18396          0.0.0.0      0.0.0.0      P=0         P=0         ----- 0 --- 0 0 -- --- 0-0
M 18404          0.0.0.0      0.0.0.0      0           0           0 --- 0 0
R rslt: L3_DENY_RESULT          rtr_rslt: L3_DENY_RESULT

V 36828          0.0.0.0      0.0.0.0      P=0         P=0         ----- 0 --- 0 0 -- --- 0-0
M 36836          0.0.0.0      0.0.0.0      0           0           0 --- 0 0
R rslt: L3_DENY_RESULT (*)      rtr_rslt: L3_DENY_RESULT (*)
Router#
```

You can also use the TTL and IP options counters to monitor the performance of the Layer 3 forwarding engine.

This example shows how to use the **show mls statistics** command to display packet statistics and errors associated with the Layer 3 forwarding engine:

```
Router# show mls statistics

Statistics for Earl in Module 6

L2 Forwarding Engine
  Total packets Switched           : 25583421

L3 Forwarding Engine
  Total packets L3 Switched        : 25433414 @ 24 pps

  Total Packets Bridged             : 937860
  Total Packets FIB Switched        : 23287640
  Total Packets ACL Routed          : 0
  Total Packets Netflow Switched    : 0
  Total Mcast Packets Switched/Routed : 96727
  Total ip packets with TOS changed : 2
  Total ip packets with COS changed : 2
  Total non ip packets COS changed  : 0
  Total packets dropped by ACL       : 33
  Total packets dropped by Policing  : 0

Errors
  MAC/IP length inconsistencies     : 0
  Short IP packets received         : 0
  IP header checksum errors         : 0
  TTL failures                      : 0
<----- TTL counters
  MTU failures                      : 0
<-----MTU failure counters

Total packets L3 Switched by all Modules: 25433414 @ 24 pps
```

Monitoring Dropped Packets Using VACL Capture

The VACL capture feature allows you to direct traffic to ports configured to forward captured traffic. The capture action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets. Only forwarded packets can be captured.

You can use VACL capture to assign traffic from each VLAN to a different interface.

VACL capture does not allow you to send one type of traffic, such as HTTP, to one interface and another type of traffic, such as DNS, to another interface. Also, VACL capture granularity is only applicable to traffic switched locally; you cannot preserve the granularity if you direct traffic to a remote switch.

This example shows how to use VACL capture to capture and forward traffic to a local interface:

```
Router(config-if)# switchport capture
Router(config-if)# switchport capture allowed vlan add 100
```

Displaying Rate-Limiter Information

The **show mls rate-limit** command displays information about the configured rate limiters.

The **show mls rate-limit usage** command displays the hardware register that is used by a rate-limiter type. If the register is not used by any rate-limiter type, Free is displayed in the output. If the register is used by a rate-limiter type, Used and the rate-limiter type are displayed.

In the command output, the rate-limit status could be one of the following:

- On indicates that a rate for that particular case has been set.
- Off indicates that the rate-limiter type has not been configured, and the packets for that case are not rate limited.
- On/Sharing indicates that a particular case (not manually configured) is affected by the configuration of another rate limiter belonging to the same sharing group.
- A hyphen indicates that the multicast partial-SC rate limiter is disabled.

In the command output, the rate-limit sharing indicates the following information:

- Whether sharing is static or dynamic
- Group dynamic sharing codes

To display the configured rate limiters, use the **show mls rate-limit** command:

```
Router# show mls rate-limit
```

Sharing Codes: S - static, D - dynamic

Codes dynamic sharing: H - owner (head) of the group, g - guest of the group

Rate Limiter Type	Status	Packets/s	Burst	Sharing
-----	-----	-----	-----	-----
MCAST NON RPF	Off	-	-	-
MCAST DFLT ADJ	On	100000	100	Not sharing
MCAST DIRECT CON	Off	-	-	-
ACL BRIDGED IN	Off	-	-	-
ACL BRIDGED OUT	Off	-	-	-
IP FEATURES	Off	-	-	-
ACL VACL LOG	On	2000	1	Not sharing
CEF RECEIVE	Off	-	-	-
CEF GLEAN	Off	-	-	-
MCAST PARTIAL SC	On	100000	100	Not sharing
IP RPF FAILURE	On	100	10	Group:0 S
TTL FAILURE	Off	-	-	-
ICMP UNREAC. NO-ROUTE	On	100	10	Group:0 S
ICMP UNREAC. ACL-DROP	On	100	10	Group:0 S
ICMP REDIRECT	Off	-	-	-
MTU FAILURE	Off	-	-	-
MCAST IP OPTION	Off	-	-	-
UCAST IP OPTION	Off	-	-	-
LAYER_2 PDU	Off	-	-	-
LAYER_2 PT	Off	-	-	-
IP ERRORS	On	100	10	Group:0 S
CAPTURE PKT	Off	-	-	-
MCAST IGMP	Off	-	-	-
MCAST IPv6 DIRECT CON	Off	-	-	-
MCAST IPv6 *G M BRIDG	Off	-	-	-
MCAST IPv6 *G BRIDGE	Off	-	-	-
MCAST IPv6 SG BRIDGE	Off	-	-	-
MCAST IPv6 ROUTE CNTL	Off	-	-	-
MCAST IPv6 DFLT DROP	Off	-	-	-
MCAST IPv6 SECOND. DR	Off	-	-	-

Router#

To display the usage of the hardware rate limiters, use the **show mls rate-limit usage** command:

```
Router# show mls rate-limit usage
```

	Rate Limiter Type	Packets/s	Burst
	-----	-----	-----
Layer3 Rate Limiters:			
RL# 0:	Free	-	-
RL# 1:	Free	-	-

```

RL# 2: Free          -          -          -
RL# 3: Used
                    MCAST DFLT ADJ      100000      100
RL# 4: Free          -          -          -
RL# 5: Free          -          -          -
RL# 6: Used
                    IP RPF FAILURE        100        10
                    ICMP UNREAC. NO-ROUTE 100        10
                    ICMP UNREAC. ACL-DROP 100        10
                    IP ERRORS             100        10
RL# 7: Used
                    ACL VACL LOG          2000        1
RL# 8: Rsvd for capture -          -          -

Layer2 Rate Limiters:
RL# 9: Reserved
RL#10: Reserved
RL#11: Free          -          -          -
RL#12: Free          -          -          -

Router#

```

Understanding How Control Plane Policing Works

The control plane policing (CoPP) feature increases security on the Catalyst 6500 series switch by protecting the 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 Supervisor Engine 720 rate limiters.



Note

The Supervisor Engine 2 does not support CoPP.

The Supervisor Engine 720 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.

CoPP Configuration Guidelines and Restrictions

When configuring CoPP, follow these guidelines and restrictions:

- In releases earlier than Release 12.2(18)SXE, the Supervisor Engine 720 does not support the MQC class-default in hardware; the class default is replaced with a normal class map. If you define a catch-all map, the MQC class-default is supported in hardware.
- 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 switches. Filtering this traffic could prevent remote access to the switch, requiring a console connection.
- Supervisor Engine 720 supports built-in special-case rate limiters, which are useful for situations where an ACL cannot be used (for example, TTL, MTU, and IP options). When you enable the special-case rate limiters, you should be aware that the special-case rate limiters will override the CoPP policy for packets matching the rate-limiter criteria.
- CoPP is not enabled in hardware unless MMLS QoS is enabled globally with the **mls qos** command. If the **mls qos** command is not entered, CoPP will only work in software and will not provide any benefit to the hardware.
- Neither egress CoPP nor silent mode is supported. CoPP is only supported on ingress (service-policy output CoPP cannot be applied to the control plane interface).
- ACE hit counters in hardware are only for ACL logic. You can rely on software ACE hit counters and the **show access-list**, **show policy-map control-plane**, and **show mls ip qos** commands to troubleshoot evaluate CPU traffic.
- CoPP is performed on a per-forwarding-engine basis and software CoPP is performed on an aggregate basis.
- CoPP is not supported in hardware for multicast packets. The combination of ACLs, multicast CPU rate limiters and CoPP software protection provides protection against multicast DoS attacks.
- CoPP does not support ACEs with the **log** keyword.
- CoPP uses hardware QoS TCAM resources. Enter the **show tcam utilization** command to verify the TCAM utilization.

Configuring CoPP

CoPP uses MQC to define traffic classification criteria and to specify the configurable policy actions for the classified traffic. You must first identify the traffic to be classified by defining a class map. The class map defines packets for a particular traffic class. After you have classified the traffic, you can create policy maps to enforce policy actions for the identified traffic. The **control-plane** global configuration command allows the CoPP service policies to be directly attached to the control plane.

For information on how to define the traffic classification criteria, refer to the [“Defining Traffic Classification” section on page 37-33](#).

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 <i>access-list-name</i> Router(config-ext-nacl)# { permit deny } <i>protocol source source-wildcard</i> destination destination-wildcard [precedence precedence] [tos tos] [established] [log log-input] [time-range <i>time-range-name</i>] [fragments]	Defines ACLs to match traffic: <ul style="list-style-type: none"> 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 <i>traffic-class-name</i> Router(config-cmap)# match { ip precedence } { ip dscp } <i>access-group</i>	Defines the packet classification criteria. Use the match statements to identify the traffic associated with the class.
Step 4	Router(config)# policy-map <i>service-policy-name</i> Router(config-pmap)# class <i>traffic-class-name</i> Router(config-pmap-c)# police { <i>bits-per-second</i> [<i>normal-burst-bytes</i>] [<i>maximum-burst-bytes</i>] [pir peak-rate-bps]} [conform-action action] [exceed-action <i>action</i>] [violate-action action]	Defines a service policy map. Use the class <i>traffic-class-name</i> 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 <i>service-policy-name</i>	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.
- In releases earlier than Release 12.2(18)SXE, the MQC class default is not supported.

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      Id
-----
CPP  5  In CoPP-normal  0    1  dscp  0      505408      83822272
CPP  9  In CoPP-normal  0    4  dscp  0          0          0
Router#
```

To display the CoPP access list information, enter the `show access-lists coppacl-bgp` command:

```
Router# show access-lists coppacl-bgp
Extended IP access list coppacl-bgp
10 permit tcp host 47.1.1.1 host 10.9.9.9 eq bgp (4 matches)
20 permit tcp host 47.1.1.1 eq bgp host 10.9.9.9
30 permit tcp host 10.86.183.120 host 10.9.9.9 eq bgp (1 match)
40 permit tcp host 10.86.183.120 eq bgp host 10.9.9.9
```

Defining Traffic Classification

The following sections contain information on how to classify CoPP traffic:

- [Traffic Classification Overview, page 37-33](#)
- [Traffic Classification Guidelines, page 37-34](#)
- [Sample Basic ACLs for CoPP Traffic Classification, page 37-34](#)

Traffic Classification Overview

You can define any number of classes, but typically traffic is grouped into classes that are based on relative importance. The following provides a sample grouping:

- **Border Gateway Protocol (BGP)**—Traffic that is crucial to maintaining neighbor relationships for BGP routing protocol, for example, BGP keepalives and routing updates. Maintaining BGP routing protocol is crucial to maintaining connectivity within a network or to a service provider. Sites that do not run BGP do not need to use this class.
- **Interior Gateway Protocol (IGP)**—Traffic that is crucial to maintaining IGP routing protocols, for example, open shortest path first OSPF, enhanced interior gateway routing protocol (EIGRP), and routing information protocol (RIP). Maintaining IGP routing protocols is crucial to maintaining connectivity within a network.
- **Management**—Necessary, frequently used traffic that is required during day-to-day operations. For example, traffic used for remote network access, and Cisco IOS image upgrades and management, such as telnet, secure shell (SSH), network time protocol (NTP), simple network management protocol (SNMP), terminal access controller access control system (TACACS), hypertext transfer protocol (HTTP), trivial file transfer protocol (TFTP), and file transfer protocol (FTP).
- **Reporting**—Traffic used for generating network performance statistics for the purpose of reporting. For example, using Cisco IOS IP service level agreements (SLAs) to generate ICMP with different DSCP settings in order to report on response times within different QoS data classes.
- **Monitoring**—Traffic used for monitoring a switch. Traffic should be permitted but should never pose a risk to the switch; with CoPP, this traffic can be permitted but limited to a low rate. For example, ICMP echo request (ping) and traceroute.

- **Critical Applications**—Critical application traffic that is specific and crucial to a particular customer environment. Traffic included in this class should be tailored specifically to the required application requirements of the user (in other words, one customer may use multicast, while another uses IPsec or generic routing encapsulation (GRE). For example, GRE, hot standby router protocol (HSRP), virtual router redundancy protocol (VRRP), session initiation protocol (SIP), data link switching (DLSw), dynamic host configuration protocol (DHCP), multicast source discovery protocol (MSDP), Internet group management protocol (IGMP), protocol independent multicast (PIM), multicast traffic, and IPsec.
- **Layer 2 Protocols**—Traffic used for address resolution protocol (ARP). Excessive ARP packets can potentially monopolize MSFC resources, starving other important processes; CoPP can be used to rate limit ARP packets to prevent this situation. Currently, ARP is the only Layer 2 protocol that can be specifically classified using the match protocol classification criteria.
- **Undesirable**—Explicitly identifies bad or malicious traffic that should be unconditionally dropped and denied access to the MSFC. The undesirable classification is particularly useful when known traffic destined for the switch should always be denied and not placed into a default category. If you explicitly deny traffic, then you can enter **show** commands to collect approximate statistics on the denied traffic and estimate its rate.
- **Default**—All remaining traffic destined for the MSFC that has not been identified. MQC provides the default class, so the user can specify the treatment to be applied to traffic not explicitly identified in the other user-defined classes. This traffic has a highly reduced rate of access to the MSFC. With a default classification in place, statistics can be monitored to determine the rate of otherwise unidentified traffic destined for the control plane. After this traffic is identified, further analysis can be performed to classify it and, if needed, the other CoPP policy entries can be updated to accommodate this traffic.

After you have classified the traffic, the ACLs build the classes of traffic that are used to define the policies. For sample basic ACLs for CoPP classification, see the [“Sample Basic ACLs for CoPP Traffic Classification”](#) section on page 37-34.

Traffic Classification Guidelines

When defining traffic classification, follow these guidelines and restrictions:

- Before you develop the actual CoPP policy, you must identify and separate the required traffic into different classes. Traffic is grouped into nine classes that are based on relative importance. The actual number of classes needed might differ and should be selected based on your local requirements and security policies.
- You do not have to define policies that match bidirectionally. You only need to identify traffic unidirectionally (from the network to the MSFC) since the policy is applied on ingress only.

Sample Basic ACLs for CoPP Traffic Classification

This section shows sample basic ACLs for CoPP classification. In the samples, the commonly required traffic is identified with these ACLs:

- ACL 120—Critical traffic
- ACL 121—Important traffic
- ACL 122—Normal traffic
- ACL 123—Explicitly denies unwanted traffic

- ACL 124—All other traffic

This example shows how to define ACL 120 for critical traffic:

```
Router(config)# access-list 120 remark CoPP ACL for critical traffic
```

This example shows how to allow BGP from a known peer to this switch's BGP TCP port:

```
Router(config)# access-list 120 permit tcp host 47.1.1.1 host 10.9.9.9 eq bgp
```

This example shows how to allow BGP from a peer's BGP port to this switch:

```
Router(config)# access-list 120 permit tcp host 47.1.1.1 eq bgp host 10.9.9.9
Router(config)# access-list 120 permit tcp host 10.86.183.120 host 10.9.9.9 eq bgp
Router(config)# access-list 120 permit tcp host 10.86.183.120 eq bgp host 10.9.9.9
```

This example shows how to define ACL 121 for the important class:

```
Router(config)# access-list 121 remark CoPP Important traffic
```

This example shows how to permit return traffic from TACACS host:

```
Router(config)# access-list 121 permit tcp host 1.1.1.1 host 10.9.9.9 established
```

This example shows how to permit SSH access to the switch from a subnet:

```
Router(config)# access-list 121 permit tcp 10.0.0.0 0.0.0.255 host 10.9.9.9 eq 22
```

This example shows how to allow full access for Telnet to the switch from a host in a specific subnet and police the rest of the subnet:

```
Router(config)# access-list 121 deny tcp host 10.86.183.3 any eq telnet
Router(config)# access-list 121 permit tcp 10.86.183.0 0.0.0.255 any eq telnet
```

This example shows how to allow SNMP access from the NMS host to the switch:

```
Router(config)# access-list 121 permit udp host 1.1.1.2 host 10.9.9.9 eq snmp
```

This example shows how to allow the switch to receive NTP packets from a known clock source:

```
Router(config)# access-list 121 permit udp host 1.1.1.3 host 10.9.9.9 eq ntp
```

This example shows how to define ACL 122 for the normal traffic class:

```
Router(config)# access-list 122 remark CoPP normal traffic
```

This example shows how to permit switch-originated traceroute traffic:

```
Router(config)# access-list 122 permit icmp any any ttl-exceeded
Router(config)# access-list 122 permit icmp any any port-unreachable
```

This example shows how to permit receipt of responses to the switch that originated the pings:

```
Router(config)# access-list 122 permit icmp any any echo-reply
```

This example shows how to allow pings to the switch:

```
Router(config)# access-list 122 permit icmp any any echo
```

This example shows how to define ACL 123 for the undesirable class.

```
Router(config)# access-list 123 remark explicitly defined "undesirable" traffic
```

**Note**

In the following example, ACL 123 is a permit entry for classification and monitoring purposes, and traffic is dropped as a result of the CoPP policy.

This example shows how to permit all traffic destined to UDP 1434 for policing:

```
Router(config)# access-list 123 permit udp any any eq 1434
```

This example shows how to define ACL 124 for all other traffic:

```
Router(config)# access-list 124 remark rest of the IP traffic for CoPP
Router(config)# access-list 124 permit ip any any
```

Configuring Sticky ARP

Sticky ARP prevents MAC address spoofing by ensuring that ARP entries (IP address, MAC address, and source VLAN) do not get overridden. The switch maintains ARP entries in order to forward traffic to end devices or other switches. ARP entries are usually updated periodically or modified when ARP broadcasts are received. During an attack, ARP broadcasts are sent using a spoofed MAC address (with a legitimate IP address) so that the switch learns the legitimate IP address with the spoofed MAC address and begins to forward traffic to that MAC address. With sticky ARP enabled, the switch learns the ARP entries and does not accept modifications received through ARP broadcasts. If you attempt to override the sticky ARP configuration, you will receive an error message. For a complete description of the system error messages, refer to the *Catalyst 6500 Series Switch Cisco IOS System Message Guide*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/msgguide/index.htm>



Note

Release 12.2(18)SXF and later releases support sticky ARP configurability.

To configure sticky ARP on a Layer 3 interface, perform the following task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	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#
```



Configuring DHCP Snooping

This chapter describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping on Catalyst 6500 series switches. Release 12.2(18)SXE and later releases support DHCP snooping.

This chapter consists of the following major sections:

- [Overview of DHCP Snooping, page 38-1](#)
- [Default Configuration for DHCP Snooping, page 38-5](#)
- [Configuring DHCP Snooping, page 38-6](#)



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

Overview of DHCP Snooping

DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding database (also referred to as a DHCP snooping binding table).

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. You can use DHCP snooping to differentiate between untrusted interfaces connected to the end user and trusted interfaces connected to the DHCP server or another switch.



Note

For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces.

An untrusted message is a message that is received from outside the network or firewall. When you use DHCP snooping in a service-provider environment, an untrusted message is sent from a device that is not in the service-provider network, such as a customer's switch. Messages from unknown devices are untrusted because they can be sources of traffic attacks.

The DHCP snooping binding database contains the MAC address, the IP address, the lease time, the binding type, the VLAN number, and the interface information that corresponds to the local untrusted interfaces of a switch. The database does not contain information regarding hosts interconnected with a trusted interface.

In a service-provider network, a trusted interface is connected to a port on a device in the same network. An untrusted interface is connected to an untrusted interface in the network or to an interface on a device that is not in the network.

When a switch receives a packet on an untrusted interface and the interface belongs to a VLAN in which DHCP snooping is enabled, the switch compares the source MAC address and the DHCP client hardware address. If addresses match (the default), the switch forwards the packet. If the addresses do not match, the switch drops the packet.

The switch drops DHCP packets when any of these situations occur:

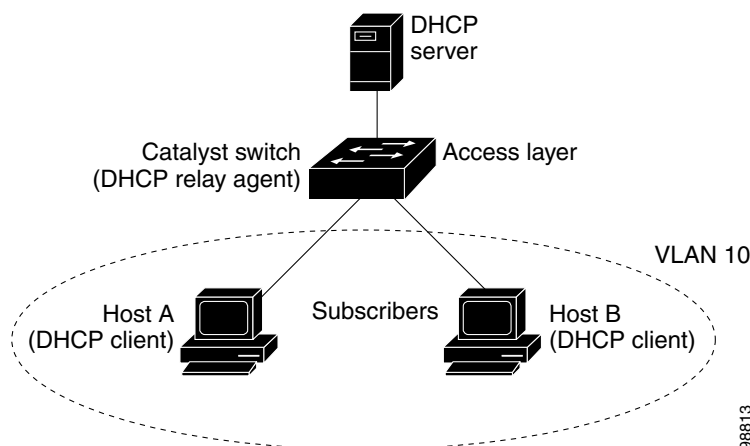
- The switch receives a packet from a DHCP server, such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet, from outside the network or firewall.
- The switch receives a packet on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match.
- The switch receives a DHCPRELEASE or DHCPDECLINE message that contains a MAC address in the DHCP snooping binding table, but the interface information in the binding table does not match the interface on which the message was received.
- The switch receives a DHCP packet that includes a relay agent IP address that is not 0.0.0.0.
- The switch receives a packet that includes option-82 information on an untrusted port.

DHCP Snooping Option-82 Data Insertion

In residential, metropolitan Ethernet-access environments, DHCP can centrally manage the IP address assignments for a large number of subscribers. When the DHCP snooping option-82 feature is enabled on the switch, a subscriber device is identified by the switch port through which it connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can be connected to the same port on the access switch and are uniquely identified.

Figure 38-1 is an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the switch at the access layer. Because the DHCP clients and their associated DHCP server do not reside on the same IP network or subnet, a DHCP relay agent is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.

Figure 38-1 DHCP Relay Agent in a Metropolitan Ethernet Network



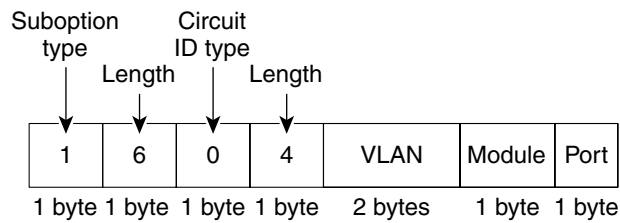
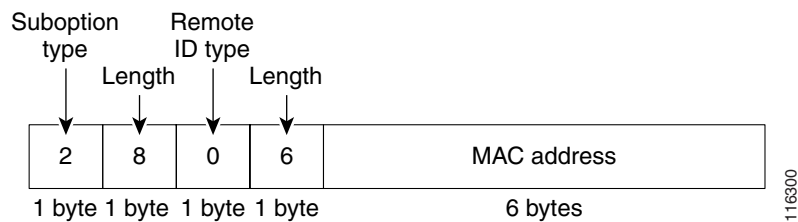
When you enable the DHCP snooping information option 82 on the switch, this sequence of events occurs:

- The host (DHCP client) generates a DHCP request and broadcasts it on the network.
- When the switch receives the DHCP request, it adds the option-82 information in the packet. The option-82 information contains the switch MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption).
- If the IP address of the relay agent is configured, the switch adds the IP address in the DHCP packet.
- The switch forwards the DHCP request that includes the option-82 field to the DHCP server.
- The DHCP server receives the packet. If the server is option-82 capable, it can use the remote ID, or the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server then echoes the option-82 field in the DHCP reply.
- The DHCP server unicasts the reply to the switch if the request was relayed to the server by the switch. When the client and server are on the same subnet, the server broadcasts the reply. The switch verifies that it originally inserted the option-82 data by inspecting the remote ID and possibly the circuit ID fields. The switch removes the option-82 field and forwards the packet to the switch port that connects to the DHCP client that sent the DHCP request.

When the previously described sequence of events occurs, the values in these fields in [Figure 38-2](#) 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 38-2](#) shows the packet formats for the remote ID suboption and the circuit ID suboption. The switch uses the packet formats when DHCP snooping is globally enabled and when the **ip dhcp snooping information option** global configuration command is entered. For the circuit ID suboption, the module field is the slot number of the module.

Figure 38-2 Suboption Packet Formats**Circuit ID Suboption Frame Format****Remote ID Suboption Frame Format**

Overview of the DHCP Snooping Database Agent

To retain the bindings across reloads, you must use the DHCP snooping database agent. Without this agent, the bindings established by DHCP snooping are lost upon reload, and connectivity is lost as well.

The database agent stores the bindings in a file at a configured location. Upon reload, the switch reads the file to build the database for the bindings. The switch keeps the file current by writing to the file as the database changes.

The format of the file that contains the bindings is as follows:

```
<initial-checksum>
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
<entry-1> <checksum-1>
<entry-2> <checksum-1-2>
...
...
<entry-n> <checksum-1-2-...-n>
END
```

Each entry in the file is tagged with a checksum that is used to validate the entries whenever the file is read. The <initial-checksum> entry on the first line helps distinguish entries associated with the latest write from entries that are associated with a previous write.

This is a sample bindings file:

```
3ebe1518
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
1.1.1.1 512 0001.0001.0005 3EBE2881 Gi1/1 e5e1e733
1.1.1.1 512 0001.0001.0002 3EBE2881 Gi1/1 4b3486ec
1.1.1.1 1536 0001.0001.0004 3EBE2881 Gi1/1 f0e02872
```

```

1.1.1.1 1024 0001.0001.0003 3EBE2881 Gi1/1          ac41adf9
1.1.1.1 1 0001.0001.0001 3EBE2881 Gi1/1            34b3273e
END

```

Each entry holds an IP address, VLAN, MAC address, lease time (in hex), and the interface associated with a binding. At the end of each entry is a checksum that is based on all the bytes from the start of the file through all the bytes associated with the entry. Each entry consists of 72 bytes of data, followed by a space, followed by a checksum.

Upon bootup, when the calculated checksum equals the stored checksum, the switch reads entries from the file and adds the bindings to the DHCP snooping database. If the calculated checksum does not equal the stored checksum, the entry read from the file is ignored and so are all the entries following the failed entry. The switch also ignores all those entries from the file whose lease time has expired. (This is possible because the lease time might indicate an expired time.) An entry from the file is also ignored if the interface referred to in the entry no longer exists on the system, or if it is a router port or a DHCP snooping-trusted interface.

When the switch learns of new bindings or when it loses some bindings, the switch writes the modified set of entries from the snooping database to the file. The writes are performed with a configurable delay to batch as many changes as possible before the actual write happens. Associated with each transfer is a timeout after which a transfer is aborted if it is not completed. These timers are referred to as the write delay and abort timeout.

Default Configuration for DHCP Snooping

Table 38-1 shows all the default configuration values for each DHCP snooping option.

Table 38-1 Default Configuration Values for DHCP Snooping

Option	Default Value/State
DHCP snooping	Disabled
DHCP snooping information option	Enabled
DHCP snooping limit rate	None
DHCP snooping trust	Untrusted
DHCP snooping vlan	Disabled

DHCP Snooping Configuration Guidelines and Restrictions

When configuring DHCP snooping, follow these guidelines and restrictions:

- Before globally enabling DHCP snooping on the switch, make sure that the devices acting as the DHCP server and the DHCP relay agent are configured and enabled.
- For DHCP server configuration information, refer to “Configuring DHCP” in the *Cisco IOS IP and IP Routing Configuration Guide* at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcprt1/1cfdhcp.htm

- When DHCP snooping is enabled, these Cisco IOS DHCP commands are not available on the switch:

- **ip dhcp relay information check** global configuration command
- **ip dhcp relay information policy** global configuration command
- **ip dhcp relay information trust-all** global configuration command
- **ip dhcp relay information option** global configuration command
- **ip dhcp relay information trusted** interface configuration command

If you enter these commands, the switch returns an error message, and the configuration is not applied.

- To use any DHCP snooping features, you must globally enable DHCP snooping on the switch.
- DHCP snooping is not active until DHCP snooping is enabled on a VLAN.
- If a Layer 2 LAN port is connected to a DHCP server, configure the port as trusted by entering the **ip dhcp snooping trust** interface configuration command.
- If a Layer 2 LAN port is connected to a DHCP client, configure the port as untrusted by entering the **no ip dhcp snooping trust** interface configuration command.
- You can enable DHCP snooping on private VLANs:
 - If DHCP snooping is enabled, any primary VLAN configuration is propagated to its associated secondary VLANs.
 - If DHCP snooping is configured on the primary VLAN and you configure DHCP snooping with different settings on an associated secondary VLAN, the configuration on the secondary VLAN does not take effect.
 - If DHCP snooping is not configured on the primary VLAN and you configure DHCP snooping on a secondary VLAN, the configuration takes affect only on the secondary VLAN.
 - When you manually configure DHCP snooping on a secondary VLAN, this message appears:
DHCP Snooping configuration may not take effect on secondary vlan XXX
 - The **show ip dhcp snooping** command displays all VLANs (both primary and secondary) that have DHCP snooping enabled.

Configuring DHCP Snooping

These sections describe how to configure DHCP snooping:

- [Enabling DHCP Snooping Globally, page 38-7](#)
- [Enabling DHCP Option-82 Data Insertion, page 38-7](#)
- [Enabling DHCP Snooping MAC Address Verification, page 38-8](#)
- [Enabling DHCP Snooping on VLANs, page 38-8](#)
- [Configuring the DHCP Trust State on Layer 2 LAN Interfaces, page 38-9](#)
- [Configuring DHCP Snooping Rate Limiting on Layer 2 LAN Interfaces, page 38-10](#)
- [Configuring the DHCP Snooping Database Agent, page 38-11](#)
- [Configuration Examples for the Database Agent, page 38-11](#)
- [Displaying a Binding Table, page 38-14](#)

Enabling DHCP Snooping Globally



Note

Enable this feature during a maintenance window, because after you enable DHCP snooping globally, the switch drops DHCP requests until you configure the ports.

To enable DHCP snooping globally, perform this task:

	Command	Purpose
Step 1	Router(config)# ip dhcp snooping	Enables DHCP snooping globally.
	Router(config)# no ip dhcp snooping	Disables DHCP snooping.
Step 2	Router(config)# do show ip dhcp snooping include Switch	Verifies the configuration.

This example shows how to enable DHCP snooping globally:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip dhcp snooping
Router(config)# do show ip dhcp snooping | include Switch
Switch DHCP snooping is enabled
Router(config)#
```

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 DHCP Snooping MAC Address Verification

With DHCP snooping MAC address verification enabled, DHCP snooping verifies that the source MAC address in DHCP packets that are received on untrusted ports match the client hardware address in the packet.

To enable DHCP snooping MAC address verification, perform this task:

	Command	Purpose
Step 1	Router(config)# ip dhcp snooping verify mac-address	Enables DHCP snooping MAC address verification.
	Router(config)# no ip dhcp snooping verify mac-address	Disables DHCP snooping MAC address verification.
Step 2	Router(config)# do show ip dhcp snooping include hwaddr	Verifies the configuration.

This example shows how to disable DHCP snooping MAC address verification:

```
Router(config)# no ip dhcp snooping verify mac-address
Router(config)# do show ip dhcp snooping | include hwaddr
Verification of hwaddr field is disabled
Router(config)#
```

This example shows how to enable DHCP snooping MAC address verification:

```
Router(config)# ip dhcp snooping verify mac-address
Router(config)# do show ip dhcp snooping | include hwaddr
Verification of hwaddr field is enabled
Router(config)#
```

Enabling DHCP Snooping on VLANs

To enable DHCP snooping on VLANs, perform this task:

	Command	Purpose
Step 1	Router(config)# ip dhcp snooping vlan {{vlan_ID [vlan_ID]} {vlan_range}}	Enables DHCP snooping on a VLAN or VLAN range.
	Router(config)# no ip dhcp snooping	Disables DHCP snooping.
Step 2	Router(config)# do show ip dhcp snooping	Verifies the configuration.

You can configure DHCP snooping for a single VLAN or a range of VLANs:

- To configure a single VLAN, enter a single VLAN number.
- To configure a range of VLANs, enter a beginning and an ending VLAN number or a dash-separated pair of VLAN numbers.
- You can enter a comma-separated list of VLAN numbers and dash-separated pairs of VLAN numbers.

This example shows how to enable DHCP snooping on VLANs 10 through 12:

```
Router# configure terminal
Router(config)# ip dhcp snooping vlan 10 12
Router(config)#
```

This example shows another way to enable DHCP snooping on VLANs 10 through 12:

```
Router# configure terminal
Router(config)# ip dhcp snooping vlan 10-12
```

This example shows another way to enable DHCP snooping on VLANs 10 through 12:

```
Router# configure terminal
Router(config)# ip dhcp snooping vlan 10,11,12
```

This example shows how to enable DHCP snooping on VLANs 10 through 12 and VLAN 15:

```
Router# configure terminal
Router(config)# ip dhcp snooping vlan 10-12,15
```

This example shows how to verify the configuration:

```
Router(config)# do show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10-12,15
DHCP snooping is operational on following VLANs:
none
DHCP snooping is configured on the following Interfaces:

Insertion of option 82 is enabled
Verification of hwaddr field is enabled
Interface           Trusted      Rate limit (pps)
-----
Router#
```

Configuring the DHCP Trust State on Layer 2 LAN Interfaces

To configure DHCP trust state on a Layer 2 LAN interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {type ¹ slot/port port-channel number}	Selects the interface to configure. Note Select only LAN ports configured with the switchport command or Layer 2 port-channel interfaces.
Step 2	Router(config-if)# ip dhcp snooping trust Router(config-if)# no ip dhcp snooping trust	Configures the interface as trusted. Reverts to the default (untrusted) state.
Step 3	Router(config-if)# do show ip dhcp snooping begin pps	Verifies the configuration.

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

This example shows how to configure Fast Ethernet port 5/12 as trusted:

```
Router# configure terminal
Router(config)# interface FastEthernet 5/12
Router(config-if)# ip dhcp snooping trust
```

```

Router(config-if)# do show ip dhcp snooping | begin pps
Interface                               Trusted      Rate limit (pps)
-----
FastEthernet5/12                        yes          unlimited
Router#

```

This example shows how to configure Fast Ethernet port 5/12 as trusted:

```

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.
Router# renew ip dhcp snooping database [validation none] [url]	(Optional) Requests the read entries from a file at the given URL.
Router# ip dhcp snooping binding mac_address vlan vlan_ID ip_address interface ifname expiry lease_in_seconds	(Optional) Adds bindings to the snooping database.
Router# no ip dhcp snooping binding mac_address vlan vlan_ID ip_address interface ifname	(Optional) Deletes bindings to the snooping database.

When configuring the DHCP snooping database agent, note the following information:

- Store the file on a TFTP server to avoid consuming storage space on the switch storage devices.
- When a switchover occurs, if the file is stored in a remote location accessible through TFTP, the newly active supervisor engine can use the binding list.
- Network-based URLs (such as TFTP and FTP) require that you create an empty file at the configured URL before the switch can write the set of bindings for the first time.

Configuration Examples for the Database Agent

These sections provide examples for the database agent:

- [Example 1: Enabling the Database Agent, page 38-11](#)
- [Example 2: Reading Binding Entries from a TFTP File, page 38-13](#)
- [Example 3: Adding Information to the DHCP Snooping Database, page 38-14](#)

Example 1: Enabling the Database Agent

The following example shows how to configure the DHCP snooping database agent to store the bindings at a given location and to view the configuration and operating state:

```
Router# configure terminal
Router(config)# ip dhcp snooping database tftp://10.1.1.1/directory/file
Router(config)# end
Router# show ip dhcp snooping database detail
Agent URL : tftp://10.1.1.1/directory/file
Write delay Timer : 300 seconds
Abort Timer : 300 seconds
```

```

Agent Running : No
Delay Timer Expiry : 7 (00:00:07)
Abort Timer Expiry : Not Running

Last Succeeded Time : None
Last Failed Time : 17:14:25 UTC Sat Jul 7 2001
Last Failed Reason : Unable to access URL.

Total Attempts      :      21   Startup Failures :      0
Successful Transfers :      0   Failed Transfers  :     21
Successful Reads    :      0   Failed Reads    :      0
Successful Writes   :      0   Failed Writes   :     21
Media Failures      :      0

First successful access: Read

Last ignored bindings counters :
Binding Collisions :      0   Expired leases :      0
Invalid interfaces :      0   Unsupported vlans :      0
Parse failures     :      0

Last Ignored Time : None

Total ignored bindings counters:
Binding Collisions :      0   Expired leases :      0
Invalid interfaces :      0   Unsupported vlans :      0
Parse failures     :      0

Router#

```

The first three lines of output show the configured URL and related timer-configuration values. The next three lines show the operating state and the amount of time left for expiry of write delay and abort timers.

Among the statistics shown in the output, startup failures indicate the number of attempts to read or create the file failed upon bootup.



Note

Create a temporary file on the TFTP server with the **touch** command in the TFTP server daemon directory. With some UNIX implementations, the file should have full read and write access permissions (777).

DHCP snooping bindings are keyed on the MAC address and VLAN combination. If an entry in the remote file has an entry for a given MAC address and VLAN set for which the switch already has a binding, the entry from the remote file is ignored when the file is read. This condition is referred to as the *binding collision*.

An entry in a file may no longer be valid because the lease indicated by the entry may have expired by the time it is read. The expired leases counter indicates the number of bindings that are ignored because of this condition. The Invalid interfaces counter refers to the number of bindings that have been ignored when the interface referred by the entry either does not exist on the system or is a router or DHCP snooping trusted interface (if it exists) when the read happened. Unsupported VLANs refers to the number of entries that have been ignored because the indicated VLAN is not supported on the system. The Parse failures counter provides the number of entries that have been ignored when the switch is unable to interpret the meaning of the entries from the file.

The switch maintains two sets of counters for these ignored bindings. One provides the counters for a read that has at least one binding ignored by at least one of these conditions. These counters are shown as the “Last ignored bindings counters.” The total ignored bindings counters provides a sum of the

number of bindings that have been ignored because of all the reads since the switch bootup. These two set of counters are cleared by the **clear** command. Therefore, the total counter set may indicate the number of bindings that have been ignored since the last clear.

Example 2: Reading Binding Entries from a TFTP File

To manually read the entries from a TFTP file, perform this task:

	Command	Purpose
Step 1	Router# show ip dhcp snooping database	Displays the DHCP snooping database agent statistics.
Step 2	Router# renew ip dhcp snoop data url	Directs the switch to read the file from the URL.
Step 3	Router# show ip dhcp snoop data	Displays the read status.
Step 4	Router# show ip dhcp snoop bind	Verifies whether the bindings were read successfully.

This is an example of how to manually read entries from the tftp://10.1.1.1/directory/file:

```
Router# show ip dhcp snooping database
Agent URL :
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : Not Running
Abort Timer Expiry : Not Running

Last Succeeded Time : None
Last Failed Time : None
Last Failed Reason : No failure recorded.

Total Attempts      :          0   Startup Failures :          0
Successful Transfers :          0   Failed Transfers :          0
Successful Reads     :          0   Failed Reads    :          0
Successful Writes    :          0   Failed Writes   :          0
Media Failures       :          0

Router# renew ip dhcp snoop data tftp://10.1.1.1/directory/file
Loading directory/file from 10.1.1.1 (via GigabitEthernet1/1): !
[OK - 457 bytes]
Database downloaded successfully.

Router#
00:01:29: %DHCP_SNOOPING-6-AGENT_OPERATION_SUCCEEDED: DHCP snooping database Read
succeeded.
Router# show ip dhcp snoop data
Agent URL :
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : Not Running
Abort Timer Expiry : Not Running

Last Succeeded Time : 15:24:34 UTC Sun Jul 8 2001
Last Failed Time : None
Last Failed Reason : No failure recorded.

Total Attempts      :          1   Startup Failures :          0
Successful Transfers :          1   Failed Transfers :          0
```

```

Successful Reads      :          1   Failed Reads      :          0
Successful Writes    :          0   Failed Writes    :          0
Media Failures       :          0
Router#
Router# show ip dhcp snoop bind
-----
MacAddress      IPAddress      Lease(sec)   Type          VLAN   Interface
-----
00:01:00:01:00:05  1.1.1.1       49810        dhcp-snooping  512    GigabitEthernet1/1
00:01:00:01:00:02  1.1.1.1       49810        dhcp-snooping  512    GigabitEthernet1/1
00:01:00:01:00:04  1.1.1.1       49810        dhcp-snooping  1536   GigabitEthernet1/1
00:01:00:01:00:03  1.1.1.1       49810        dhcp-snooping  1024   GigabitEthernet1/1
00:01:00:01:00:01  1.1.1.1       49810        dhcp-snooping  1      GigabitEthernet1/1
Router# clear ip dhcp snoop bind
Router# show ip dhcp snoop bind
-----
MacAddress      IPAddress      Lease(sec)   Type          VLAN   Interface
-----
Router#

```

Example 3: Adding Information to the DHCP Snooping Database

To manually add a binding to the DHCP snooping database, perform 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 <i>binding_id</i> vlan <i>vlan_id</i> interface <i>interface</i> expiry <i>lease_time</i>	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 1.1.1.1 vlan 1 1.1.1.1 interface gi1/1 expiry 1000

Router# show ip dhcp snooping binding
-----
MacAddress      IPAddress      Lease(sec)   Type          VLAN   Interface
-----
00:01:00:01:00:01  1.1.1.1       992          dhcp-snooping  1      GigabitEthernet1/1
Router#

```

Displaying a Binding Table

The DHCP snooping binding table for each switch contains binding entries that correspond to untrusted ports. The table does not contain information about hosts interconnected with a trusted port because each interconnected switch will have its own DHCP snooping binding table.

This example shows how to display the DHCP snooping binding information for a switch:

```

Router# show ip dhcp snooping binding
-----
MacAddress      IPAddress      Lease(sec)   Type          VLAN   Interface
-----
00:02:B3:3F:3B:99  55.5.5.2       6943        dhcp-snooping  10     FastEthernet6/10

```

[Table 38-2](#) describes the fields in the **show ip dhcp snooping binding** command output.

Table 38-2 *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



Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) on the Catalyst 6500 series switch. Release 12.2(18)SX and later releases support DAI.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding DAI, page 39-1](#)
- [Default DAI Configuration, page 39-5](#)
- [DAI Configuration Guidelines and Restrictions, page 39-5](#)
- [Configuring DAI, page 39-6](#)
- [DAI Configuration Samples, page 39-16](#)

Understanding DAI

These sections describe how DAI helps prevent ARP spoofing attacks:

- [Understanding ARP, page 39-1](#)
- [Understanding ARP Spoofing Attacks, page 39-2](#)
- [Understanding DAI and ARP Spoofing Attacks, page 39-2](#)

Understanding ARP

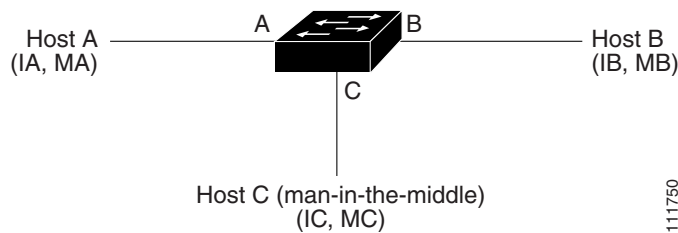
ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, Host B wants to send information to Host A but does not have the MAC address of Host A in its ARP cache. Host B generates a broadcast message for all hosts within the broadcast domain 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 39-1](#) shows an example of ARP cache poisoning.

Figure 39-1 ARP Cache Poisoning



Hosts A, B, and C are connected to the switch on interfaces A, B and C, all of which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, Host A uses IP address IA and MAC address MA. When Host A needs to communicate to Host B at the IP layer, it broadcasts an ARP request for the MAC address associated with IP address IB. When the switch and Host B receive the ARP request, they populate their ARP caches with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When Host B responds, the switch and Host A populate their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can poison the ARP caches of the switch, Host A, and Host B by broadcasting forged ARP responses with bindings for a host with an IP address of IA (or IB) and a MAC address of MC. Hosts with poisoned ARP caches use the MAC address MC as the destination MAC address for traffic intended for IA or IB. This means that Host C intercepts that traffic. Because Host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. Host C has inserted itself into the traffic stream from Host A to Host B, which is the topology of the classic *man-in-the middle* attack.

Understanding DAI and ARP Spoofing Attacks

DAI is a security feature that validates ARP packets in a network. DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. This capability protects the network from some man-in-the-middle attacks.

DAI ensures that only valid ARP requests and responses are relayed. The switch performs these activities:

- Intercepts all ARP requests and responses on untrusted ports
- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

DAI determines the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a trusted database, the DHCP snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the switch. If the ARP packet is received on a trusted interface, the switch forwards the packet without any checks. On untrusted interfaces, the switch forwards the packet only if it is valid.

DAI can validate ARP packets against user-configured ARP access control lists (ACLs) for hosts with statically configured IP addresses (see [“Applying ARP ACLs for DAI Filtering”](#) section on page 39-8). The switch logs dropped packets (see the [“Logging of Dropped Packets”](#) section on page 39-4).

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 39-11).

Interface Trust States and Network Security

DAI associates a trust state with each interface on the switch. Packets arriving on trusted interfaces bypass all DAI validation checks, and those arriving on untrusted interfaces undergo the DAI validation process.

In a typical network configuration, you configure all switch ports connected to host ports as untrusted and configure all switch ports connected to switches as trusted. With this configuration, all ARP packets entering the network from a given switch bypass the security check. No other validation is needed at any other place in the VLAN or in the network. You configure the trust setting by using the **ip arp inspection trust** interface configuration command.

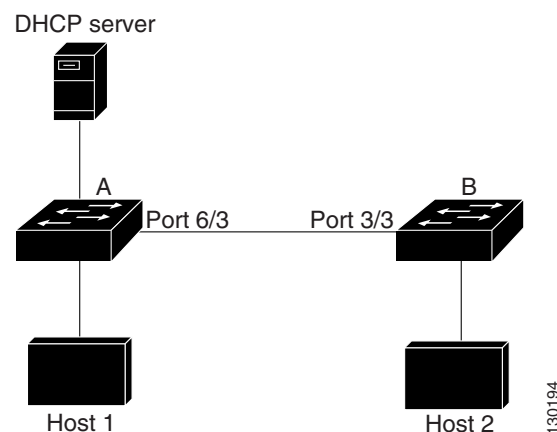


Caution

Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

In [Figure 39-2](#), assume that both Switch A and Switch B are running DAI on the VLAN that includes Host 1 and Host 2. If Host 1 and Host 2 acquire their IP addresses from the DHCP server connected to Switch A, only Switch A binds the IP-to-MAC address of Host 1. Therefore, if the interface between Switch A and Switch B is untrusted, the ARP packets from Host 1 are dropped by Switch B. Connectivity between Host 1 and Host 2 is lost.

Figure 39-2 ARP Packet Validation on a VLAN Enabled for DAI



Configuring interfaces to be trusted when they are actually untrusted leaves a security hole in the network. If Switch A is not running DAI, Host 1 can easily poison the ARP cache of Switch B (and Host 2, if the link between the switches is configured as trusted). This condition can occur even though Switch B is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a switch running DAI do not poison the ARP caches of other hosts in the network. However, DAI does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a switch running DAI.

In cases in which some switches in a VLAN run DAI and other switches do not, configure the interfaces connecting such switches as untrusted. However, to validate the bindings of packets from switches where DAI is not configured, configure ARP ACLs on the switch running DAI. When you cannot determine such bindings, isolate switches running DAI at Layer 3 from switches not running DAI. For configuration information, see the [“Sample Two: One Switch Supports DAI”](#) section on page 39-20.

**Note**

Depending on the setup of the DHCP server and the network, it might not be possible to validate a given ARP packet on all switches in the VLAN.

Rate Limiting of ARP Packets

The switch performs DAI validation checks, which rate limits incoming ARP packets to prevent a denial-of-service attack. By default, the rate for untrusted interfaces is 15 packets per second (pps). Trusted interfaces are not rate limited. You can change this setting by using the **ip arp inspection limit** interface configuration command.

When the rate of incoming ARP packets exceeds the configured limit, the switch places the port in the error-disabled state. The port remains in that state until you intervene. You can use the **errdisable recovery** global configuration command to enable error disable recovery so that ports automatically emerge from this state after a specified timeout period.

For configuration information, see the [“Configuring ARP Packet Rate Limiting”](#) section on page 39-9.

Relative Priority of ARP ACLs and DHCP Snooping Entries

DAI uses the DHCP snooping binding database for the list of valid IP-to-MAC address bindings.

ARP ACLs take precedence over entries in the DHCP snooping binding database. The switch uses ACLs only if you configure them by using the **ip arp inspection filter** global configuration command. The switch first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the switch also denies the packet even if a valid binding exists in the database populated by DHCP snooping.

Logging of Dropped Packets

When the switch drops a packet, it places an entry in the log buffer and then generates system messages on a rate-controlled basis. After the message is generated, the switch clears the entry from the log buffer. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You use the **ip arp inspection log-buffer** global configuration command to configure the number of entries in the buffer and the number of entries needed in the specified interval to generate system messages. You specify the type of packets that are logged by using the **ip arp inspection vlan logging** global configuration command. For configuration information, see the [“Configuring DAI Logging” section on page 39-12](#).

Default DAI Configuration

Table 39-1 shows the default DAI configuration.

Table 39-1 Default DAI Configuration

Feature	Default Setting
DAI	Disabled on all VLANs.
Interface trust state	All interfaces are untrusted.
Rate limit of incoming ARP packets	The rate is 15 pps on untrusted interfaces, assuming that the network is a Layer 2-switched network with a host connecting to as many as 15 new hosts per second. The rate is unlimited on all trusted interfaces. The burst interval is 1 second.
ARP ACLs for non-DHCP environments	No ARP ACLs are defined.
Validation checks	No checks are performed.
Log buffer	When DAI is enabled, all denied or dropped ARP packets are logged. The number of entries in the log is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second.
Per-VLAN logging	All denied or dropped ARP packets are logged.

DAI Configuration Guidelines and Restrictions

When configuring DAI, follow these guidelines and restrictions:

- DAI is an ingress security feature; it does not perform any egress checking.
- DAI is not effective for hosts connected to switches that do not support DAI or that do not have this feature enabled. Because man-in-the-middle attacks are limited to a single Layer 2 broadcast domain, separate the domain with DAI checks from the one with no checking. This action secures the ARP caches of hosts in the domain enabled for DAI.
- DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses. For configuration information, see [Chapter 38, “Configuring DHCP Snooping.”](#)

- When DHCP snooping is disabled or in non-DHCP environments, use ARP ACLs to permit or to deny packets.
- DAI is supported on access ports, trunk ports, EtherChannel ports, and private VLAN ports.
- A physical port can join an EtherChannel port channel only when the trust state of the physical port and the channel port match. Otherwise, the physical port remains suspended in the port channel. A port channel inherits its trust state from the first physical port that joins the channel. Consequently, the trust state of the first physical port need not match the trust state of the channel.

Conversely, when you change the trust state on the port channel, the switch configures a new trust state on all the physical ports that comprise the channel.

- The operating rate for the port channel is cumulative across all the physical ports within the channel. For example, if you configure the port channel with an ARP rate-limit of 400 pps, all the interfaces combined on the channel receive an aggregate 400 pps. The rate of incoming ARP packets on EtherChannel ports is equal to the sum of the incoming rate of packets from all the channel members. Configure the rate limit for EtherChannel ports only after examining the rate of incoming ARP packets on the channel-port members.

The rate of incoming packets on a physical port is checked against the port-channel configuration rather than the physical-ports configuration. The rate-limit configuration on a port channel is independent of the configuration on its physical ports.

If the EtherChannel receives more ARP packets than the configured rate, the channel (including all physical ports) is placed in the error-disabled state.

- Make sure to limit the rate of ARP packets on incoming trunk ports. Configure trunk ports with higher rates to reflect their aggregation and to handle packets across multiple DAI-enabled VLANs. You also can use the **ip arp inspection limit none** interface configuration command to make the rate unlimited. A high rate-limit on one VLAN can cause a denial-of-service attack to other VLANs when the software places the port in the error-disabled state.

Configuring DAI

These sections describe how to configure DAI:

- [Enabling DAI on VLANs, page 39-7](#)
- [Configuring the DAI Interface Trust State, page 39-7](#)
- [Applying ARP ACLs for DAI Filtering, page 39-8](#)
- [Configuring ARP Packet Rate Limiting, page 39-9](#)
- [Enabling DAI Error-Disabled Recovery, page 39-10](#)
- [Enabling Additional Validation, page 39-11](#)
- [Configuring DAI Logging, page 39-12](#)
- [Displaying DAI Information, page 39-15](#)

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 switch does not check ARP packets that it receives from the other switch on the trusted interface. It simply forwards the packets.

On untrusted interfaces, the switch intercepts all ARP requests and responses. It verifies that the intercepted packets have valid IP-to-MAC address bindings before updating the local cache and before forwarding the packet to the appropriate destination. The switch drops invalid packets and logs them in the log buffer according to the logging configuration specified with the **ip arp inspection vlan logging** global configuration command. For more information, see the “[Configuring DAI Logging](#)” section on page 39-12.

To configure the DAI interface trust state, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# interface {type ¹ slot/port port-channel number}	Specifies the interface connected to another switch, and enter interface configuration mode.
Step 3	Router(config-if)# ip arp inspection trust	Configures the connection between switches as trusted (default: untrusted).
	Router(config)# no ip arp inspection trust	Configures the connection between switches as untrusted.
Step 4	Router(config-if)# do show ip arp inspection interfaces	Verify the DAI configuration.

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

This example shows how to configure Fast Ethernet port 5/12 as trusted:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/12
Router(config-if)# ip arp inspection trust
Router(config-if)# do show ip arp inspection interfaces | include Int|--|5/12
Interface          Trust State      Rate (pps)      Burst Interval
-----
Fa5/12             Trusted          None            N/A
```

Applying ARP ACLs for DAI Filtering



Note

See the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX, for information about the **arp access-list** command.

To apply an ARP ACL, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router# ip arp inspection filter arp_acl_name vlan {vlan_ID vlan_range} [static]	Applies the ARP ACL to a VLAN.
Step 3	Router(config)# do show ip arp inspection vlan {vlan_ID vlan_range}	Verifies your entries.

When applying ARP ACLs, note the following information:

- For *vlan_range*, you can specify a single VLAN or a range of VLANs:
 - To specify a single VLAN, enter a single VLAN number.
 - To specify a range of VLANs, enter a dash-separated pair of VLAN numbers.
 - You can enter a comma-separated list of VLAN numbers and dash-separated pairs of VLAN numbers.
- (Optional) Specify **static** to treat implicit denies in the ARP ACL as explicit denies and to drop packets that do not match any previous clauses in the ACL. DHCP bindings are not used.

If you do not specify this keyword, it means that there is no explicit deny in the ACL that denies the packet, and DHCP bindings determine whether a packet is permitted or denied if the packet does not match any clauses in the ACL.

- ARP packets containing only IP-to-MAC address bindings are compared against the ACL. Packets are permitted only if the access list permits them.

This example shows how to apply an ARP ACL named `example_arp_acl` to VLANs 10 through 12 and VLAN 15:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip arp inspection filter example_arp_acl vlan 10-12,15
Router(config)# do show ip arp inspection vlan 10-12,15 | begin Vlan
```

Vlan	Configuration	Operation	ACL Match	Static ACL
10	Enabled	Inactive	example_arp_acl	No
11	Enabled	Inactive	example_arp_acl	No
12	Enabled	Inactive	example_arp_acl	No
15	Enabled	Inactive	example_arp_acl	No

Vlan	ACL Logging	DHCP Logging
10	Deny	Deny
11	Deny	Deny
12	Deny	Deny
15	Deny	Deny

Configuring ARP Packet Rate Limiting

When DAI is enabled, the switch performs ARP packet validation checks, which makes the switch vulnerable to an ARP-packet denial-of-service attack. ARP packet rate limiting can prevent an ARP-packet denial-of-service attack.

To configure ARP packet rate limiting on a port, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# interface {type ¹ slot/port port-channel number}	Selects the interface to be configured.
Step 3	Router(config-if)# ip arp inspection limit {rate pps [burst interval seconds] none} Router(config-if)# no ip arp inspection limit	(Optional) Configures ARP packet rate limiting. Clears the ARP packet rate-limiting configuration.
Step 4	Router(config-if)# do show ip arp inspection interfaces	Verifies the configuration.

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

When configuring ARP packet rate limiting, note the following information:

- The default rate is 15 pps on untrusted interfaces and unlimited on trusted interfaces.
- For **rate** *pps*, specify an upper limit for the number of incoming packets processed per second. The range is 0 to 2048 pps.
- The **rate none** keywords specify that there is no upper limit for the rate of incoming ARP packets that can be processed.
- (Optional) For **burst interval** *seconds* (default is 1), specify the consecutive interval, in seconds, over which the interface is monitored for a high rate of ARP packets. The range is 1 to 15.
- When the rate of incoming ARP packets exceeds the configured limit, the switch places the port in the error-disabled state. The port remains in the error-disabled state until you enable error-disabled recovery, which allows the port to emerge from the error-disabled state after a specified timeout period.
- Unless you configure a rate-limiting value on an interface, changing the trust state of the interface also changes its rate-limiting value to the default value for the configured trust state. After you configure the rate-limiting value, the interface retains the rate-limiting value even when you change its trust state. If you enter the **no ip arp inspection limit** interface configuration command, the interface reverts to its default rate-limiting value.
- For configuration guidelines about limiting the rate of incoming ARP packets on trunk ports and EtherChannel ports, see the [“DAI Configuration Guidelines and Restrictions” section on page 39-5](#).

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, page 39-12](#)
- [Configuring the DAI Logging Buffer Size, page 39-13](#)
- [Configuring the DAI Logging System Messages, page 39-13](#)
- [Configuring DAI Log Filtering, page 39-14](#)

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 <i>number</i>	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 <i>number_of_messages interval length_in_seconds</i>	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 <i>vlan_range</i> 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 <i>vlan_range</i>	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 39-2](#).

Table 39-2 Commands for Displaying DAI Information

Command	Description
show arp access-list [<i>acl_name</i>]	Displays detailed information about ARP ACLs.
show ip arp inspection interfaces [<i>interface_id</i>]	Displays the trust state and the rate limit of ARP packets for the specified interface or all interfaces.
show ip arp inspection vlan <i>vlan_range</i>	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 39-3](#).

Table 39-3 Commands for Clearing or Displaying DAI Statistics

Command	Description
clear ip arp inspection statistics	Clears DAI statistics.
show ip arp inspection statistics [<i>vlan vlan_range</i>]	Displays statistics for forwarded, dropped, MAC validation failure, IP validation failure, ACL permitted and denied, and DHCP permitted and denied packets for the specified VLAN. If no VLANs are specified or if a range is specified, displays information only for VLANs with DAI enabled (active).

For the **show ip arp inspection statistics** command, the switch increments the number of forwarded packets for each ARP request and response packet on a trusted DAI port. The switch increments the number of ACL-permitted or DHCP-permitted packets for each packet that is denied by source MAC, destination MAC, or IP validation checks, and the switch increments the appropriate failure count.

To clear or display DAI logging information, use the privileged EXEC commands in [Table 39-4](#):

Table 39-4 *Commands for Clearing or Displaying DAI Logging Information*

Command	Description
<code>clear ip arp inspection log</code>	Clears the DAI log buffer.
<code>show ip arp inspection log</code>	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 39-16](#)
- [Sample Two: One Switch Supports DAI, page 39-20](#)

Sample One: Two Switches Support DAI

This procedure shows how to configure DAI when two switches support this feature. Host 1 is connected to Switch A, and Host 2 is connected to Switch B as shown in [Figure 39-2 on page 39-3](#). Both switches are running DAI on VLAN 1 where the hosts are located. A DHCP server is connected to Switch A. Both hosts acquire their IP addresses from the same DHCP server. Switch A has the bindings for Host 1 and Host 2, and Switch B has the binding for Host 2. Switch A Fast Ethernet port 6/3 is connected to the Switch B Fast Ethernet port 3/3.



Note

- DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses. For configuration information, see [Chapter 38, “Configuring DHCP Snooping.”](#)
- This configuration does not work if the DHCP server is moved from Switch A to a different location.
- To ensure that this configuration does not compromise security, configure Fast Ethernet port 6/3 on Switch A and Fast Ethernet port 3/3 on Switch B as trusted.

Configuring Switch A

To enable DAI and configure Fast Ethernet port 6/3 on Switch A as trusted, follow these steps:

Step 1

Verify the connection between switches Switch A and Switch B:

```
SwitchA# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID         Local Intrfce   Holdtme    Capability  Platform  Port ID
SwitchB           Fas 6/3        177        R S I      WS-C6506  Fas 3/3
SwitchA#
```

Step 2 Enable DAI on VLAN 1 and verify the configuration:

```
SwitchA# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchA(config)# ip arp inspection vlan 1
SwitchA(config)# end
SwitchA# show ip arp inspection vlan 1
```

Source Mac Validation	:	Disabled
Destination Mac Validation	:	Disabled
IP Address Validation	:	Disabled

Vlan	Configuration	Operation	ACL Match	Static ACL
----	-----	-----	-----	-----
1	Enabled	Active		

Vlan	ACL Logging	DHCP Logging
----	-----	-----
1	Deny	Deny

```
SwitchA#
```

Step 3 Configure Fast Ethernet port 6/3 as trusted:

```
SwitchA# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchA(config)# interface fastethernet 6/3
SwitchA(config-if)# ip arp inspection trust
SwitchA(config-if)# end
SwitchA# show ip arp inspection interfaces fastethernet 6/3
```

Interface	Trust State	Rate (pps)
-----	-----	-----
Fa6/3	Trusted	None

```
SwitchA#
```

Step 4 Verify the bindings:

```
SwitchA# show ip dhcp snooping binding
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
-----	-----	-----	-----	----	-----
00:02:00:02:00:02	1.1.1.2	4993	dhcp-snooping	1	FastEthernet6/4

```
SwitchA#
```

Step 5 Check the statistics before and after DAI processes any packets:

```
SwitchA# show ip arp inspection statistics vlan 1
```

Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
----	-----	-----	-----	-----
1	0	0	0	0

Vlan	DHCP Permits	ACL Permits	Source MAC Failures
----	-----	-----	-----
1	0	0	0

Vlan	Dest MAC Failures	IP Validation Failures
----	-----	-----
1	0	0

```
SwitchA#
```

If Host 1 then sends out two ARP requests with an IP address of 1.1.1.2 and a MAC address of 0002.0002.0002, both requests are permitted, as reflected in the following statistics:

```
SwitchA# show ip arp inspection statistics vlan 1
```

Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
----	-----	-----	-----	-----
1	2	0	0	0

Vlan	DHCP Permits	ACL Permits	Source MAC Failures
----	-----	-----	-----
1	2	0	0

Vlan	Dest MAC Failures	IP Validation Failures
----	-----	-----
1	0	0

```
SwitchA#
```

If Host 1 then tries to send an ARP request with an IP address of 1.1.1.3, the packet is dropped and an error message is logged:

```
00:12:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 2 Invalid ARPs (Req) on Fa6/4, vlan
1. ([0002.0002.0002/1.1.1.3/0000.0000.0000/0.0.0.0/02:42:35 UTC Tue Jul 10 2001])
```

```
SwitchA# show ip arp inspection statistics vlan 1
```

```
SwitchA#
```

The statistics will display as follows:

Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
----	-----	-----	-----	-----
1	2	2	2	0

Vlan	DHCP Permits	ACL Permits	Source MAC Failures
----	-----	-----	-----
1	2	0	0

Vlan	Dest MAC Failures	IP Validation Failures
----	-----	-----
1	0	0

```
SwitchA#
```

Configuring Switch B

To enable DAI and configure Fast Ethernet port 3/3 on Switch B as trusted, follow these steps:

Step 1 Verify the connectivity:

```
SwitchA# show cdp neighbors
```

```
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
```

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
SwitchB	Fas 3/3	120	R S I	WS-C6506	Fas 6/3

```
SwitchB#
```


Step 2 Enable DAI on VLAN 1, and verify the configuration:

```
SwitchB# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchB(config)# ip arp inspection vlan 1
SwitchB(config)# end
SwitchB# show ip arp inspection vlan 1
```

Source Mac Validation	:	Disabled
Destination Mac Validation	:	Disabled
IP Address Validation	:	Disabled

Vlan	Configuration	Operation	ACL Match	Static ACL
----	-----	-----	-----	-----
1	Enabled	Active		

Vlan	ACL Logging	DHCP Logging
----	-----	-----
1	Deny	Deny

```
SwitchB#
```

Step 3 Configure Fast Ethernet port 3/3 as trusted:

```
SwitchB# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchB(config)# interface fastethernet 3/3
SwitchB(config-if)# ip arp inspection trust
SwitchB(config-if)# end
SwitchB# show ip arp inspection interfaces
```

Interface	Trust State	Rate (pps)
-----	-----	-----
Gi1/1	Untrusted	15
Gi1/2	Untrusted	15
Gi3/1	Untrusted	15
Gi3/2	Untrusted	15
Fa3/3	Trusted	None
Fa3/4	Untrusted	15
Fa3/5	Untrusted	15
Fa3/6	Untrusted	15
Fa3/7	Untrusted	15

```
<output truncated>
SwitchB#
```

Step 4 Verify the list of DHCP snooping bindings:

```
SwitchB# show ip dhcp snooping binding
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
-----	-----	-----	-----	----	-----
00:01:00:01:00:01	1.1.1.1	4995	dhcp-snooping	1	FastEthernet3/4

```
SwitchB#
```

Step 5 Check the statistics before and after DAI processes any packets:

```
SwitchB# show ip arp inspection statistics vlan 1
```

Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
----	-----	-----	-----	-----
1	0	0	0	0

Vlan	DHCP Permits	ACL Permits	Source MAC Failures
----	-----	-----	-----
1	0	0	0

```

Vlan    Dest MAC Failures    IP Validation Failures
----    -
1              0              0
SwitchB#

```

If Host 2 then sends out an ARP request with the IP address 1.1.1.1 and the MAC address 0001.0001.0001, the packet is forwarded and the statistics are updated appropriately:

```
SwitchB# show ip arp inspection statistics vlan 1
```

```

Vlan    Forwarded    Dropped    DHCP Drops    ACL Drops
----    -
1              1              0              0              0

Vlan    DHCP Permits    ACL Permits    Source MAC Failures
----    -
1              1              0              0

Vlan    Dest MAC Failures    IP Validation Failures
----    -
1              0              0
SwitchB#

```

If Host 2 attempts to send an ARP request with the IP address 1.1.1.2, DAI drops the request and logs a system message:

```

00:18:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa3/4, vlan
1. ([0001.0001.0001/1.1.1.2/0000.0000.0000/0.0.0.0/01:53:21 UTC Fri May 23 2003])
SwitchB#

```

The statistics display as follows:

```
SwitchB# show ip arp inspection statistics vlan 1
```

```

Vlan    Forwarded    Dropped    DHCP Drops    ACL Drops
----    -
1              1              1              1              0

Vlan    DHCP Permits    ACL Permits    Source MAC Failures
----    -
1              1              0              0

Vlan    Dest MAC Failures    IP Validation Failures
----    -
1              0              0
SwitchB#

```

Sample Two: One Switch Supports DAI

This procedure shows how to configure DAI when Switch B shown in [Figure 39-2 on page 39-3](#) does not support DAI or DHCP snooping.

If switch Switch B does not support DAI or DHCP snooping, configuring Fast Ethernet port 6/3 on Switch A as trusted creates a security hole because both Switch A and Host 1 could be attacked by either Switch B or Host 2.

To prevent this possibility, you must configure Fast Ethernet port 6/3 on Switch A as untrusted. To permit ARP packets from Host 2, you must set up an ARP ACL and apply it to VLAN 1. If the IP address of Host 2 is not static, which would make it impossible to apply the ACL configuration on Switch A, you must separate Switch A from Switch B at Layer 3 and use a router to route packets between them.

To set up an ARP ACL on switch Switch A, follow these steps:

- Step 1** Configure the access list to permit the IP address 1.1.1.1 and the MAC address 0001.0001.0001, and verify the configuration:

```
SwitchA# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchA(config)# arp access-list H2
SwitchA(config-arp-nacl)# permit ip host 1.1.1.1 mac host 1.1.1
SwitchA(config-arp-nacl)# end
SwitchA# show arp access-list
ARP access list H2
    permit ip host 1.1.1.1 mac host 0001.0001.0001
```

- Step 2** Apply the ACL to VLAN 1, and verify the configuration:

```
SwitchA# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchA(config)# ip arp inspection filter H2 vlan 1
SwitchA(config)# end
SwitchA#

SwitchA# show ip arp inspection vlan 1

Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

Vlan    Configuration    Operation    ACL Match    Static ACL
----    -
    1    Enabled          Active      H2           No

Vlan    ACL Logging      DHCP Logging
----    -
    1    Deny            Deny

SwitchA#
```

- Step 3** Configure Fast Ethernet port 6/3 as untrusted, and verify the configuration:

```
SwitchA# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchA(config)# interface fastethernet 6/3
SwitchA(config-if)# no ip arp inspection trust
SwitchA(config-if)# end
Switch# show ip arp inspection interfaces fastethernet 6/3

Interface    Trust State    Rate (pps)
-----
Fa6/3        Untrusted      15

Switch#
```

When Host 2 sends 5 ARP requests through Fast Ethernet port 6/3 on Switch A and a “get” is permitted by Switch A, the statistics are updated appropriately:

```
Switch# show ip arp inspection statistics vlan 1
Vlan      Forwarded      Dropped      DHCP Drops      ACL Drops
-----
1          5                0              0                0
Vlan      DHCP Permits      ACL Permits      Source MAC Failures
-----
1          0                5                0
Vlan      Dest MAC Failures      IP Validation Failures
-----
1          0                      0
Switch#
```



Configuring Traffic Storm Control

This chapter describes how to configure the traffic storm control feature on the Catalyst 6500 series switches.



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- The WS-X6548-GE-TX, WS-X6548V-GE-TX, WS-X6148-GE-TX, and WS-X6148V-GE-TX switching modules do not support traffic storm control.

This chapter consists of these sections:

- [Understanding Traffic Storm Control, page 40-1](#)
- [Default Traffic Storm Control Configuration, page 40-3](#)
- [Enabling Traffic Storm Control, page 40-3](#)

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).

**Note**

- The switch supports multicast and unicast traffic storm control only on Gigabit Ethernet LAN ports.
- The switch supports broadcast traffic storm control on all LAN ports.
- Except for STP BPDUs, traffic storm control does not differentiate between control traffic and data traffic.
- Traffic storm control suppresses STP BPDUs on these modules:
 - WS-X6704-10GE
 - WS-X6748-SFP
 - WS-X6724-SFP
 - WS-X6748-GE-TX

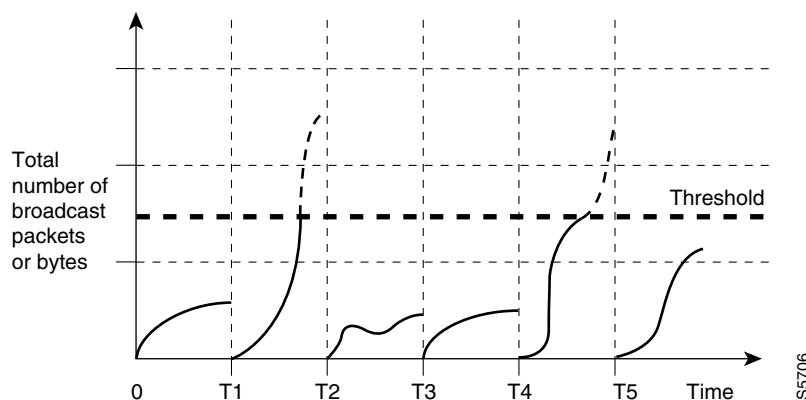
On the listed modules, do not configure traffic storm control on STP-protected ports that interconnect network devices.

Except on the listed modules, traffic storm control does not suppress STP BPDUs.

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 40-1 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 40-1 Broadcast Suppression



The traffic storm control threshold numbers and the time interval combination make the traffic storm control algorithm work with different levels of granularity. A higher threshold allows more packets to pass through.

Traffic storm control on the Catalyst 6500 series switches is implemented in hardware. The traffic storm control circuitry monitors packets passing from a LAN interface to the switching bus. Using the Individual/Group bit in the packet destination address, the traffic storm control circuitry determines if the packet is unicast or broadcast, keeps track of the current count of packets within the 1-second interval, and when a threshold is reached, filters out subsequent packets.

Because hardware traffic storm control uses a bandwidth-based method to measure traffic, the most significant implementation factor is setting the percentage of total available bandwidth that can be used by controlled traffic. Because packets do not arrive at uniform intervals, the 1-second interval during which controlled traffic activity is measured can affect the behavior of traffic storm control.

The following are examples of traffic storm control behavior:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast traffic until the end of the traffic storm control interval.
- If you enable broadcast and multicast traffic storm control, and the combined broadcast and multicast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.
- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within a 1-second traffic storm control interval, traffic storm control drops all broadcast and multicast traffic until the end of the traffic storm control interval.

Default Traffic Storm Control Configuration

Traffic storm control is disabled by default.

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]	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.
	Note The storm-control multicast command is supported only on Gigabit Ethernet interfaces.	
	Router(config-if)# no storm-control multicast level	Disables multicast traffic storm control on the interface.

	Command	Purpose
Step 4	Router(config-if)# storm-control unicast level <i>level[.level]</i>	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.
	Note The storm-control unicast command is supported only on Gigabit Ethernet interfaces.	
	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 <i>interface</i>	Verifies the configuration.

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

When configuring the traffic storm control level, note the following information:

- You can configure traffic storm control on an EtherChannel (a port channel interface).
- Do not configure traffic storm control on ports that are members of an EtherChannel. Configuring traffic storm control on ports that are configured as members of an EtherChannel puts the ports into a suspended state.
- Specify the level as a percentage of the total interface bandwidth:
 - The level can be from 0 to 100.
 - The optional fraction of a level can be from 0 to 99.
 - 100 percent means no traffic storm control.
 - 0.0 percent suppresses all traffic.



Note On these modules, a level value of 0.33 percent or less suppresses all traffic:

- WS-X6704-10GE
- WS-X6748-SFP
- WS-X6724-SFP
- WS-X6748-GE-TX

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 40-1](#).

Table 40-1 *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	Displays the total number of packets discarded for all three traffic storm control modes, on all interfaces or on the specified interface.
Router# show interfaces counters storm-control [module slot_number]	

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.



Unknown Unicast Flood Blocking

This chapter describes how to configure the unknown unicast flood blocking (UUFB) feature on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

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
```



Configuring PFC QoS

This chapter describes how to configure quality of service (QoS) as implemented on the Policy Feature Card (PFC) on the Catalyst 6500 series switches.



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- With a Supervisor Engine 720 and all versions of the PFC3, Release 12.2(18)SXE and later releases support QoS for IPv6 unicast and multicast traffic.
- Supervisor Engine 720 does not support Network-Based Application Recognition (NBAR).
- With a Supervisor Engine 2, PFC2, and MSFC2, you can configure NBAR on Layer 3 interfaces instead of PFC QoS:
 - The PFC2 provides hardware support for input ACLs on ports where you configure NBAR.
 - When PFC QoS is enabled, the traffic through ports where you configure NBAR passes through the ingress and egress queues and drop thresholds. When PFC QoS is enabled, the MSFC2 sets egress CoS equal to egress IP precedence in NBAR traffic.
 - After passing through an ingress queue, all traffic is processed in software on the MSFC2 on interfaces where you configure NBAR.
 - To configure NBAR, refer to this publication:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm>

This chapter contains these sections:

- [Understanding How PFC QoS Works, page 42-2](#)
- [PFC QoS Default Configuration, page 42-24](#)
- [PFC QoS Configuration Guidelines and Restrictions, page 42-37](#)
- [Configuring PFC QoS, page 42-41](#)

Understanding How PFC QoS Works

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 selects network traffic (both unicast and multicast), prioritizes it according to its relative importance, and uses congestion avoidance to provide priority-indexed treatment; QoS can also limit the bandwidth used by network traffic. QoS makes network performance more predictable and bandwidth utilization more effective.

**Note**

- On the Catalyst 6500 series switches, queue architecture and dequeuing algorithms such as Weighted-Round Robin (WRR) and Weighted Random Early Detection (WRED) are implemented with a fixed configuration in Application Specific Integrated Circuits (ASICs). The queue architecture cannot be reconfigured. For more information, see the [“Receive Queues” section on page 42-13](#) and the [“Transmit Queues” section on page 42-22](#).
- The QoS features implemented in the port ASICs (queue architecture and dequeuing algorithms) support IPv4 and IPv6 traffic.

These sections describe PFC QoS:

- [Hardware Supported by PFC QoS, page 42-2](#)
- [QoS Terminology, page 42-3](#)
- [PFC QoS Feature Flowcharts, page 42-5](#)
- [PFC QoS Feature Summary, page 42-11](#)
- [Ingress LAN Port Features, page 42-12](#)
- [PFC Marking and Policing, page 42-15](#)
- [Egress LAN Port Features, page 42-22](#)

Hardware Supported by PFC QoS

With Release 12.2(17b)SXA and later releases, PFC QoS supports both LAN ports and optical services module (OSM) ports:

- *LAN ports* are Ethernet ports on Ethernet switching modules, 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. With earlier releases, PFC QoS supports only LAN ports.
- *OSM ports* are the WAN ports on OSMs. The PFC provides ingress QoS for traffic from OSM ports. For more information, see the following sections:
 - [“Ingress OSM Port Features” section on page 42-11](#)
 - [“Egress OSM Port Features” section on page 42-12](#)
 - [“PFC Marking and Policing” section on page 42-15](#)
 - [“Ingress and Egress PFC QoS” section on page 42-20](#)
 - [“Configuring the Trust State of Ethernet LAN and OSM Ingress Ports” section on page 42-74](#)

- Refer to the following publication for information about additional OSM QoS features, including support for IPv6 traffic:
<http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/optical/122sx/qos.htm>
- The PFC does not provide QoS for FlexWAN module ports. Refer to the following publications for information about FlexWAN module QoS features:
 - *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm
 - *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.1:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm
 - *Class-Based Marking*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm>
 - *Traffic Policing*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtpoli.htm>
 - *Distributed Class-Based Weighted Fair Queueing and Distributed Weighted Random Early Detection*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtcbwred.htm>
 - *Distributed Low Latency Queueing*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtllqip.htm>
 - *Configuring Burst Size in Low Latency Queueing*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtcfgbst.htm>
 - *Distributed Traffic Shaping*:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/dtdts.htm>
 - For information about MPLS QoS:
http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/osm_inst/mpls.htm

QoS Terminology

This section defines some QoS terminology:

- *Packets* carry traffic at Layer 3.
- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *Classification* is the selection of traffic for QoS.
- *Marking*, according to RFC 2475, 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.
- *Scheduling* is the assignment of Layer 2 frames to a queue. PFC QoS assigns frames to a queue based on Layer 2 CoS values.

- *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.
- *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.
- *Labels* are prioritization values carried in Layer 3 packets and Layer 2 frames:
 - Layer 2 class of service (CoS) values, which range between zero for low priority and seven for high priority:
 Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p CoS value in the three least significant bits.
 Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value in the three most significant bits, which are called the User Priority bits.
 Other frame types cannot carry Layer 2 CoS values.

**Note**

On LAN ports configured as Layer 2 ISL trunks, all traffic is in ISL frames. On LAN ports configured as Layer 2 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.

- Layer 3 IP precedence values—The IP version 4 specification defines the three most significant bits of the 1-byte Type of Service (ToS) field as IP precedence. IP precedence values range between zero for low priority and seven for high priority.
- Layer 3 differentiated services code point (DSCP) values—The Internet Engineering Task Force (IETF) has defined the six most significant bits of the 1-byte IP ToS field as the DSCP. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63 (see the [“Configuring DSCP Value Maps”](#) section on page 42-70).

**Note**

Layer 3 IP packets can carry either an IP precedence value or a DSCP value. PFC QoS supports the use of either value, since DSCP values are backwards compatible with IP precedence values (see [Table 42-1](#)).

Table 42-1 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
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

PFC QoS Feature Flowcharts

Figure 42-1 show how traffic flows through the components that support PFC QoS features.

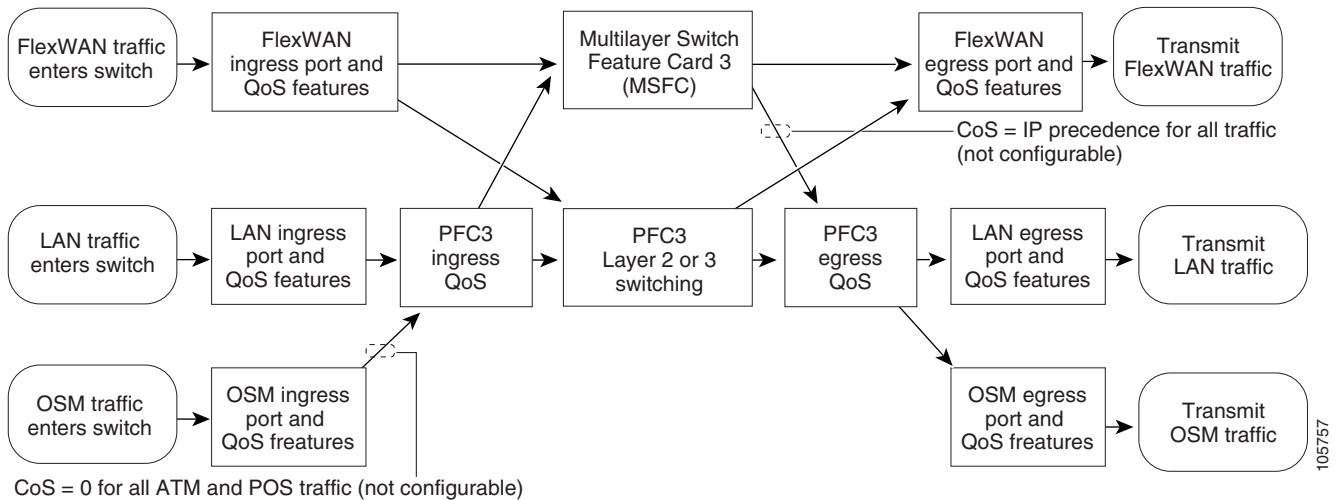
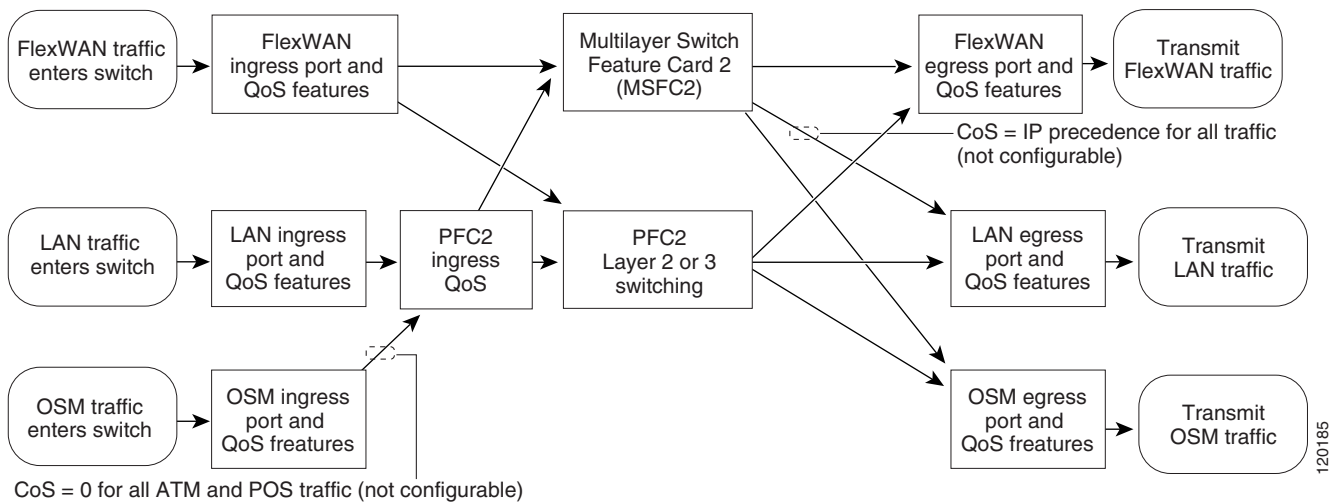
Figure 42-1 Traffic Flow Through PFC QoS Features with PFC3**Figure 42-2 Traffic Flow Through PFC QoS Features with PFC2**

Figure 42-3 through Figure 42-8 show how the PFC QoS features are implemented on the switch components.

Figure 42-3 Ingress LAN Port Layer 2 PFC QoS Features

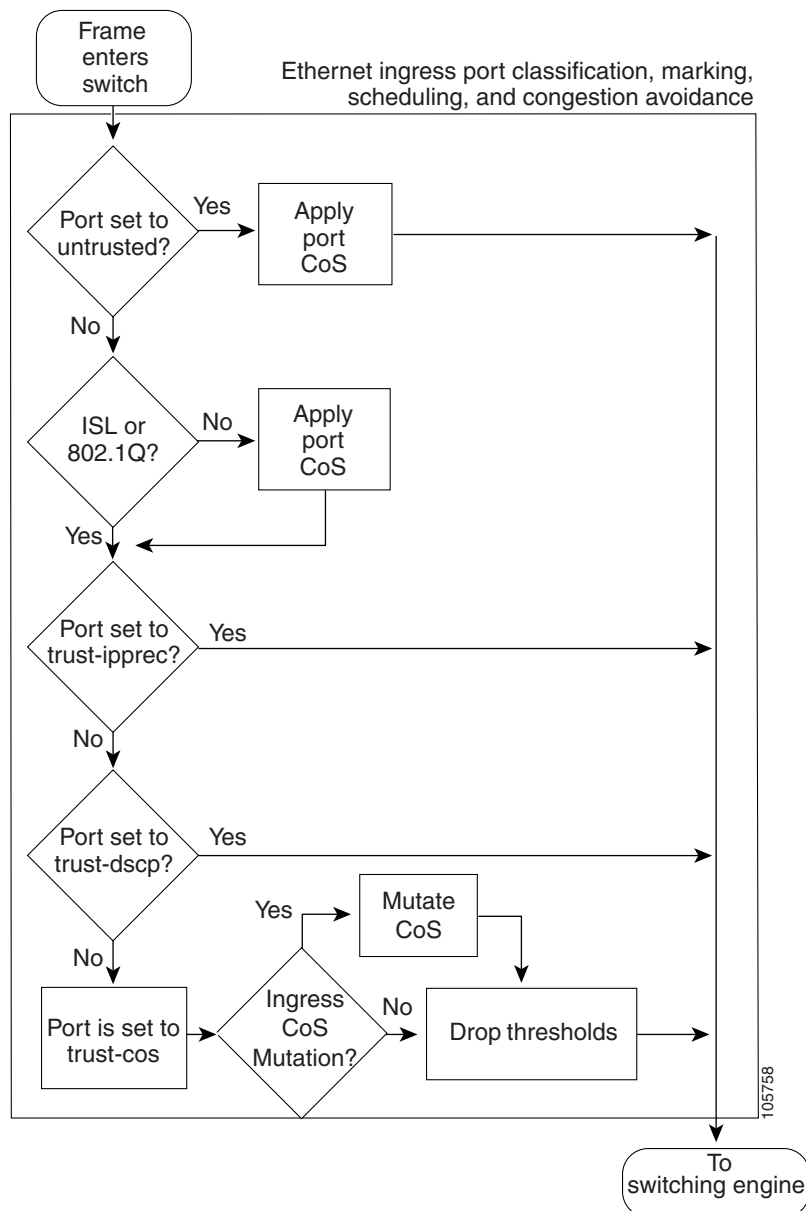


Figure 42-4 PFC Classification, Marking, and Policing

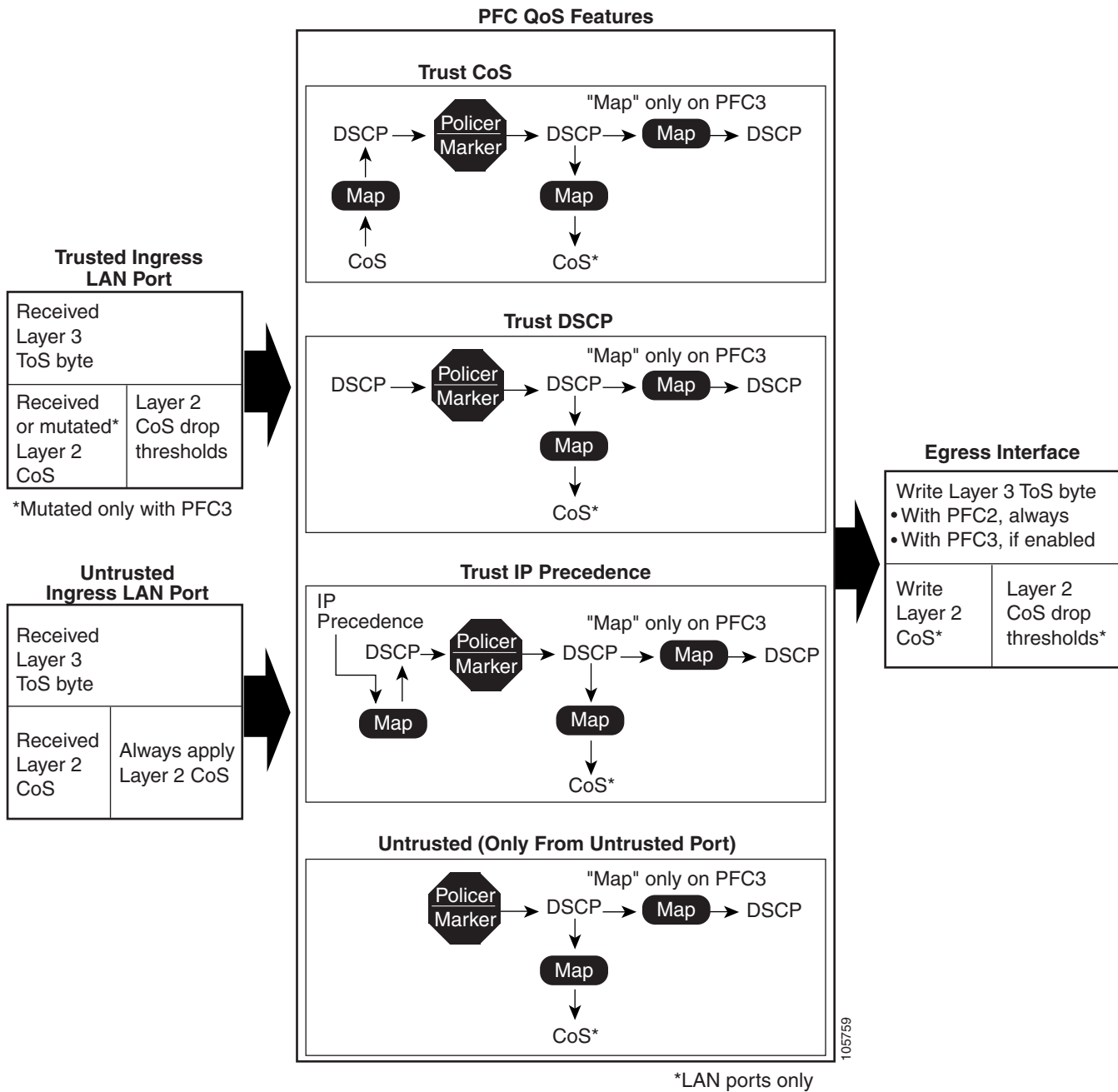


Figure 42-5 Marking with PFC3 and Multilayer Switch Feature Card 3

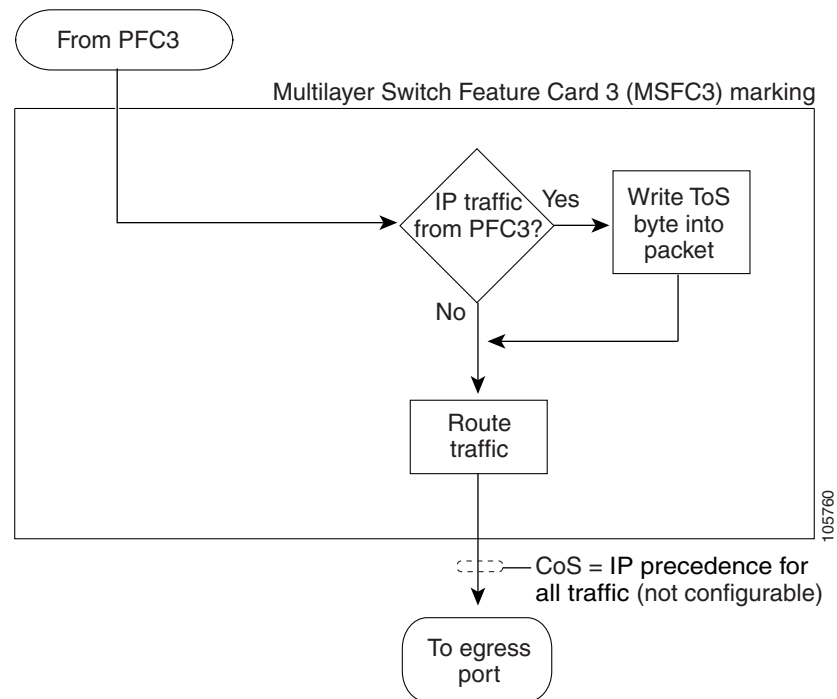


Figure 42-6 Marking with PFC2 and Multilayer Switch Feature Card 2

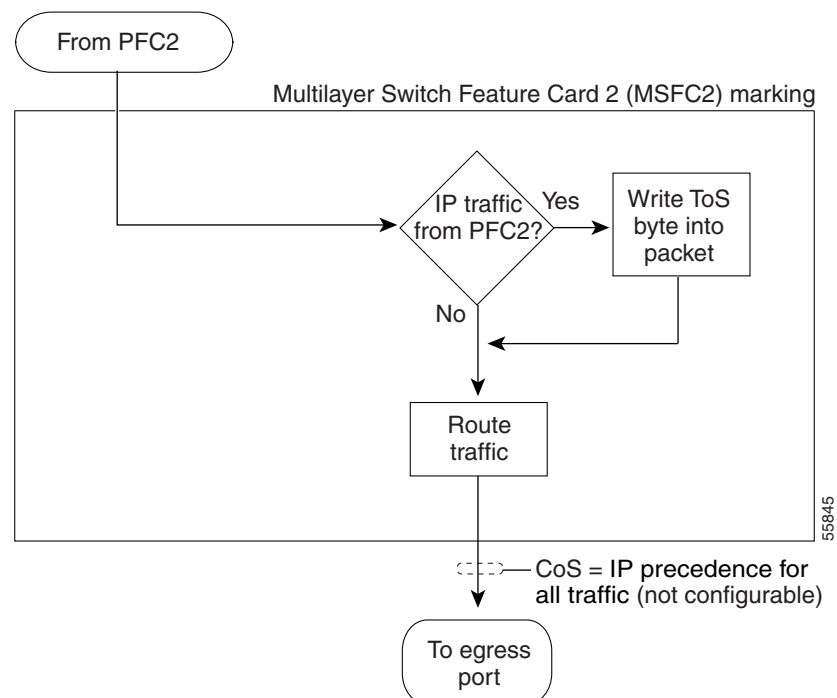


Figure 42-7 Egress WAN Port Marking

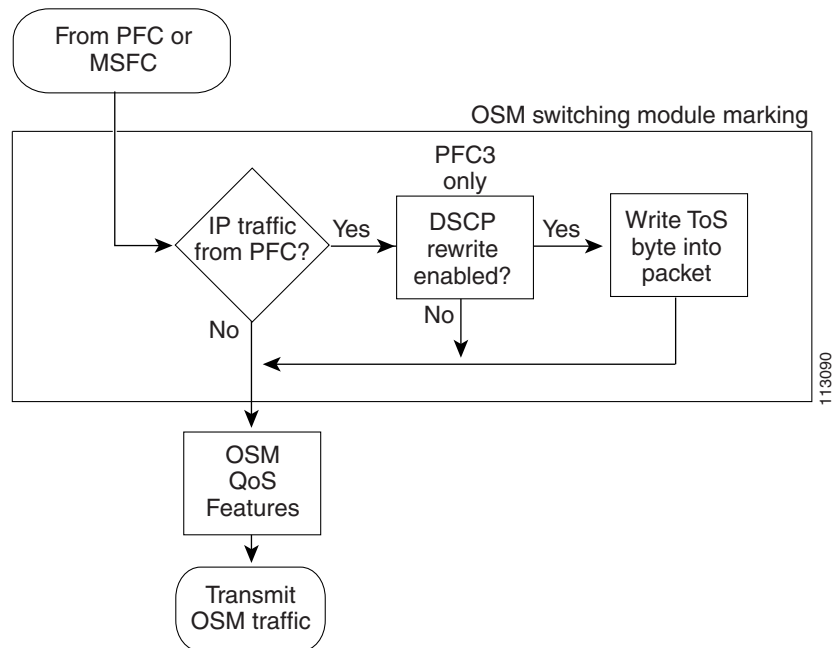
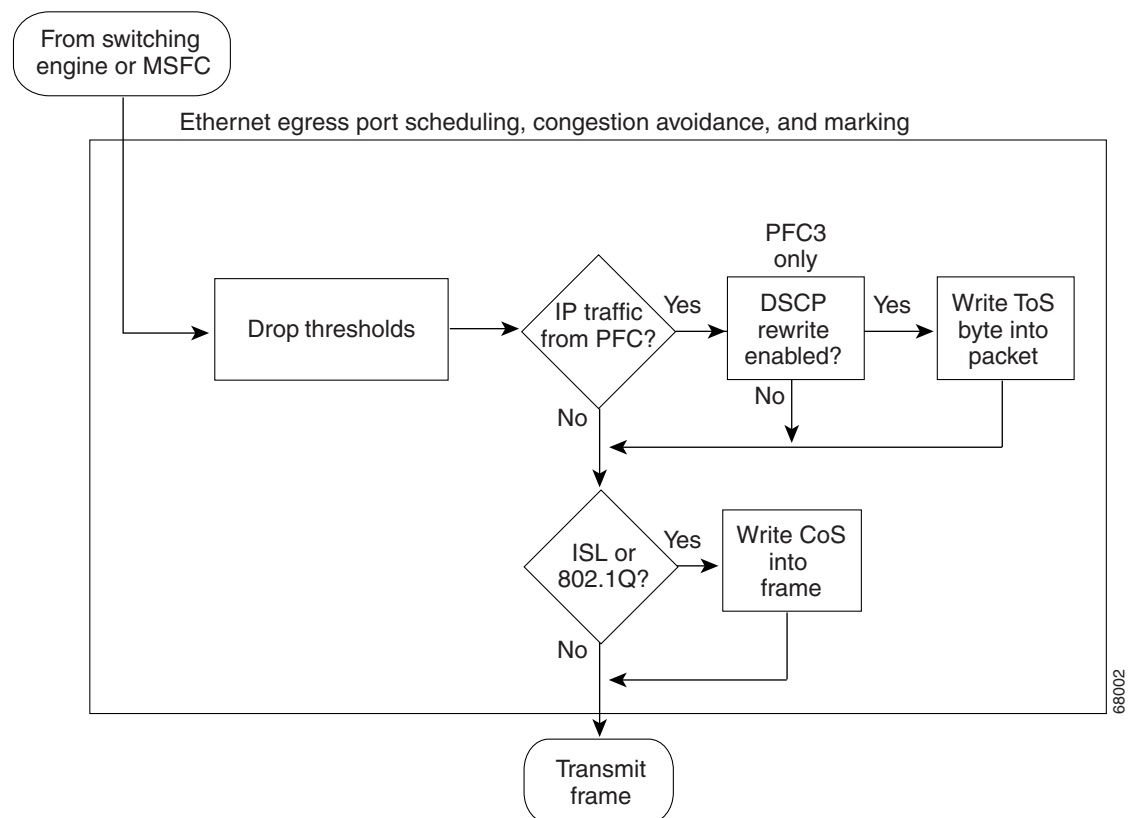


Figure 42-8 Egress LAN Port Scheduling, Congestion Avoidance, and Marking



PFC QoS Feature Summary

These sections summarize the PFC QoS features:

- [Ingress LAN Port Features, page 42-11](#)
- [Ingress OSM Port Features, page 42-11](#)
- [PFC QoS Features, page 42-11](#)
- [Egress LAN Port Features, page 42-12](#)
- [MSFC Features, page 42-12](#)

Ingress LAN Port Features

PFC QoS supports classification, marking, scheduling, and congestion avoidance using Layer 2 CoS values at ingress LAN ports. Classification, marking, scheduling, and congestion avoidance at ingress LAN ports do not use or set Layer 3 IP precedence or DSCP values. You can configure ingress LAN port trust states that can be used by the PFC to set Layer 3 IP precedence or DSCP values and the Layer 2 CoS value. See [Figure 42-3](#) and the “[Ingress LAN Port Features](#)” section on [page 42-12](#).

Ingress OSM Port Features

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 Layer 3 IP precedence or DSCP values and the Layer 2 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 because the default port CoS is not configurable on OSM ports.)

PFC QoS Features



Note

- With a PFC3, to preserve the received Layer 3 ToS byte in all traffic, you can globally disable egress ToS-byte rewrite with the **no mls qos rewrite ip dscp** command (see the “[Preserving the Received ToS Byte](#)” section on [page 42-42](#)).
- You can globally disable marking and policing with the **mls qos queueing-only** command (see the “[Enabling Queueing-Only Mode](#)” section on [page 42-43](#)).
- With DFCs installed, PFC2 does not support VLAN-based QoS: you cannot enter the **mls qos vlan-based** command or attach service policies to VLAN interfaces.

On the PFC, PFC QoS supports ingress classification, marking, and policing using policy maps. You can attach one policy map to each ingress port or EtherChannel and to VLAN interfaces for application to ingress traffic. With a PFC3, you can attach one policy map to each Layer 3 interface (either a LAN port configured as a Layer 3 interface or a VLAN interface) for application to egress traffic. 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. See the “[PFC Marking and Policing](#)” section on [page 42-15](#).

Egress LAN Port Features

PFC QoS supports egress LAN port scheduling and congestion avoidance using Layer 2 CoS values. Egress LAN port marking sets Layer 2 CoS values and Layer 3 DSCP values. See the [“Egress LAN Port Features” section on page 42-22](#).

Egress OSM Port Features

Ingress PFC QoS sets Layer 3 DSCP values that can be used by the OSM egress QoS features.

MSFC Features

PFC QoS marks IP traffic transmitted to the MSFC with rewritten Layer 3 DSCP values. CoS is equal to IP precedence in all traffic sent from the MSFC to egress ports.

**Note**

Traffic that is Layer 3 switched does not go through the MSFC and retains the CoS value assigned by the PFC.

Ingress LAN Port Features

These sections describe ingress LAN port PFC QoS features:

- [Ingress LAN Port Trust States, page 42-12](#)
- [Marking at Untrusted Ingress LAN Ports, page 42-13](#)
- [Marking at Trusted Ingress LAN Ports, page 42-13](#)
- [Receive Queues, page 42-13](#)
- [Scheduling and Congestion Avoidance, page 42-14](#)

Ingress LAN Port Trust States

The trust state of an ingress LAN port determines how the port marks, schedules, and classifies received Layer 2 frames, and whether or not congestion avoidance is implemented. You can configure the trust state of each ingress LAN port as follows:

- Untrusted (default)
- Trust IP precedence (not supported on **1q4t** LAN ports except Gigabit Ethernet)
- Trust DSCP (not supported on **1q4t** LAN ports except Gigabit Ethernet)
- Trust CoS (not supported on **1q4t** LAN ports except Gigabit Ethernet)

See the [“Configuring the Trust State of Ethernet LAN and OSM Ingress Ports” section on page 42-74](#). PFC QoS implements ingress LAN port congestion avoidance only on LAN ports configured to trust CoS.

**Note**

Ingress LAN port marking, scheduling, and congestion avoidance use Layer 2 CoS values and does not use or set Layer 3 IP precedence or DSCP values.

Marking at Untrusted Ingress LAN Ports

PFC QoS marks all frames received through untrusted ingress LAN ports with the ingress port CoS value (the default is zero). PFC QoS does not implement ingress port congestion avoidance on untrusted ingress LAN ports.



Note

- To use the ingress port CoS value applied to untrusted traffic as the basis of egress DSCP, configure a trust-CoS policy map that matches the ingress traffic.
- The ingress port CoS value is configurable for each ingress LAN port (see the [“Configuring the Ingress LAN Port CoS Value”](#) section on page 42-75).

Marking at Trusted Ingress LAN Ports

When an ISL frame enters the Catalyst 6500 series switch through a trusted ingress LAN port, PFC QoS accepts the three least significant bits in the User field as a CoS value. When an 802.1Q frame enters the switch through a trusted ingress LAN port, PFC QoS accepts the User Priority bits as a CoS value. PFC QoS marks all traffic received in untagged frames with the ingress port CoS value.



Note

- PFC QoS uses the received CoS value in tagged trusted traffic as the basis of egress DSCP, unless there is a policy map that changes the trust state of the traffic.
- PFC QoS uses the ingress port CoS value applied to untagged trusted traffic as the basis of egress DSCP, unless there is a policy map that changes the trust state of the traffic.
- The ingress port CoS value is configurable for each ingress LAN port (see the [“Configuring the Ingress LAN Port CoS Value”](#) section on page 42-75).

Receive Queues

To see the queue structure of a LAN port, enter the **show queueing interface {ethernet | fastethernet | gigabitethernet | tengigabitethernet} slot/port | include type** command. The command displays one of the following:

- **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 (usable only on Gigabit Ethernet ports).
- **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 eight thresholds, each configurable as either WRED-drop or tail-drop
 - One non-configurable (100 percent) tail-drop threshold

Strict-priority queues are queues that are serviced in preference to other queues. PFC QoS services traffic in a strict-priority queue before servicing the standard queue. When PFC QoS services the standard queue, after receiving a packet, it checks for traffic in the strict-priority queue. If PFC QoS 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.

Scheduling and Congestion Avoidance

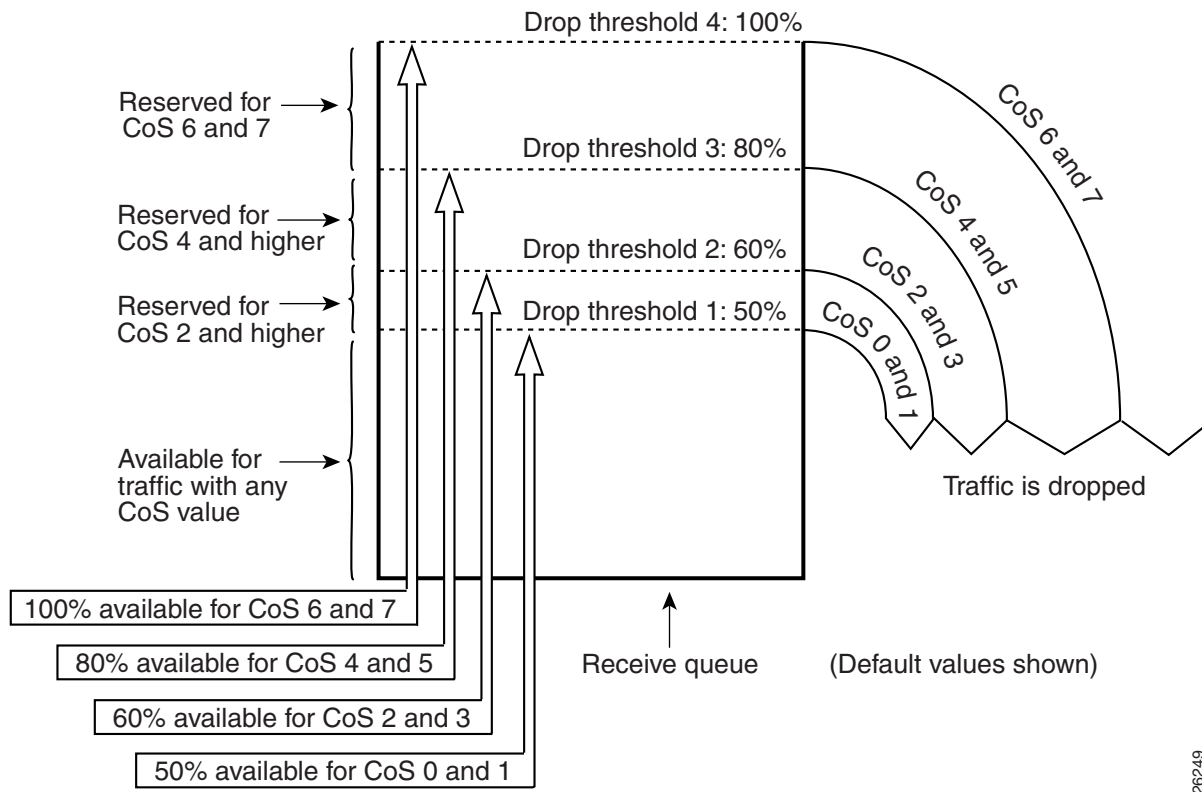
PFC QoS schedules traffic through the receive queues based on Layer 2 CoS values. In the default configuration, PFC QoS assigns all traffic with CoS 5 to the strict-priority queues (if present); PFC QoS assigns all other traffic to the standard queues. In the absence of a strict priority queue, PFC QoS assigns all traffic to the standard queues. See the [“PFC QoS Default Configuration” section on page 42-24](#) for information about the default CoS-to-queue and threshold mapping.

If an ingress LAN port is configured to trust CoS, PFC QoS implements Layer 2 CoS-value-based receive-queue drop thresholds to avoid congestion in received traffic.

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. See the [“Configuring Standard-Queue Drop Threshold Percentages” section on page 42-76](#) and the [“Configuring PFC QoS” section on page 42-41](#)). All LAN ports of the same type use the same drop-threshold configuration.

[Figure 42-9](#) illustrates the drop thresholds for a **1q4t** ingress LAN port. Drop thresholds in other configurations function similarly.

Figure 42-9 Receive Queue Drop Thresholds



PFC Marking and Policing



Note

- To mark untrusted traffic without policing, use the **set ip dscp** or **set ip precedence** policy map class commands (see the “[Configuring Policy Map Class Actions](#)” section on page 42-58).
- Filtering for PFC QoS can use Layer 2, 3, and 4 values. Marking uses Layer 2 CoS values and Layer 3 IP precedence or DSCP values.

These sections describe PFC marking and policing:

- [Internal DSCP Values, page 42-16](#)
- [Policy Maps, page 42-16](#)
- [Policers, page 42-17](#)
- [Ingress and Egress PFC QoS, page 42-20](#)
- [Egress DSCP and CoS Values, page 42-21](#)

Internal DSCP Values

During processing, PFC QoS represents the priority of all traffic (including non-IP traffic) with an internal DSCP value. PFC QoS derives the internal DSCP value from the following:

- For trust-cos traffic, from received or ingress port Layer 2 CoS values

**Note**

Traffic from an untrusted ingress LAN port has the ingress port CoS value and if traffic from an untrusted ingress Ethernet port matches a trust-cos policer, PFC QoS derives the internal DSCP value from the ingress port CoS value.

- For trust-ipprec traffic, from received IP precedence values
- For trust-dscp traffic, from received DSCP values
- For untrusted traffic, from ingress port CoS or configured DSCP values

The trust state of traffic is the trust state of the ingress LAN port unless set otherwise by a matching ACE.

**Note**

A **trust-cos** policer cannot restore received CoS in traffic from untrusted ingress LAN ports. Traffic from untrusted ingress LAN ports always has the ingress port CoS value.

PFC QoS uses configurable mapping tables to derive the internal 6-bit DSCP value from CoS or IP precedence, which are 3-bit values (see the [“Mapping Received CoS Values to Internal DSCP Values”](#) section on page 42-71 or the [“Mapping Received IP Precedence Values to Internal DSCP Values”](#) section on page 42-71).

Policy Maps

**Note**

- With a PFC3, to preserve the received Layer 3 ToS byte in all traffic, you can globally disable egress ToS-byte rewrite with the **no mls qos rewrite ip dscp** command (see the [“Preserving the Received ToS Byte”](#) section on page 42-42).
- You can globally disable marking and policing with the **mls qos queueing-only** command (see the [“Enabling Queueing-Only Mode”](#) section on page 42-43).

PFC QoS supports filtering, marking, and policing using policy maps (see the [“Configuring a Policy Map”](#) section on page 42-57). Each policy map can contain multiple policy-map classes. You can configure a separate policy-map class for each type of traffic.

Policy-map classes specify filtering with the following:

- Access control lists (ACLs):
 - Optional for IP
 - Required for MAC-Layer filtering
 - With a PFC2, required for IPX filtering
- Class-map **match** commands for Layer 3 IP precedence and DSCP values

Policy-map classes specify actions with the following:

- (Optional for ingress interfaces) Policy-map class **trust** commands. If specified, PFC QoS applies the policy-map class trust state to matched ingress traffic. Policy-map class trust states supersede ingress LAN port trust states.



Note If traffic matches a policy-map class that does not contain a **trust** command, the trust state remains as set on the ingress LAN port.

- (Optional) Aggregate and microflow policers, which can use bandwidth limits to either mark or drop both conforming and nonconforming traffic. See the [“PFC Marking and Policing” section on page 42-15](#).



Note With a PFC3, you cannot apply microflow policing to ARP traffic.

PFC QoS uses the trust state (set by the ingress LAN port configuration or by a **trust** policy-map class command) to select the Layer 2 and Layer 3 PFC QoS labels that the egress port writes into the packets and frames before it is transmitted:

- Trust IP precedence—Sets the internal DSCP value to a mapped value based on received IP precedence (see the [“Mapping Received IP Precedence Values to Internal DSCP Values” section on page 42-71](#)).
- Trust DSCP—Sets the internal DSCP value to the received DSCP value.
- Trust CoS—Sets the internal DSCP value to a mapped value based on received or port CoS. With trust CoS, note the following information:
 - Received CoS is overwritten with port CoS in traffic received through ports not configured to trust CoS.
 - Received CoS is preserved only in traffic received through ports configured to trust CoS.
 - Port CoS is applied to all traffic received in untagged frames, regardless of the port trust state.
 - For information about mapping, see the [“Mapping Received CoS Values to Internal DSCP Values” section on page 42-71](#).
- Untrusted—Sets the internal DSCP value to a configured DSCP value.



Note With the default values, PFC QoS applies DSCP zero to traffic from ingress LAN ports configured as untrusted.

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 (see the [“Configuring the Policy Map Class Trust State” section on page 42-59](#)).
- With Release 12.2(18)SX and later releases, you can apply aggregate and microflow policers to IPv6 traffic.

You can create policers that do the following:

- Mark traffic
- Limit bandwidth utilization and mark traffic

For more information, see the [“Creating Named Aggregate Policers”](#) section on page 42-46 and the [“Configuring Policy Map Class Actions”](#) section on page 42-58.

Policers can act on ingress traffic per-port or per-VLAN and on egress traffic per-VLAN.

With a PFC3, policing uses the Layer 2 frame size. With a PFC2, policing uses the Layer 3 packet 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 whether 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 (see the [“Configuring DSCP Markdown Values”](#) section on page 42-72). 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

- 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.

You can create two kinds of policers: *aggregate* and *microflow*:

- PFC QoS applies the bandwidth limits specified in an aggregate policer cumulatively to all flows in matched traffic. You can create up to 1023 aggregate policers. You can create two types of aggregate policer: named and per port. Both types can be attached to more than one port:
 - 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.



Note

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.

- PFC QoS applies the bandwidth limit specified in a microflow policer separately to each flow in matched traffic as follows:
 - You can create microflow policers with up to 63 different rate and burst parameter combinations.
 - You create microflow policers in a policy map class with the **police flow** command.
 - You can configure a microflow policer to use only source addresses, which applies the microflow policer to all traffic from a source address regardless of the destination addresses.
 - You can configure a microflow policer to use only destination addresses, which applies the microflow policer to all traffic to a destination address regardless of the source addresses.
 - For MAC-Layer microflow policing, PFC QoS considers MAC-Layer traffic with the same protocol and the same source and destination MAC-Layer addresses to be part of the same flow, including traffic with different EtherTypes. With a PFC3, you can configure MAC ACLs to filter IPX traffic (see the “[Configuring MAC ACLs](#)” section on page 42-50).
 - With a PFC2, you can configure IPX 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 MSFC3. To enable microflow policing of other traffic, including traffic in bridge groups, enter the **mls qos bridged** command (see the “[Enabling Microflow Policing of Bridged Traffic](#)” section on page 42-43).
 - With a PFC3, you cannot apply microflow policing to ARP traffic.



Note The flowmask requirements of microflow policing, NetFlow, and NetFlow data export (NDE) might conflict.

You can include both an aggregate policer and a microflow policer in each policy map class to police a flow based on both its own bandwidth utilization and on its bandwidth utilization combined with that of other flows.



Note If traffic is both aggregate and microflow policed, then the aggregate and microflow policers must both be in the same policy-map class and each must use the same **conform-action** and **exceed-action** keyword option: **drop**, **set-dscp-transmit**, **set-prec-transmit**, or **transmit**.

For example, you could create a microflow policer with a bandwidth limit suitable for individuals in a group and you could create a named aggregate policer with bandwidth limits suitable for the group as a whole. You could include both policers in policy map classes that match the group’s traffic. The combination would affect individual flows separately and the group aggregately.

For policy map classes that include both an aggregate and a microflow policer, PFC QoS responds to an out-of-profile status from either policer and, as specified by the policer, applies a new DSCP value or drops the packet. If both policers return an out-of-profile status, then if either policer specifies that the packet is to be dropped, it is dropped; otherwise PFC QoS applies a marked-down DSCP value.



Note To avoid inconsistent results, ensure that all traffic policed by the same aggregate policer has the same trust state.

Ingress and Egress PFC QoS

These sections describe ingress and egress PFC QoS:

- [Ingress PFC QoS, page 42-20](#)
- [Egress PFC QoS, page 42-20](#)
- [Egress ACL Support for Remarked DSCP, page 42-20](#)

Ingress PFC QoS

You can configure each ingress LAN port for either physical port-based PFC QoS (default) or VLAN-based PFC QoS (see the [“Enabling VLAN-Based PFC QoS on Layer 2 LAN Ports”](#) section on [page 42-44](#)) and attach a policy map to the selected interface (see the [“Attaching a Policy Map to an Interface”](#) section on [page 42-64](#)).

On ports configured for port-based PFC QoS, you can attach a policy map to the ingress LAN port as follows:

- On a nontrunk ingress LAN port configured for port-based PFC QoS, all traffic received through the port is classified, marked, and policed according 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 classified, marked, and policed according to the policy map attached to the port.

On a nontrunk ingress LAN port configured for VLAN-based PFC QoS, traffic received through the port is classified, marked, and policed according 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 classified, marked, and policed according to the policy map attached to the *traffic's* VLAN.

You can attach policy maps to OSM ports.

Egress PFC QoS

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 the policy to egress traffic (see [“Attaching a Policy Map to an Interface”](#) section on [page 42-64](#)).

**Note**

- Output policies do not support microflow policing.
- With a PFC3, you cannot apply microflow policing to ARP traffic.
- You cannot set a trust state in an output policy.

Egress ACL Support for Remarked DSCP

Egress ACL support for remarked DSCP enables IP precedence-based or DSCP-based egress QoS filtering to use any IP precedence or DSCP policing or marking changes made by ingress PFC QoS. Supervisor Engine 720 supports egress ACL support for remarked DSCP for IP unicast traffic with Release 12.2(18)SX and later releases (see the [“Enabling Egress ACL Support for Remarked DSCP”](#) section on [page 42-45](#)).

Without egress ACL support for remarked DSCP, IP precedence-based or DSCP-based egress QoS filtering uses received IP precedence or DSCP values; it does not use any IP precedence or DSCP changes made by ingress PFC QoS as the result of policing or marking.

The PFC3 provides egress PFC QoS only for Layer 3-switched and routed traffic on egress Layer 3 interfaces (either LAN ports configured as Layer 3 interfaces or VLAN interfaces).

You configure egress ACL support for remarked DSCP on ingress Layer 3 interfaces (either LAN ports configured as Layer 3 interfaces or VLAN interfaces).

On interfaces where egress ACL support for remarked DSCP is configured, the PFC3 processes each QoS-filtered IP packet twice:

- Once to apply ingress PFC QoS
- Once to apply egress PFC QoS

**Caution**

If the switch is operating in PFC3A mode with egress ACL support for remarked DSCP configured, when the PFC3 processes traffic to apply ingress PFC QoS, it applies ingress PFC QoS filtering and ingress PFC QoS, and incorrectly applies any egress QoS filtering and egress PFC QoS configured on the ingress interface, which results in unexpected behavior if QoS filtering is configured on an interface where egress ACL support for remarked DSCP is enabled. This problem does not occur in other PFC3 modes.

After packets have been processed by ingress PFC QoS and any policing or marking changes have been made, the packets are processed again on the ingress interface by any configured Layer 2 features (for example, VACLs) before being processed by egress PFC QoS.

**Note**

If a Layer 2 feature that is configured on an interface where egress ACL support for remarked DSCP is configured matches the ingress-QoS-modified IP precedence or DSCP value, the Layer 2 feature might redirect or drop the matched packets, preventing 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.

**Note**

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.

Egress DSCP and CoS Values

These sections describe egress DSCP and CoS values:

- [Egress CoS Values, page 42-21](#)
- [Egress DSCP Mutation with a PFC3, page 42-22](#)
- [Egress ToS Byte, page 42-22](#)

Egress CoS Values

For all egress traffic, PFC QoS uses a configurable mapping table to derive a CoS value from the internal DSCP value associated with traffic (see the [“Internal DSCP Values”](#) section on page 42-16 and the [“Mapping Internal DSCP Values to Egress CoS Values”](#) section on page 42-73).

PFC QoS sends the CoS value to the egress LAN ports for use in scheduling and to be written into ISL and 802.1Q frames (see the [“Egress LAN Port Features”](#) section on page 42-22).

**Note**

- PFC QoS derives the egress CoS value from the internal DSCP value.
- With a PFC3, if you configure egress DSCP mutation, PFC QoS does not derive the egress CoS value from the mutated DSCP value.
- The PFC2 does not support egress DSCP mutation.

Egress DSCP Mutation with a PFC3**Note**

The PFC2 does not support egress DSCP mutation.

With a PFC3, you can configure 15 egress DSCP mutation maps to mutate the internal DSCP value before it is written as the egress DSCP value (see the [“Configuring Egress DSCP Mutation on a PFC3” section on page 42-65](#)). You can attach egress DSCP mutation maps to any interface that PFC QoS supports.

Egress ToS Byte

For egress IP traffic, if received ToS-byte preservation is not enabled (see the [“Preserving the Received ToS Byte” section on page 42-42](#)), PFC QoS creates a ToS byte from the 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 2 least-significant bits from the received ToS byte.

**Note**

The internal or mutated DSCP value can mimic an IP precedence value (see [Table 42-1 on page 42-5](#)).

Egress LAN Port Features

These sections describe how PFC QoS schedules traffic through the transmit queues based on CoS values and uses CoS-value-based transmit-queue drop thresholds to avoid congestion in traffic transmitted from egress LAN ports:

- [Transmit Queues, page 42-22](#)
- [Scheduling and Congestion Avoidance, page 42-23](#)
- [Marking, page 42-23](#)

**Note**

Egress LAN port scheduling and congestion avoidance uses Layer 2 CoS values. Egress LAN port marking writes Layer 2 CoS values into trunk traffic and the Layer 3 ToS byte into all IP traffic.

Transmit Queues

To see the queue structure of an egress LAN port, enter the **show queueing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} **slot/port** | **include type** command.

The command displays one of the following:

- **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, each with one threshold configurable as either WRED-drop or tail-drop
 - One non-configurable (100 percent) tail-drop threshold
- **1p2q1t** indicates the following:
 - One strict-priority queue
 - Two standard queues, each with 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

For port types with a strict priority queue, the switch services traffic in the strict-priority transmit queue before servicing the standard queues. When the switch is servicing a standard queue, after transmitting a packet, it 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.

Scheduling and Congestion Avoidance

PFC QoS schedules traffic through the transmit queues based on Layer 2 CoS values. In the default configuration, PFC QoS assigns all traffic with CoS 5 to the strict-priority queue (if present); PFC QoS assigns all other traffic to standard queues. In the absence of a strict priority queue, PFC QoS assigns all traffic to the standard queues. See the [“PFC QoS Default Configuration” section on page 42-24](#) for information about the default CoS-to-queue and threshold mapping.

On some port types, you can configure each standard transmit queue to use both a non-configurable 100 percent tail-drop threshold and a configurable WRED-drop threshold (see the [“Configuring Standard-Queue Drop Threshold Percentages” section on page 42-76](#)).

Marking

When traffic is transmitted from the switch, PFC QoS writes the ToS byte into IP packets. On LAN ports, PFC QoS also writes the CoS value that was used for scheduling and congestion avoidance into ISL and 802.1Q frames (see the [“Egress CoS Values” section on page 42-21](#)).

PFC QoS Default Configuration

These sections describe the PFC QoS default configuration:

- [PFC QoS Global Settings, page 42-24](#)
- [Default Values with PFC QoS Enabled, page 42-25](#)
- [Default Values With PFC QoS Disabled, page 42-36](#)

PFC QoS Global Settings

The following global PFC QoS settings apply:

Feature	Default Value
PFC QoS global enable state	Disabled Note With PFC QoS enabled and all other PFC QoS parameters at default values, PFC QoS sets Layer 3 DSCP to zero and Layer 2 CoS to zero in all traffic transmitted from the switch.
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
CoS to DSCP map (DSCP set from 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
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

Feature	Default Value
DSCP to CoS map (CoS set from DSCP values):	DSCP 0–7 = CoS 0 DSCP 8–15 = CoS 1 DSCP 16–23 = CoS 2 DSCP 24–31 = CoS 3 DSCP 32–39 = CoS 4 DSCP 40–47 = CoS 5 DSCP 48–55 = CoS 6 DSCP 56–63 = CoS 7
Marked-down DSCP from DSCP map:	Marked-down DSCP value equals original DSCP value (no markdown)
Policers	None
Policy maps	None
Protocol-independent MAC ACL filtering	Disabled
VLAN-Based MAC ACL QoS filtering	Disabled

Default Values with PFC QoS Enabled

These sections list the default values that apply when PFC QoS is enabled:

- [Receive-Queue Size Percentages, page 42-25](#)
- [Transmit-Queue Size Percentages, page 42-26](#)
- [Bandwidth Allocation Ratios, page 42-26](#)
- [Default Drop-Threshold Percentages and CoS Value Mapping, page 42-26](#)



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%
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 Mapping

The following tables list the default drop-thresholds values and CoS mappings for different queue types:

- [1q4t Receive Queues, page 42-27](#)
- [1p1q4t Receive Queues, page 42-28](#)
- [1p1q0t Receive Queues, page 42-28](#)

- [1p1q8t Receive Queues, page 42-29](#)
- [1q8t Receive Queues, page 42-30](#)
- [2q8t Receive Queues, page 42-31](#)
- [8q8t Receive Queues, page 42-32](#)
- [2q2t Transmit Queues, page 42-32](#)
- [1p2q2t Transmit Queues, page 42-33](#)
- [1p3q8t Transmit Queues, page 42-34](#)
- [1p7q8t Transmit Queues, page 42-35](#)
- [1p3q1t Transmit Queues, page 42-36](#)
- [1p2q1t Transmit Queues, page 42-36](#)

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
Standard receive queues 2–7 (intermediate priorities)	Thresholds 5–8	CoS	None
		Tail-drop	Disabled; 100%
		WRED-drop	Enabled; 50% low, 100% high
Standard receive queue 8 (highest priority)	Thresholds 1–8	CoS	None
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
	Threshold 1	CoS	5
		Tail-drop	Enabled; 100%
		WRED-drop	Disabled; 100% low, 100% high
Standard receive queue 8 (highest priority)	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:

- With a Supervisor Engine 720 and all versions of the PFC3, Release 12.2(18)SXE and later releases support QoS for IPv6 unicast and multicast traffic.
- To display information about IPv6 PFC QoS, enter the **show mls qos ipv6** command.
- The QoS features implemented in the port ASICs (queue architecture and dequeuing algorithms) support IPv4 and IPv6 traffic.
- The PFC3 supports IPv6 named extended ACLs and named standard ACLs.
- The **match ip precedence** and **match ip dscp** commands only filter IPv4 traffic.
- In Release 12.2(18)SXE and later releases, the **match precedence** and **match dscp** commands filter IPv4 and IPv6 traffic.
- In Release 12.2(18)SXE and later releases, the **set ip dscp** and **set ip precedence** commands are saved in the configuration file as **set dscp** and **set precedence** commands.
- In Release 12.2(18)SXE and later releases, PFC QoS supports the **set dscp** and **set precedence** policy map class commands for IPv4 and IPv6 traffic.
- In Release 12.2(18)SXE and later releases, Supervisor Engine 720 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.
- In Release 12.2(18)SXE and later releases, you can apply aggregate and microflow policers to 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.
- With egress ACL support for remarked DSCP configured, the PFC3 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 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.)
- With a PFC3, you cannot apply microflow policing to ARP traffic.
- Supervisor Engine 2, PFC2 and MSFC2 support the **match protocol** class map command, which configures NBAR and sends all traffic on the Layer 3 interface, both ingress and egress, to be processed in software on the MSFC2. To configure NBAR, refer to this publication:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm>
- 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**.
- PFC QoS does not rewrite the payload ToS byte in tunnel traffic.
- The PFC2 does not support these PFC QoS features:
 - Egress policing
 - Egress DSCP mutation
 - Received ToS byte preservation with the **no mls qos rewrite ip dscp** command.
 - VLAN-based QoS with DFCs installed
- Supervisor Engine 2 does not support the modules that support ingress CoS mutation on IEEE 802.1Q tunnel ports.
- The PFC3 does not apply egress policing to traffic that is being bridged to the MSFC3.
- The PFC3 does not apply egress policing or egress DSCP mutation to multicast traffic from the MSFC3.
- With a PFC3, PFC QoS does not rewrite the ToS byte in bridged multicast traffic.
- You cannot configure PFC QoS features on tunnel interfaces.
- PFC QoS filters only by ACLs, dscp values, or IP precedence values.
- PFC QoS supports class maps that contain a *single* **match** command.

- PFC QoS does not support these class map commands:
 - **match cos**
 - **match classmap**
 - **match destination-address**
 - **match input-interface**
 - **match mpls**
 - **match qos-group**
 - **match source-address**
- With Release 12.2(18)SXE and later releases, PFC QoS supports the **match any** class map command.
- 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**
- PFC QoS does not support these policy map class commands:
 - **bandwidth**
 - **priority**
 - **queue-limit**
 - **random-detect**
 - **set qos-group**
 - **service-policy**
- PFC QoS supports the **set ip dscp** and **set ip precedence** policy map class commands (see the “Configuring Policy Map Class Actions” section on page 42-58).
- 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)

CIR and PIR Rate Value Range	Granularity
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.

- 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.

- 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

Configuring PFC QoS

These sections describe how to configure PFC QoS on the Catalyst 6500 series switches:

- [Enabling PFC QoS Globally, page 42-41](#)
- [Preserving the Received ToS Byte, page 42-42](#)
- [Enabling Queueing-Only Mode, page 42-43](#)
- [Enabling Microflow Policing of Bridged Traffic, page 42-43](#)
- [Enabling VLAN-Based PFC QoS on Layer 2 LAN Ports, page 42-44](#)
- [Enabling Egress ACL Support for Remarked DSCP, page 42-45](#)
- [Creating Named Aggregate Policers, page 42-46](#)
- [Configuring a PFC QoS Policy, page 42-48](#)
- [Configuring Egress DSCP Mutation on a PFC3, page 42-65](#)
- [Configuring Ingress CoS Mutation on IEEE 802.1Q Tunnel Ports, page 42-67](#)
- [Configuring DSCP Value Maps, page 42-70](#)
- [Configuring the Trust State of Ethernet LAN and OSM Ingress Ports, page 42-74](#)
- [Configuring the Ingress LAN Port CoS Value, page 42-75](#)
- [Configuring Standard-Queue Drop Threshold Percentages, page 42-76](#)
- [Mapping CoS Values to Drop Thresholds, page 42-82](#)
- [Allocating Bandwidth Between LAN-Port Standard Transmit Queues, page 42-86](#)
- [Setting the Receive-Queue Size Ratio on 1p1q0t and 1p1q8t Ingress LAN Ports, page 42-87](#)
- [Setting the LAN-Port Transmit-Queue Size Ratio, page 42-88](#)



Note

PFC QoS processes both unicast and multicast traffic.

Enabling PFC QoS Globally

To enable PFC QoS globally, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos	Enables PFC QoS globally on the switch.
	Router(config)# no mls qos	Disables PFC QoS globally on the switch.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos [ipv6]	Verifies the configuration.

This example shows how to enable PFC QoS globally:

```
Router# configure terminal
Router(config)# mls qos
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos
QoS is enabled globally
Microflow QoS is enabled globally

QoS global counters:
Total packets: 544393
IP shortcut packets: 1410
Packets dropped by policing: 0
IP packets with TOS changed by policing: 467
IP packets with COS changed by policing: 59998
Non-IP packets with COS changed by policing: 0

Router#
```

Preserving the Received ToS Byte



Note

- In addition to support for other IP traffic, the PFC3B and PFC3BXL support the **no mls qos rewrite ip dscp** command for MPLS traffic, traffic in IP in IP tunnels, and traffic in GRE tunnels.
- The PFC3A supports the **no mls qos rewrite ip dscp** command for all IP traffic except MPLS traffic, traffic in IP in IP tunnels, and traffic in GRE tunnels.

To preserve the received Layer 3 ToS byte, perform this task:

	Command	Purpose
Step 1	Router(config)# no mls qos rewrite ip dscp	Disables egress ToS-byte rewrite globally on the switch.
	Router(config)# mls qos rewrite ip dscp	Enables egress ToS-byte rewrite globally on the switch.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos [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#
```

Enabling Queueing-Only Mode

To enable queueing-only mode on the switch, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos queueing-only	Enables queueing-only mode on the switch.
	Router(config)# no mls qos queueing-only	Disables PFC QoS globally on the switch. Note You cannot disable queueing-only mode separately.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos [ipv6]	Verifies the configuration.

When you enable queueing-only mode, the switch does the following:

- Disables marking and policing globally
- Configures all ports to trust Layer 2 CoS



Note

The switch applies the port CoS value to untagged ingress traffic and to traffic that is received through ports that cannot be configured to trust CoS.

This example shows how to enable queueing-only mode:

```
Router# configure terminal
Router(config)# mls qos queueing-only
Router(config)# end
Router#
```

Enabling Microflow Policing of Bridged Traffic



Note

With a PFC2, to apply microflow policing to multicast traffic, you must enter the **mls qos bridged** command on the Layer 3 multicast ingress interfaces.

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

- With a PFC2, PFC QoS does not support VLAN-based QoS with DFCs installed.
- With a PFC3, PFC QoS supports VLAN-based QoS with DFC3s installed.
- With a PFC3, 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 (see the “[Ingress and Egress PFC QoS](#)” section on page 42-20). 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 (see the [“Egress ACL Support for Remarked DSCP” section on page 42-20](#)) 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] Router(config-if)# no platform ip features sequential [access-group IP_acl_name_or_number]	Enables egress ACL support for remarked DSCP on the ingress interface. 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 (see the “Policers” section on page 42-17), perform this task:

Command	Purpose
<pre>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-transmit¹ dscp_value set-prec-transmit¹ ip_precedence_value transmit}]] exceed-action {drop policed-dscp transmit}]] violate-action {drop policed-dscp transmit}]]</pre>	Creates a named aggregate policer.
<pre>Router(config)# no mls qos aggregate-policer policer_name</pre>	Deletes a named aggregate policer.

1. The **set-dscp-transmit** and **set-prec-transmit** keywords are only supported for IP traffic.



Note

Aggregate policers can be applied to ingress interfaces on multiple modules, but 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.

When creating a named aggregate policer, note the following information:

- In Release 12.2(18)SXE and later releases, you can apply aggregate policers to IPv6 traffic.
- With a PFC3, policing uses the Layer 2 frame size.
- With a PFC2, policing uses the Layer 3 packet size.
- See the “PFC QoS Configuration Guidelines and Restrictions” section on page 42-37 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—4 gigabits per second, entered as 4000000000
- 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—4 gigabits per second, entered as 4000000000
- (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 (see the “Policy Maps” section on page 42-16 and the “Configuring Policy Map Class Actions” section on page 42-58).
 - 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 (see the “Configuring DSCP Markdown Values” section on page 42-72).



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 (see the “Configuring DSCP Markdown Values” section on page 42-72).
 - For marking without policing, enter the **transmit** keyword to transmit all matched out-of-profile traffic.

**Note**

When you apply both ingress policing and egress policing to the same traffic, both the input policy and the output policy must either mark down traffic or drop traffic. PFC QoS does not support ingress markdown with egress drop or ingress drop with egress markdown.

This example shows how to create a named aggregate policer with a 1-Mbps rate limit and a 10-MB burst size that transmits conforming traffic and marks down out-of-profile traffic:

```
Router(config)# mls qos aggregate-policer aggr-1 1000000 10000000 conform-action transmit
exceed-action policed-dscp-transmit
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos aggregate-policer aggr-1
ag1 1000000 1000000 conform-action transmit exceed-action policed-dscp-transmit AgId=0
[pol4]
Router#
```

The output displays the following:

- The **AgId** parameter displays the hardware policer ID.
- The policy maps that use the policer are listed in the square brackets ([]).

Configuring a PFC QoS Policy

These sections describe PFC QoS policy configuration:

- [PFC QoS Policy Configuration Overview, page 42-48](#)
- [Configuring MAC ACLs, page 42-50](#)
- [Configuring ARP ACLs for QoS Filtering, page 42-53](#)
- [Configuring a Class Map, page 42-54](#)
- [Verifying Class Map Configuration, page 42-56](#)
- [Configuring a Policy Map, page 42-57](#)
- [Verifying Policy Map Configuration, page 42-63](#)
- [Attaching a Policy Map to an Interface, page 42-64](#)

**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
IPX (Supported only with PFC2)	Yes: 800 to 899	Yes: 900 to 999	Yes
MAC Layer	No	No	Yes
ARP	No	No	Yes

- The PFC3 supports IPv6 named extended ACLs and named standard ACLs in Release 12.2(18)SXE and later releases.
- The PFC3 supports ARP ACLs in Release 12.2(18)SXD and later releases.



Note

- The PFC2 on a Supervisor Engine 2 applies IP ACLs to ARP traffic.
- The PFC3 does not apply IP ACLs to ARP traffic.
- With a PFC3, you cannot apply microflow policing to ARP traffic.

- The PFC3 does not support IPX ACLs. With the PFC3, you can configure MAC ACLs (see the “Configuring MAC ACLs” section on page 42-50) to filter IPX traffic.
- With a PFC2, PFC QoS supports IPX ACLs that contain a *source-network* parameter and the optional *destination-network* and *destination-node* parameters. PFC QoS does not support IPX ACLs that contain other parameters (for example, *source-node*, *protocol*, *source-socket*, *destination-socket*, or *service-type*).
- With a PFC2 or PFC3, PFC QoS supports time-based Cisco IOS ACLs.
- Except for MAC ACLs (see the “Configuring MAC ACLs” section on page 42-50) and ARP ACLs (see the “Configuring ARP ACLs for QoS Filtering” section on page 42-53), refer to the *Cisco IOS Security Configuration Guide*, Release 12.2, “Traffic Filtering and Firewalls,” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/trafwl/index.htm
- See Chapter 34, “Configuring Network Security,” for additional information about ACLs on the Catalyst 6500 series switches.
- **class-map** (optional)—Enter the **class-map** command to define one or more traffic classes by specifying the criteria by which traffic is classified (see the “Configuring a Class Map” section on page 42-54).

- **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 42-50](#)
- [Enabling VLAN-Based MAC QoS Filtering, page 42-51](#)
- [Configuring MAC ACLs, page 42-52](#)



Note

You can use MAC ACLs with VLAN ACLs (VACLs; refer to [“Configuring VLAN ACLs”](#)).

Configuring Protocol-Independent MAC ACL Filtering

With Release 12.2(18)SXD and later releases, PFC3BXL and PFC3B modes support protocol-independent MAC ACL filtering. Protocol-independent MAC ACL filtering applies MAC ACLs to all ingress traffic types (for example, IPv4 traffic, IPv6 traffic, and MPLS traffic, in addition to MAC-layer traffic).

You can configure these interface types for protocol-independent MAC ACL filtering:

- VLAN interfaces without IP addresses
- Physical LAN ports configured to support EoMPLS
- Logical LAN subinterfaces configured to support EoMPLS

Ingress traffic permitted or denied by a MAC ACL on an interface configured for protocol-independent MAC ACL filtering is processed by egress interfaces as MAC-layer traffic. You cannot apply egress IP ACLs to traffic that was permitted or denied by a MAC ACL on an interface configured for protocol-independent MAC ACL filtering.

To configure protocol-independent MAC ACL filtering, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port[.subinterface]} {port-channel number[.subinterface]}}	Selects the interface to configure.
Step 2	Router(config-if)# mac packet-classify	Enables protocol-independent MAC ACL filtering on the interface.
	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.
- Protocol-independent MAC ACL filtering supports microflow policing when the permitted traffic is routed in software by the MSFC3.

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

In Release 12.2(18)SXD and later releases in PFC3BXL or PFC3B mode, you can globally enable or disable VLAN-based QoS filtering in MAC ACLs (see the [“Configuring MAC ACLs”](#) section on [page 42-52](#)). VLAN-based QoS filtering in MAC ACLs is disabled by default.

To enable VLAN-based QoS filtering in MAC ACLs, perform this task:

Command	Purpose
Router(config)# mac packet-classify use vlan	Enables VLAN-based QoS filtering in MAC ACLs.

To disable VLAN-based QoS filtering in MAC ACLs, perform this task:

Command	Purpose
Router(config)# no mac packet-classify use vlan	Disables VLAN-based QoS filtering in MAC ACLs.

Configuring MAC ACLs

You can configure named ACLs that filter IPX, DECnet, AppleTalk, VINES, or XNS traffic based on MAC addresses (IPX filtering with a MAC ACL is supported only with a PFC3).

In Release 12.2(17b)SXA and later releases in PFC3BXL or PFC3B mode, you can configure MAC ACLs that do VLAN-based filtering or CoS-based filtering or both.

In Release 12.2(18)SXD and later releases, 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 <i>list_name</i>	Configures a MAC ACL.
	Router(config)# no mac access-list extended <i>list_name</i>	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 <i>vlan_ID</i>] [cos <i>cos_value</i>]	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 <i>vlan_ID</i>] [cos <i>cos_value</i>]	Deletes an ACE from a MAC ACL.

When configuring an entry in a MAC-Layer ACL, note the following information:

- The PFC3 supports the **ipx-arpa** and **ipx-non-arpa** keywords.
- The PFC2 does not support the **ipx-arpa** and **ipx-non-arpa** keywords.
- The **vlan** and **cos** keywords are supported in PFC3BXL or PFC3B mode with Release 12.2(17b)SXA and later releases.
- The **vlan** and **cos** keywords are not supported in MAC ACLs used for VACL filtering.
- With Release 12.2(18)SXD and later releases, the **vlan** keyword for VLAN-based QoS filtering in MAC ACLs can be globally enabled or disabled and is disabled by default (see the [“Enabling VLAN-Based MAC QoS Filtering”](#) section on page 42-51).
- 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 PFC2 on a Supervisor Engine 2 applies IP ACLs to ARP traffic.
- The PFC3 does not apply IP ACLs to ARP traffic.
- With a PFC3, you cannot apply microflow policing to ARP traffic.

With a PFC3 and Release 12.2(18)SXD and later releases, 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 <i>list_name</i>	Configures an ARP ACL for QoS filtering.
	Router(config)# no arp access-list <i>list_name</i>	Deletes an ARP ACL.
Step 2	Router(config-arp-nacl)# { permit deny } { ip { any host <i>sender_ip</i> <i>sender_ip</i> <i>sender_ip_wildcardmask</i> } 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 <i>sender_ip</i> <i>sender_ip</i> <i>sender_ip_wildcardmask</i> } 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 PFC3. Any other ARP ACL syntax displayed by the CLI help when you enter a question mark (“?”) is not supported and cannot be used to filter ARP traffic for QoS.
- ACLs entries are scanned in the order you enter them. The first matching entry is used. To improve performance, place the most commonly used entries near the beginning of the ACL.
- An implicit **deny ip any mac any** entry exists at the end of an ACL unless you include an explicit **permit ip any mac any** entry at the end of the list.
- All new entries to an existing list are placed at the end of the list. You cannot add entries to the middle of a list.

This example shows how to create an ARP ACL named `arp_filtering` that only permits ARP traffic from IP address 1.1.1.1:

```
Router(config)# arp access-list arp_filtering
Router(config-arp-nacl)# permit ip host 1.1.1.1 mac any
```

Configuring a Class Map

These sections describe class map configuration:

- [Creating a Class Map, page 42-54](#)
- [Class Map Filtering Guidelines and Restrictions, page 42-55](#)
- [Configuring Filtering in a Class Map, page 42-55](#)

Creating a Class Map

To create a class map, perform this task:

Command	Purpose
Router(config)# class-map <i>class_name</i>	Creates a class map.
Router(config)# no class-map <i>class_name</i>	Deletes a class map.

Class Map Filtering Guidelines and Restrictions

When configuring class map filtering, follow these guidelines and restrictions:

- With Release 12.2(18)SXE and later releases, PFC QoS supports multiple match criteria in class maps configured with the **match-any** keywords.
- With releases earlier than Release 12.2(18)SXE, PFC QoS supports class maps that contain a single **match** command.
- With Release 12.2(18)SXE and later releases, Supervisor Engine 720 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.
- Release 12.2(18)SXE and later releases support 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 mpls**, **match qos-group**, and **match source-address** class map commands.
- Catalyst 6500 series switches do not detect the use of unsupported commands until you attach a policy map to an interface (see the “Attaching a Policy Map to an Interface” section on page 42-64).
- Supervisor Engine 2, PFC2, and MSFC2 support the **match protocol** class map command, which configures NBAR and sends all traffic on the Layer 3 interface, both ingress and egress, to be processed in software on the MSFC2. To configure NBAR, refer to this publication:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm>
- This chapter includes the following ACL documentation:
 - MAC ACLs (see the “Configuring MAC ACLs” section on page 42-50)
 - ARP ACLs (see the “Configuring ARP ACLs for QoS Filtering” section on page 42-53)
 Other ACLs are not documented in this publication. See the reference under **access-list** in the “PFC QoS Policy Configuration Overview” section on page 42-48.
- 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.

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 <i>acl_index_or_name</i>	(Optional) Configures the class map to filter using an ACL.
Router(config-cmap) # no match access-group name <i>acl_index_or_name</i>	Clears the ACL configuration from the class map.

Command	Purpose
Router (config-cmap)# match protocol ipv6	(Optional—for IPv6 traffic) Configures the class map to filter IPv6 traffic.
Router (config-cmap)# no match protocol ipv6	Clears IPv6 filtering.
Router (config-cmap)# match precedence <i>ipp_value1</i> [<i>ipp_value2</i> [<i>ipp_valueN</i>]]	(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 <i>ipp_value1</i> [<i>ipp_value2</i> [<i>ipp_valueN</i>]]	Clears configured IP precedence values from the class map.
Router (config-cmap)# match dscp <i>dscp_value1</i> [<i>dscp_value2</i> [<i>dscp_valueN</i>]]	(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 <i>dscp_value1</i> [<i>dscp_value2</i> [<i>dscp_valueN</i>]]	Clears configured DSCP values from the class map.
Router (config-cmap)# match ip precedence <i>ipp_value1</i> [<i>ipp_value2</i> [<i>ipp_valueN</i>]]	(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 <i>ipp_value1</i> [<i>ipp_value2</i> [<i>ipp_valueN</i>]]	Clears configured IP precedence values from the class map.
Router (config-cmap)# match ip dscp <i>dscp_value1</i> [<i>dscp_value2</i> [<i>dscp_valueN</i>]]	(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 <i>dscp_value1</i> [<i>dscp_value2</i> [<i>dscp_valueN</i>]]	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 <i>class_name</i>	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 42-57](#)
- [Policy Map Class Configuration Guidelines and Restrictions, page 42-57](#)
- [Creating a Policy Map Class and Configuring Filtering, page 42-58](#)
- [Configuring Policy Map Class Actions, page 42-58](#)

Creating a Policy Map

To create a policy map, perform this task:

Command	Purpose
Router(config)# policy-map <i>policy_name</i>	Creates a policy map.
Router(config)# no policy-map <i>policy_name</i>	Deletes the policy map.

Policy Map Class Configuration Guidelines and Restrictions

When configuring policy map classes, follow the guidelines and restrictions:

- Supervisor Engine 2, PFC2 and MSFC2 support the **class** *class_name* **protocol** policy map command, which configures NBAR and sends all traffic on the Layer 3 interface, both ingress and egress, to be processed in software on the MSFC2. To configure NBAR, refer to this publication: <http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/dtnbarad.htm>
- 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.
- With Release 12.2(18)SXE and later releases, 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 (see the [“Attaching a Policy Map to an Interface”](#) section on page 42-64).

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 <i>class_name</i>	Creates a policy map class and configures it to filter with a class map (see the “Creating a Class Map” section on page 42-54).
Router(config-pmap) # no class <i>class_name</i>	<p>Note PFC QoS supports class maps that contain a single match command.</p> <p>Clears use of the class map.</p>

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.
 - In Release 12.2(18)SXD and later releases and in Release 12.2(17d)SXB and later releases, 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.
 - In Release 12.2(18)SXE and later releases, the **set ip dscp** and **set ip precedence** commands are saved in the configuration file as **set dscp** and **set precedence** commands.
 - See the [“Configuring Policy Map Class Marking”](#) section on page 42-59).
- In Release 12.2(18)SXE and later releases, 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 (see the [“Configuring Policy Map Class Policing”](#) section on page 42-60).

These sections describe policy map class action configuration:

- [Configuring Policy Map Class Marking, page 42-59](#)
- [Configuring the Policy Map Class Trust State, page 42-59](#)
- [Configuring Policy Map Class Policing, page 42-60](#)

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 <i>dscp_value</i> precedence <i>ip_precedence_value</i> }	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 <i>dscp_value</i> precedence <i>ip_precedence_value</i> }	Clears the marking configuration.



Note Releases earlier than Release 12.2(18)SXE support the **set ip dscp** and **set ip precedence** policy map class commands.

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 internal DSCP value (see the “Internal DSCP Values” section on page 42-16).
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 (see the [“Mapping Received CoS Values to Internal DSCP Values”](#) section on page 42-71).
- With the **dscp** keyword, PFC QoS uses received DSCP.
- With the **ip-precedence** keyword, PFC QoS sets DSCP from received IP precedence (see the [“Mapping Received IP Precedence Values to Internal DSCP Values”](#) section on page 42-71).

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 (see the [“Attaching a Policy Map to an Interface”](#) section on page 42-64).

These sections describe configuration of policy map class policing:

- [Using a Named Aggregate Policer](#), page 42-60
- [Configuring a Per-Interface Policer](#), page 42-60



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 (see the [“Creating Named Aggregate Policers”](#) section on page 42-46), perform this task:

Command	Purpose
Router(config-pmap-c)# police aggregate <i>aggregate_name</i>	Configures the policy map class to use a previously defined named aggregate policer.
Router(config-pmap-c)# no police aggregate <i>aggregate_name</i>	Clears use of the named aggregate policer.

Configuring a Per-Interface Policer

To configure a per-interface policer (see the [“Policers”](#) section on page 42-17), perform this task:

Command	Purpose
Router(config-pmap-c)# police [flow [mask { src-only dest-only full-flow }] <i>bits_per_second</i> <i>normal_burst_bytes</i> [<i>maximum_burst_bytes</i>] [pir <i>peak_rate_bps</i>] [[[conform-action { drop set-dscp-transmit <i>dscp_value</i> set-prec-transmit <i>ip_precedence_value</i> 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 }] <i>bits_per_second</i> <i>normal_burst_bytes</i> [<i>maximum_burst_bytes</i>] [pir <i>peak_rate_bps</i>] [[[conform-action { drop set-dscp-transmit <i>dscp_value</i> set-prec-transmit <i>ip_precedence_value</i> transmit }] exceed-action { drop policed-dscp transmit }] violate-action { drop policed-dscp transmit }]	Deletes the per-interface policer from the policy-map class.

When configuring a per-interface policer, note the following information:

- With Release 12.2(18)SXE and later releases, you can apply aggregate and microflow policers to IPv6 traffic.
- With a PFC3, policing uses the Layer 2 frame size.
- With a PFC2, policing uses the Layer 3 packet size.
- See the “PFC QoS Configuration Guidelines and Restrictions” section on page 42-37 for information about rate and burst size granularity.
- You can enter the **flow** keyword to define a microflow policer (you cannot apply microflow policing to ARP traffic). When configuring a microflow policer, note the following information:
 - With a PFC3, 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. Release 12.2(17d)SXB and later releases support the **mask src-only** keywords for both IP traffic and MAC traffic. Releases earlier than Release 12.2(17d)SXB support the **mask src-only** keywords only for IP traffic.
 - With a PFC3, 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. Release 12.2(17d)SXB and later releases support the **mask dest-only** keywords for both IP traffic and MAC traffic. Releases earlier than Release 12.2(17d)SXB support the **mask dest-only** keywords only for IP 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.
 - With a PFC2, PFC QoS considers IPX traffic with same source network, destination network, and destination node to be part of the same flow, including traffic with different source nodes or sockets.
 - 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—4 gigabits per second, entered as 4000000000
- 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—4 gigabits per second, entered as 4000000000
- (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 (see the “Policy Maps” section on page 42-16 and the “Configuring Policy Map Class Actions” section on page 42-58).
 - 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 (see the “Configuring DSCP Markdown Values” section on page 42-72).



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 (see the “[Configuring DSCP Markdown Values](#)” section on page 42-72).

**Note**

- 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.
- With a PFC3, 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 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 <i>policy_name</i>	Verifies the configuration.

This example shows how to verify the configuration:

```
Router# show policy-map max-pol-ipp5
Policy Map max-pol-ipp5
  class ipp5

    class ipp5
      police flow 10000000 10000 conform-action set-prec-transmit 6 exceed-action
policed-dscp-transmit
      trust precedence
      police 2000000000 2000000 2000000 conform-action set-prec-transmit 6 exceed-action
policed-dscp-transmit

Router#
```

Attaching a Policy Map to an Interface

To attach a policy map to an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port[.subinterface]} {port-channel number[.subinterface]}}	Selects the interface to configure.
Step 2	Router(config-if)# service-policy [input output] policy_map_name Router(config-if)# no service-policy [input output] policy_map_name	Attaches a policy map to the interface. Removes the policy map from the interface.
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.
- With DFCs installed, PFC2 does not support VLAN-based QoS: you cannot enter the **mls qos vlan-based** command or attach service policies to VLAN interfaces.
- PFC QoS supports the **output** keyword only with a PFC3 and only on Layer 3 interfaces (either LAN ports configured as Layer 3 interfaces or VLAN interfaces). With a PFC3, 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.
- 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.
- See the [“Ingress and Egress PFC QoS” section on page 42-20](#) for more information about attaching policy maps to interfaces.

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 PFC3



Note

The PFC2 does not support egress DSCP mutation.

These sections describe how to configure egress DSCP mutation on a PFC3:

- [Configuring Named DSCP Mutation Maps, page 42-66](#)
- [Attaching an Egress DSCP Mutation Map to an Interface, page 42-67](#)

See the [“Egress DSCP Mutation with a PFC3” section on page 42-22](#) for information about how egress DSCP mutation works.

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 <i>map_name dscp1 [dscp2 [dscp3 [dscp4 [dscp5 [dscp6</i> <i>[dscp7 [dscp8]]]]]] to output_dscp</i>	Configures a named DSCP mutation map.
	Router(config)# no mls qos map dscp-mutation <i>map_name</i>	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



Note

Supervisor Engine 2 does not support the switching modules that support ingress CoS mutation.

Release 12.2(17b)SXA and later releases support ingress CoS mutation on IEEE 802.1Q tunnel ports configured to trust received CoS (see the [“Applying Ingress CoS Mutation Maps to IEEE 802.1Q Tunnel Ports”](#) section on page 42-69 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 (see [Chapter 18, “Configuring IEEE 802.1Q Tunneling”](#) and [“Configuring the Trust State of Ethernet LAN and OSM Ingress Ports”](#) section on page 42-74).

These sections describe how to configure ingress CoS mutation:

- [Ingress CoS Mutation Configuration Guidelines and Restrictions](#), page 42-68
- [Configuring Ingress CoS Mutation Maps](#), page 42-69
- [Applying Ingress CoS Mutation Maps to IEEE 802.1Q Tunnel Ports](#), page 42-69

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.
- Release 12.2(17b)SXA and later releases support ingress CoS mutation on WS-X6704-10GE, WS-X6748-SFP, WS-X6724-SFP, and WS-X6748-GE-TX switching modules.
- Ingress CoS mutation configuration applies to all ports in a port group. The port groups are:
 - WS-X6704-10GE—4 ports, 4 port groups, 1 port in each group
 - WS-X6748-SFP—48 ports, 4 port groups: ports 1–12, 13–24, 25–36, and 37–48
 - WS-X6724-SFP—24 ports, 2 port groups: ports 1–12 and 13–24
 - WS-X6748-GE-TX—48 ports, 4 port groups: ports 1–12, 13–24, 25–36, and 37–48
- To avoid ingress CoS mutation configuration failures, only create EtherChannels where all member ports support ingress CoS mutation or where no member ports support ingress CoS mutation. Do not create EtherChannels with mixed support for ingress CoS mutation.
- If you configure ingress CoS mutation on a port that is a member of an EtherChannel, the ingress CoS mutation is applied to the port-channel interface.
- You can configure ingress CoS mutation on port-channel interfaces.
- With ingress CoS mutation configured on a port-channel interface, the following occurs:
 - The ingress CoS mutation configuration is applied to the port groups of all member ports of the EtherChannel. If any member port cannot support ingress CoS mutation, the configuration fails.
 - If a port in the port group is a member of a second EtherChannel, the ingress CoS mutation configuration is applied to the second port-channel interface and to the port groups of all member ports of the second EtherChannel. If any member port of the second EtherChannel cannot support ingress CoS mutation, the configuration fails on the first EtherChannel. If the configuration originated on a nonmember port in a port group that has a member port of the first EtherChannel, the configuration fails on the nonmember port.
 - The ingress CoS mutation configuration propagates without limit through port groups, member ports, and port-channel interfaces, regardless of whether or not the ports are configured to trust CoS or are configured as IEEE 802.1Q tunnel ports.
- An EtherChannel where you want to configure ingress CoS mutation must not have member ports that are in port groups containing member ports of other EtherChannels that have member ports that do not support ingress CoS mutation. (This restriction extends without limit through all port-group-linked member ports and port-channel-interface-linked ports.)
- A port where you want to configure ingress CoS mutation must not be in a port group that has a member port of an EtherChannel that has members that do not support ingress CoS mutation. (This restriction extends without limit through all port-group-linked member ports and port-channel-interface-linked ports.)
- There can be only be one ingress CoS mutation configuration applied to all port-group-linked member ports and port-channel-interface-linked ports.

Configuring Ingress CoS Mutation Maps

To configure an ingress CoS mutation map, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos map cos-mutation <i>mutation_map_name</i> <i>mutated_cos1</i> <i>mutated_cos2</i> <i>mutated_cos3</i> <i>mutated_cos4</i> <i>mutated_cos5</i> <i>mutated_cos6</i> <i>mutated_cos7</i> <i>mutated_cos8</i>	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 <i>map_name</i>	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 <i>mutation_map_name</i>	Attaches an ingress CoS mutation map to the interface.
	Router(config-if)# no mls qos cos-mutation <i>mutation_map_name</i>	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 42-71](#)
- [Mapping Received IP Precedence Values to Internal DSCP Values, page 42-71](#)
- [Configuring DSCP Markdown Values, page 42-72](#)
- [Mapping Internal DSCP Values to Egress CoS Values, page 42-73](#)

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 (see the “[Internal DSCP Values](#)” section on page 42-16), 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 (see the “[Internal DSCP Values](#)” section on page 42-16), 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 (see the [“Policers”](#) section on page 42-17), perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos map policed-dscp { normal-burst max-burst } <i>dscp1</i> [<i>dscp2</i> [<i>dscp3</i> [<i>dscp4</i> [<i>dscp5</i> [<i>dscp6</i> [<i>dscp7</i> [<i>dscp8</i>]]]]]]] to <i>markdown_dscp</i>	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 (see the [“Mapping Internal DSCP Values to Egress CoS Values”](#) section on page 42-73).

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 (see the [“Internal DSCP Values”](#) section on page 42-16 and the [“Egress LAN Port Features”](#) section on page 42-22), perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos map dscp-cos <i>dscp1</i> [<i>dscp2</i> [<i>dscp3</i> [<i>dscp4</i> [<i>dscp5</i> [<i>dscp6</i> [<i>dscp7</i> [<i>dscp8</i>]]]]]] to <i>cos_value</i>	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 00 01
  1 :  01 01 01 01 01 01 00 02 02 02 02
  2 :  02 02 02 02 00 03 03 03 03 03 03
  3 :  03 03 00 04 04 04 04 04 04 04 04
  4 :  00 05 05 05 05 05 05 05 00 06 06
  5 :  06 06 06 06 00 06 07 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 except non-Gigabit Ethernet **1q4t/2q2t** ports (see the [“Ingress LAN Port Features” section on page 42-12](#)). You can configure the ingress port trust state on OSM ports (see the [“Ingress OSM Port Features” section on page 42-11](#)).

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] Router(config-if)# no mls qos trust	Configures the trust state of the port. 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**.
- 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.
- With Release 12.2(17b)SXA and later releases, you can configure IEEE 802.1Q 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 42-67).
- Use the **no mls qos trust** command to set the port state to untrusted.

This example shows how to configure Gigabit Ethernet port 1/1 with the **trust cos** keywords:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# mls qos trust cos
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | include trust
Trust state: trust COS
Router#
```

Configuring the Ingress LAN Port CoS Value



Note

Whether or not PFC QoS uses the CoS value applied with the **mls qos cos** command depends on the trust state of the port and the trust state of the traffic received through the port. The **mls qos cos** command does not configure the trust state of the port or the trust state of the traffic received through the port.

To use the CoS value applied with the **mls qos cos** command as the basis of internal DSCP (see the “[Internal DSCP Values](#)” section on page 42-16):

- 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.

	Command	Purpose
Step 2	Router(config-if)# mls qos cos <i>port_cos</i>	Configures the ingress LAN port CoS value.
	Router(config-if)# no mls qos cos <i>port_cos</i>	Reverts to the default port CoS value.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface { ethernet fastethernet gigabitethernet } <i>slot/port</i>	Verifies the configuration.

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

This example shows how to configure the CoS value 5 on Fast Ethernet port 5/24 and verify the configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/24
Router(config-if)# mls qos cos 5
Router(config-if)# end
Router# show queueing interface fastethernet 5/24 | include Default COS
      Default COS is 5
Router#
```

Configuring Standard-Queue Drop Threshold Percentages

These sections describe configuring standard-queue drop threshold percentages:

- [Configuring a Tail-Drop Receive Queue, page 42-77](#)
- [Configuring a WRED-Drop Transmit Queue, page 42-78](#)
- [Configuring a WRED-Drop and Tail-Drop Receive Queue, page 42-78](#)
- [Configuring a WRED-Drop and Tail-Drop Transmit Queue, page 42-79](#)
- [Configuring 1q4t/2q2t Tail-Drop Threshold Percentages, page 42-80](#)



Note

- Enter the **show queueing interface** {**ethernet** | **fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* | **include type** command to see the queue structure of a port (see the “Receive Queues” section on page 42-13 and the “Transmit Queues” section on page 42-22).
- **1p1q0t** ports have no configurable thresholds.
- **1p3q1t** (transmit), **1p2q1t** (transmit), and **1p1q8t** (receive) ports also have nonconfigurable tail-drop thresholds (see the “Mapping CoS Values to Standard Transmit-Queue Thresholds” section on page 42-83).

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 } <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# rcv-queue threshold <i>queue_id</i> <i>thr1% thr2% thr3% thr4% {thr5% thr6% thr7% thr8%}</i>	Configures the receive-queue tail-drop threshold percentages.
	Router(config-if)# no rcv-queue threshold [<i>queue_id</i>]	Reverts to the default receive-queue tail-drop threshold percentages.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface { fastethernet gigabitethernet } <i>slot/port</i>	Verifies the configuration.

This example shows how to configure the receive-queue drop thresholds for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# rcv-queue threshold 1 60 75 85 100
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | begin Receive queues
Receive queues [type = 1p1q4t]:
  Queue Id      Scheduling  Num of thresholds
  -----
      1          Standard      4
      2          Priority       1

Trust state: trust COS

queue tail-drop-thresholds
-----
  1          60[1] 75[2] 85[3] 100[4]
<...Output Truncated...>
Router#
```

Configuring a WRED-Drop Transmit Queue

These port types have only WRED-drop thresholds in their transmit queues:

- **1p2q2t** (transmit)
- **1p2q1t** (transmit)

	Command	Purpose
Step 1	Router(config)# interface type ¹ slot/port	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue random-detect min-threshold queue_id thr1% [thr2%] Router(config-if)# no wrr-queue random-detect min-threshold [queue_id]	Configures the low WRED-drop thresholds. Reverts to the default low WRED-drop thresholds.
Step 3	Router(config-if)# wrr-queue random-detect max-threshold queue_id thr1% [thr2%] Router(config-if)# no wrr-queue random-detect max-threshold [queue_id]	Configures the high WRED-drop thresholds. Reverts to the default high WRED-drop thresholds.
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show queueing interface type ¹ slot/port	Verifies the configuration.

1. type = fastethernet, gigabitethernet, or tengigabitethernet

Configuring a WRED-Drop and Tail-Drop Receive Queue

These port types have both WRED-drop and tail-drop thresholds in their receive queues:

- **8q8t** (receive)
- **1p1q8t** (receive)

To configure the drop thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# rcv-queue threshold <i>queue_id</i> <i>thr1% thr2% thr3% thr4% thr5% thr6% thr7% thr8%</i> Router(config-if)# no rcv-queue threshold [<i>queue_id</i>]	Configures the tail-drop thresholds. Reverts to the default tail-drop thresholds.
Step 3	Router(config-if)# rcv-queue random-detect min-threshold <i>queue_id</i> <i>thr1% thr2% thr3% thr4% thr5% thr6% thr7% thr8%</i> Router(config-if)# no rcv-queue random-detect min-threshold [<i>queue_id</i>]	Configures the low WRED-drop thresholds. Reverts to the default low WRED-drop thresholds.
Step 4	Router(config-if)# rcv-queue random-detect max-threshold <i>queue_id</i> <i>thr1% thr2% thr3% thr4% thr5% thr6% thr7% thr8%</i> Router(config-if)# no rcv-queue random-detect max-threshold [<i>queue_id</i>]	Configures the high WRED-drop thresholds. Reverts to the default high WRED-drop thresholds.
Step 5	Router(config-if)# rcv-queue random-detect <i>queue_id</i> Router(config-if)# no rcv-queue random-detect [<i>queue_id</i>]	Enables WRED-drop thresholds. Enables tail-drop thresholds.
Step 6	Router(config-if)# end	Exits configuration mode.
Step 7	Router# show queueing interface <i>type</i> ¹ <i>slot/port</i>	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 <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue threshold <i>queue_id</i> <i>thr1% [thr2% thr3% thr4% thr5% thr6% thr7% thr8%]</i> Router(config-if)# no wrr-queue threshold [<i>queue_id</i>]	Configures the tail-drop thresholds. Reverts to the default tail-drop thresholds.
Step 3	Router(config-if)# wrr-queue random-detect min-threshold <i>queue_id</i> <i>thr1% [thr2% thr3% thr4% thr5% thr6% thr7% thr8%]</i> Router(config-if)# no wrr-queue random-detect min-threshold [<i>queue_id</i>]	Configures the low WRED-drop thresholds. Reverts to the default low WRED-drop thresholds.

	Command	Purpose
Step 4	Router(config-if)# wrr-queue random-detect max-threshold <i>queue_id</i> <i>thr1%</i> [<i>thr2% thr3% thr4% thr5% thr6% thr7% thr8%</i>]	Configures the high WRED-drop thresholds.
	Router(config-if)# no wrr-queue random-detect max-threshold [<i>queue_id</i>]	Reverts to the default high WRED-drop thresholds.
Step 5	Router(config-if)# wrr-queue random-detect <i>queue_id</i>	Enables WRED-drop thresholds.
	Router(config-if)# no wrr-queue random-detect [<i>queue_id</i>]	Enables tail-drop thresholds.
Step 6	Router(config-if)# end	Exits configuration mode.
Step 7	Router# show queueing interface <i>type</i> ¹ <i>slot/port</i>	Verifies the configuration.

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

This example shows how to configure the low-priority transmit queue high-WRED-drop thresholds for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# wrr-queue random-detect max-threshold 1 70 70
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1 | begin Transmit queues
Transmit queues [type = lp2q2t]:
  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 } <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue threshold <i>queue_id</i> <i>thr1% thr2%</i>	Configures the receive- and transmit-queue tail-drop thresholds.
	Router(config-if)# no wrr-queue threshold [<i>queue_id</i>]	Reverts to the default receive- and transmit-queue tail-drop thresholds.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface { ethernet fastethernet gigabitethernet } <i>slot/port</i>	Verifies the configuration.

When configuring the receive- and transmit-queue tail-drop thresholds, note the following information:

- You must use the transmit queue and threshold numbers.
- The *queue_id* is 1 for the standard low-priority queue and 2 for the standard high-priority queue.
- The percentages range from 1 to 100. A value of 10 indicates a threshold when the buffer is 10-percent full.
- Always set threshold 2 to 100 percent.
- Ethernet and Fast Ethernet **1q4t** ports do not support receive-queue tail-drop thresholds.

This example shows how to configure receive queue 1/threshold 1 and transmit queue 1/threshold 1 for Gigabit Ethernet port 2/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 2/1
Router(config-if)# wrr-queue threshold 1 60 100
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 2/1
Transmit queues [type = 2q2t]:
```

```
<...Output Truncated...>
```

```
queue tail-drop-thresholds
-----
1      60[1] 100[2]
2      40[1] 100[2]
```

```
<...Output Truncated...>
```

```
Receive queues [type = 1q4t]:
```

```
<...Output Truncated...>
```

```
queue tail-drop-thresholds
-----
1      60[1] 100[2] 40[3] 100[4]
```

```
<...Output Truncated...>
```

```
Router#
```

Mapping CoS Values to Drop Thresholds

These sections describe mapping CoS values to drop thresholds:



Note

Enter the **show queueing interface {ethernet | fastethernet | gigabitethernet | tengigabitethernet} slot/port | include type** command to see the queue structure of a port.

These sections describe how to map CoS values:

- [Mapping CoS Values to Standard Receive-Queue Thresholds, page 42-82](#)
- [Mapping CoS Values to Standard Transmit-Queue Thresholds, page 42-83](#)
- [Mapping CoS Values to Strict-Priority Queues, page 42-84](#)
- [Mapping CoS Values to Tail-Drop Thresholds on 1q4t/2q2t LAN Ports, page 42-85](#)

When CoS values to thresholds, note the following information:

- 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.
- 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)

See the “[Configuring Standard-Queue Drop Threshold Percentages](#)” section on page 42-76 for more information about configuring thresholds as either tail-drop or WRED-drop.

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 { fastethernet gigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# rcv-queue cos-map <i>queue_# threshold_# cos1 [cos2 [cos3 [cos4 [cos5 [cos6 [cos7 [cos8]]]]]]]</i> Router(config-if)# no rcv-queue cos-map	Maps CoS values to the standard receive queue thresholds. Reverts to the default mapping.

	Command	Purpose
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface {fastethernet gigabitethernet} slot/port	Verifies the configuration.

This example shows how to map the CoS values 0 and 1 to threshold 1 in the standard receive queue for Gigabit Ethernet port 1/1:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/1
Router(config-if)# rcv-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/1
<...Output Truncated...>
  queue thresh cos-map
  -----
    1      1      0 1
    1      2      2 3
    1      3      4 5
    1      4      6 7
<...Output Truncated...>
Router#
```

Mapping CoS Values to Standard Transmit-Queue Thresholds

To map CoS values to standard transmit-queue thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {fastethernet gigabitethernet} 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]]]]]] Router(config-if)# no wrr-queue cos-map	Maps CoS values to a standard transmit-queue threshold. Reverts to the default mapping.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface {fastethernet gigabitethernet} slot/port	Verifies the configuration.

This example shows how to map the CoS values 0 and 1 to standard transmit queue 1/threshold 1 for Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# wrr-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 5/36 | begin queue thresh cos-map
queue thresh cos-map
-----
1      1      0 1
1      2      2 3
2      1      4 5
2      2      6 7
<...Output Truncated...>
Router#
```

Mapping CoS Values to Strict-Priority Queues

To map CoS values to the receive and transmit strict-priority queues, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# priority-queue cos-map <i>queue_#</i> <i>cos1</i> [<i>cos2</i> [<i>cos3</i> [<i>cos4</i> [<i>cos5</i> [<i>cos6</i> [<i>cos7</i> [<i>cos8</i>]]]]]]] Router(config-if)# no priority-queue cos-map	Maps CoS values to the receive and transmit strict-priority queues. Reverts to the default mapping.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface <i>type</i> ¹ <i>slot/port</i>	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 queueing interface gigabitethernet 1/1
<...Output Truncated...>
Transmit queues [type = 1p2q2t]:
<...Output Truncated...>
  queue thresh cos-map
-----
  1      1      0 1
  1      2      2 3
  2      1      4
  2      2      6
  3      1      5 7

  Receive queues [type = 1p1q4t]:
<...Output Truncated...>
  queue thresh cos-map
-----
  1      1      0 1
  1      2      2 3
  1      3      4
  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 queueing interface {ethernet | fastethernet | gigabitethernet | tengigabitethernet} slot/port | include type** command to see the queue structure of a port.

On **1q4t/2q2t** LAN ports, the receive- and transmit-queue tail-drop thresholds have this relationship:

- Receive queue 1 (standard) threshold 1 = transmit queue 1 (standard low priority) threshold 1
- Receive queue 1 (standard) threshold 2 = transmit queue 1 (standard low priority) threshold 2
- Receive queue 1 (standard) threshold 3 = transmit queue 2 (standard high priority) threshold 1
- Receive queue 1 (standard) threshold 4 = transmit queue 2 (standard high priority) threshold 2

To map CoS values to tail-drop thresholds, perform this task:

	Command	Purpose
Step 1	Router(config)# interface type ¹ slot/port	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue cos-map transmit_queue_# threshold_# cos1 [cos2 [cos3 [cos4 [cos5 [cos6 [cos7 [cos8]]]]]]]	Maps CoS values to a tail-drop threshold.
Step 3	Router(config-if)# no wrr-queue cos-map	Reverts to the default mapping.
Step 4	Router(config-if)# end	Exits configuration mode.
Step 5	Router# show queueing interface type ¹ slot/port	Verifies the configuration.

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

When mapping CoS values to a tail-drop threshold, note the following information:

- Use the transmit queue and threshold numbers.
- Queue 1 is the low-priority standard transmit queue.
- Queue 2 is the high-priority standard transmit queue.
- There are two thresholds in each queue.
- Enter up to 8 CoS values to map to the threshold.

This example shows how to map the CoS values 0 and 1 to standard transmit queue 1/threshold 1 for Fast Ethernet port 5/36:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/36
Router(config-if)# wrr-queue cos-map 1 1 0 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 5/36 | begin queue thresh cos-map
queue thresh cos-map
-----
1      1      0 1
1      2      2 3
2      1      4 5
2      2      6 7
<...Output Truncated...>
Router#
```

Allocating Bandwidth Between LAN-Port Standard Transmit Queues

The switch transmits frames from one standard queue at a time using one of these dequeuing algorithms, which use 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.

The higher the weight that is assigned to a queue, the more transmit bandwidth is allocated to it. 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 weights.

To allocate bandwidth for an egress LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue bandwidth <i>low_priority_queue_weight</i> <i>[intermediate_priority_queue_weights]</i> <i>high_priority_queue_weight</i> ² Router(config-if)# no wrr-queue bandwidth	Allocates bandwidth between standard transmit queues. The valid values for weight range from 1 to 255. Reverts to the default bandwidth allocation.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface <i>type</i> ¹ <i>slot/port</i>	Verifies the configuration.

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

2. You must enter weights for all the standard transmit queues on the interface (2, 3, or 7 weights).

This example shows how to allocate a 3-to-1 bandwidth ratio for Gigabit Ethernet port 1/2:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface gigabitethernet 1/2
Router(config-if)# wrr-queue bandwidth 3 1
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface gigabitethernet 1/2 | include bandwidth
WRR bandwidth ratios:      3[queue 1]    1[queue 2]
Router#
```

Setting the Receive-Queue Size Ratio on 1p1q0t and 1p1q8t Ingress LAN Ports

To set the size ratio between the strict-priority and standard receive queues on a **1p1q0t** or **1p1q8t** ingress LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface { fastethernet tengigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# rcv-queue queue-limit <i>standard_queue_weight</i> <i>strict_priority_queue_weight</i> Router(config-if)# no rcv-queue queue-limit	Sets the size ratio between the strict-priority and standard receive queues. Reverts to the default size ratio.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface { fastethernet tengigabitethernet } <i>slot/port</i>	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** ingress LAN ports, where valid values for the strict priority queue are from 3 to 100 percent.

This example shows how to set the receive-queue size ratio for Fast Ethernet port 2/2:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 2/2
Router(config-if)# rcv-queue queue-limit 75 15
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show queueing interface fastethernet 2/2 | include queue-limit
queue-limit ratios: 75[queue 1] 15[queue 2]
Router#
```

Setting the LAN-Port Transmit-Queue Size Ratio

To set the transmit-queue size ratio on an egress LAN port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the interface to configure.
Step 2	Router(config-if)# wrr-queue queue-limit <i>low_priority_queue_weight</i> <i>[intermediate_priority_queue_weights]</i> <i>high_priority_queue_weight</i> Router(config-if)# no wrr-queue queue-limit	Sets the transmit-queue size ratio between transmit queues. Reverts to the default transmit-queue size ratio.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show queueing interface <i>type</i> ¹ <i>slot/port</i>	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).
- On LAN ports that have an egress strict priority queue, 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 queueing interface gigabitethernet 1/2 | include queue-limit  
queue-limit ratios:      75[queue 1]  25[queue 2]  
Router#
```




Configuring PFC3BXL or PFC3B Mode MPLS QoS

This chapter describes how to configure PFC3BXL or PFC3B mode MPLS quality of service (QoS) on the Catalyst 6500 series switches.



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- PFC3BXL or PFC3B mode MPLS QoS extends to MPLS traffic the PFC QoS features described in Chapter 42, “Configuring PFC QoS.”
- This chapter provides supplemental information on PFC3BXL or PFC3B mode MPLS QoS features. Be sure that you understand the PFC QoS features before you read this chapter.
- All policing and marking available for PFC3BXL or PFC3B mode MPLS QoS are managed from the modular QoS command-line interface (CLI). The modular QoS CLI (MQC) is a command-line interface that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach those traffic policies to interfaces. A detailed description of the modular QoS CLI can be found in the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2 at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt8/

This chapter contains these sections:

- [Terminology, page 43-2](#)
- [PFC3BXL or PFC3B Mode MPLS QoS Overview, page 43-4](#)
- [PFC3BXL or PFC3B Mode MPLS QoS, page 43-5](#)
- [Understanding PFC3BXL or PFC3B Mode MPLS QoS, page 43-7](#)
- [PFC3BXL or PFC3B MPLS QoS Default Configuration, page 43-15](#)
- [MPLS QoS Commands, page 43-16](#)
- [PFC3BXL or PFC3B Mode MPLS QoS Restrictions and Guidelines, page 43-17](#)
- [Configuring PFC3BXL or PFC3B Mode MPLS QoS, page 43-17](#)
- [MPLS DiffServ Tunneling Modes, page 43-30](#)
- [Configuring Short Pipe Mode, page 43-34](#)
- [Configuring Uniform Mode, page 43-38](#)

Terminology

This section defines some MPLS QoS terminology:

- *Class of Service* (CoS) refers to three bits in either an 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 domain, the 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 packet 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 differentiated 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.

PFC3BXL or PFC3B Mode MPLS QoS Features

QoS enables a network to provide improved service to selected network traffic. This section explains the following PFC3BXL or PFC3B mode MPLS QoS features, which are supported in an MPLS network:

- [MPLS Experimental Field, page 43-3](#)
- [Preserving IP ToS, page 43-4](#)
- [Classification, page 43-3](#)
- [Policing and Marking, page 43-4](#)
- [Preserving IP ToS, page 43-4](#)
- [EXP Mutation, page 43-4](#)
- [MPLS DiffServ Tunneling Modes, page 43-4](#)

MPLS Experimental Field

Setting the MPLS experimental field (EXP) 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 PFC3BXL or PFC3B mode MPLS QoS policy.

Trust

For received MPLS packets, the PFC3BXL or PFC3B always trusts EXP in the received topmost label. None of these have any effect on MPLS packets:

- Interface trust state
- Port CoS value
- Policy-map **trust** command

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 PFC3BXL or PFC3B make classification decisions based on the EXP bits in the received topmost label of received MPLS packets (after a policy is installed). See [“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 PFC3BXL or PFC3B 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 43-22](#) for information.

Preserving IP ToS

The PFC3BXL or PFC3B automatically preserves the IP ToS during all MPLS operations including imposition, swapping, and disposition. You do not need to enter a command to save the IP ToS.

EXP Mutation

You can configure up to eight egress EXP mutation maps to mutate the internal EXP value before it is written as the egress EXP value. You can attach egress EXP mutation maps to these interface types:

- 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 [“Configuring PFC3BXL or PFC3B mode MPLS QoS Egress EXP Mutation” section on page 43-28](#).

MPLS DiffServ Tunneling Modes

The PFC3BXL or PFC3B uses MPLS DiffServ tunneling modes. Tunneling provides QoS transparency from one edge of a network to the other edge of the network. See [“MPLS DiffServ Tunneling Modes” section on page 43-30](#) for information.

PFC3BXL or PFC3B Mode MPLS QoS Overview

PFC3BXL or PFC3B 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.

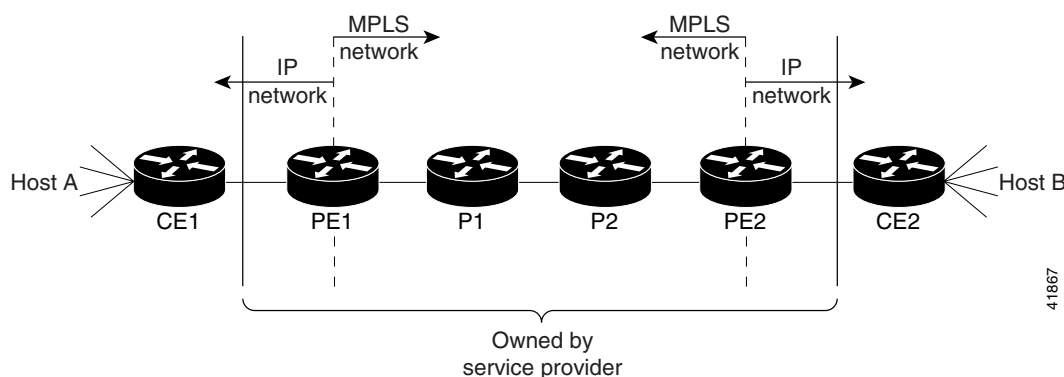
For more information, see [“MPLS DiffServ Tunneling Modes”](#) section on page 43-30.

PFC3BXL or PFC3B Mode MPLS QoS

This section describes how PFC3BXL or PFC3B 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.

- [LERs at the Input Edge of an MPLS Network, page 43-6](#)
- [LSRs in the Core of an MPLS Network, page 43-6](#)
- [LERs at the Output Edge of an MPLS Network, page 43-7](#)

**Note**

The QoS capabilities at the input interface differ depending on whether the input interface is a LAN port, a WAN port on an OSM, or a port adapter on a FlexWAN or Enhanced FlexWAN module. This section is for LAN ports. For information on OSMs, see the *OSM Configuration Note, 12.2SX*. For information on a FlexWAN or Enhanced FlexWAN module, see the *FlexWAN and Enhanced FlexWAN Installation and Configuration Note*.

LERs at the Input Edge of an MPLS Network

**Note**

Incoming labels are aggregate or nonaggregate. The aggregate label indicates that the arriving MPLS or MPLS VPN packet must be switched through an IP lookup to find the next hop and the outgoing interface. The nonaggregate label indicates that the packet contains the IP next hop information.

This section describes how edge LERs can operate at either the ingress or the egress side of an MPLS network.

At the ingress side of an MPLS network, LERs process packets as follows:

1. Layer 2 or Layer 3 traffic enters the edge of the MPLS network at the edge LER (PE1).
2. The PFC3BXL or PFC3B receives the traffic from the input interface and uses the 802.1p bits or the IP ToS bits to determine the EXP bits and to perform any classification, marking, and policing. For classification of incoming IP packets, the input service-policy can also use ACLs.
3. For each incoming packet, the PFC3BXL or PFC3B performs a lookup on the IP address to determine the next-hop router.
4. The appropriate label is pushed (imposition) into the packet, and the EXP value resulting from the QoS decision is copied into the MPLS EXP field in the label header.
5. The PFC3BXL or PFC3B forwards the labeled packets to the appropriate output interface for processing.
6. The PFC3BXL or PFC3B also forwards the 802.1p bits or the IP ToS bits to the output interface.
7. At the output interface, the labeled packets are differentiated by class for marking or policing. For LAN interfaces, egress classification is still based on IP, not on MPLS.
8. The labeled packets (marked by EXP) are sent to the core MPLS network.

LSRs in the Core of an MPLS Network

This section describes how LSRs used at the core of an MPLS network process packets:

1. Incoming MPLS labeled packets (and 802.1p bits or IP ToS bits) from an edge LER (or other core device) arrive at the core LSR.
2. The PFC3BXL or PFC3B receives the traffic from the input interface and uses the EXP bits and to perform classification, marking, and policing.

3. The PFC3BXL or PFC3B performs a table lookup to determine the next-hop LSR.
4. An appropriate label is placed (swapped) into the packet and the MPLS EXP bits are copied into the label header.
5. The PFC3BXL or PFC3B forwards the labeled packets to the appropriate output interface for processing.
6. The PFC3BXL or PFC3B also forwards the 802.1p bits or the IP ToS bits to the output interface.
7. The outbound packet is differentiated by the MPLS EXP field for marking or policing.
8. The labeled packets (marked with EXP) are sent to another LSR in the core MPLS network or to an LER at the output edge.

**Note**

Within the service provider network, there is no IP precedence field for the queueing algorithm to use because the packets are MPLS packets. The packets remain MPLS packets until they arrive at PE2, the provider edge router.

LERs at the Output Edge of an MPLS Network

At the egress side of an MPLS network, LERs process packets as follows:

1. MPLS-labeled packets (and 802.1p bits or IP ToS bits) from a core LSR arrive at the egress LER (PE2) from the MPLS network backbone.
2. The PFC3BXL or PFC3B pops the MPLS labels (disposition) from the packets. Aggregate labels are classified using the original 802.1p bits or the IP ToS bits. Non-aggregate labels are classified with the EXP value by default.
3. For aggregate labels, the PFC3BXL or PFC3B performs a lookup on the IP address to determine the packet's destination; the PFC3BXL or PFC3B then forwards the packet to the appropriate output interface for processing. For non-aggregate labels, forwarding is based on the label. By default, non-aggregate labels are popped at the penultimate-hop router (next to last), not egress PE.
4. The PFC3BXL or PFC3B also forwards the 802.1p bits or the IP ToS bits to the output interface.
5. The packets are differentiated according to the 802.1p bits or the IP ToS bits and treated accordingly.

**Note**

The MPLS experimental 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 PFC3BXL or PFC3B Mode MPLS QoS

PFC3BXL or PFC3B mode MPLS QoS supports IP QoS. For MPLS packets, the EXP value is mapped into an internal DSCP so that the PFC3BXL or PFC3B can apply non-MPLS QoS marking and policing.

For both the ingress and egress policy, PFC3BXL or PFC3B mode MPLS QoS marking and policing decisions are made on a per-interface basis at an ingress PFC3BXL or PFC3B. The ingress interfaces are physical ports, subinterfaces, or VLANs.

The QoS policy ACLs are programmed in QoS TCAM separately for ingress and egress lookup. The 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 DSCP.

This section describes PFC3BXL or PFC3B 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 43-12](#)


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 EoMPLS header (VC type 4); otherwise, queueing is based on the CoS derived from the QoS decision.

Ethernet to MPLS

For Ethernet to MPLS, the ingress interface, PFC3BXL or PFC3B mode MPLS QoS, and egress interface features are similar to corresponding features for IP to MPLS. For more information, see these sections:

- [IP-to-MPLS Ingress Port Classification, page 43-10](#)
- [IP-to-MPLS PFC3BXL or PFC3B Mode MPLS QoS Classification, page 43-10](#)
- [IP-to-MPLS Egress Port Classification, page 43-10](#)

MPLS to Ethernet

For MPLS to Ethernet, the ingress interface, PFC3BXL or PFC3B 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:

- [MPLS-to-IP Ingress Port Classification, page 43-11](#)
- [MPLS-to-IP PFC3BXL or PFC3B Mode MPLS QoS Classification, page 43-11](#)
- [MPLS-to-IP Egress Port Classification, page 43-11.](#)

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 PFC3BXL or PFC3B provides the following MPLS QoS capabilities at the IP-to-MPLS edge:

- Assigning an EXP value based on the **mls qos trust** or **policy-map** command
- Marking an EXP value using a policy
- Policing traffic using a policy

This section provides information about the MPLS QoS classification that the 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.

IP-to-MPLS Classification

The PFC3BXL or PFC3B ingress and egress policies for IP traffic classify traffic on the original received IP using **match** commands for IP precedence, IP DSCP, and IP ACLs. Egress policies do not classify traffic on the imposed EXP value nor on a marking done by an ingress policy.

After the PFC3BXL or PFC3B applies the port trust and QoS policies, it assigns the internal DSCP. The PFC3BXL or PFC3B then assigns the EXP value based on the internal DSCP-to-EXP global map for the labels that it imposes. If more than one label is imposed, the EXP value is the same in each label. The PFC3BXL or PFC3B preserves the original IP ToS when the MPLS labels are imposed.

The PFC3BXL or PFC3B assigns the egress CoS based on the internal DSCP-to-CoS global map. If the default internal DSCP-to-EXP and the internal DSCP-to-CoS maps are consistent, then the egress CoS has the same value as the imposed EXP.

If the ingress port receives both IP-to-IP and IP-to-MPLS traffic, classification should be used to separate the two types of traffic. For example, if the IP-to-IP and IP-to-MPLS traffic have different destination address ranges, you can classify traffic on the destination address, and then apply IP ToS policies to the IP-to-IP traffic and apply a policy (that marks or sets the EXP value in the imposed MPLS header) to the IP-to-MPLS traffic. See the following two examples:

- A PFC3BXL or PFC3B policy to mark IP ToS sets the internal DSCP—If it is applied to all traffic, then for IP-to-IP traffic, the egress port will rewrite the CoS (derived from the internal DSCP) to the IP ToS byte in the egress packet. For IP-to-MPLS traffic, the PFC3BXL or PFC3B will map the internal DSCP to the imposed EXP value.
- A PFC3BXL or PFC3B policy to mark MPLS EXP sets the internal DSCP—If it is applied to all traffic, then for IP-to-IP traffic, the egress port rewrites the IP ToS according to the ingress IP policy (or trust). The CoS is mapped from the ToS. For IP-to-MPLS traffic, the PFC3BXL or PFC3B will map the internal DSCP to the imposed EXP value.

IP-to-MPLS PFC3BXL or PFC3B Mode MPLS QoS Classification

PFC3BXL or PFC3B mode MPLS QoS at the ingress to PE1 supports:

- Matching on IP precedence or DSCP values or filtering with an access group
- The **set mpls experimental imposition** and **police** commands

PFC3BXL or PFC3B mode MPLS QoS at the egress of PE1 supports the **mpls experimental topmost** command.

IP-to-MPLS Ingress Port Classification

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.

IP-to-MPLS Egress Port Classification

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, LAN ports and OSM GE-WAN ports copy the egress CoS into the egress 802.1Q field.

MPLS to IP

PFC3BXL or PFC3B mode MPLS QoS supports these capabilities at the MPLS-to-IP edge:

- Option to propagate EXP value into IP DSCP on exit from an MPLS domain per egress interface.
- Option to use IP service policy on the MPLS-to-IP egress interface.

This section provides information about the MPLS-to-IP MPLS QoS classification. Additionally, this section provides information about the capabilities provided by the ingress and egress modules.

MPLS-to-IP Classification

The PFC3BXL or PFC3B assigns the internal DSCP (internal priority that the PFC3BXL or PFC3B assigns to each frame) based on the QoS result. The QoS result is affected by the following:

- Default trust EXP value
- Label type (per-prefix or aggregate)
- Number of VPNs
- Explicit NULL use
- QoS policy

There are three different classification modes:

- Regular MPLS classification—For nonaggregate labels, in the absence of MPLS recirculation, the PFC3BXL or PFC3B classifies the packet based on MPLS EXP ingress or egress policy. The PFC3BXL queues the packet based on COS derived from EXP-to-DSCP-to-CoS mapping. The underlying IP DSCP is either preserved after egress decapsulation, or overwritten from the EXP (through the EXP-to-DSCP map).
- IP classification for aggregate label hits in VPN CAM—The PFC3BXL or PFC3B does one of the following:
 - Preserves the underlying IP ToS
 - Rewrites the IP ToS by a value derived from 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 PFC3BXL or PFC3B differentiates the MPLS-to-IP packets from the regular IP-to-IP packets based on the ingress reserved VLAN specified in the MPLS decapsulation adjacency. The reserved VLAN is allocated per VRF both for VPN and non-VPN cases. The ingress ToS after recirculation can be either the original IP ToS value, or derived from the original EXP value. The egress IP policy can overwrite this ingress ToS to an arbitrary value.



Note

For information about recirculation, see the [“Recirculation” section on page 25-4](#).

For incoming MPLS packets on the PE-to-CE ingress, the PFC3BXL or PFC3B supports MPLS classification only. Ingress IP policies are not supported. PE-to-CE traffic from the MPLS core is classified or policed on egress as IP.

MPLS-to-IP PFC3BXL or PFC3B Mode MPLS QoS Classification

PFC3BXL or PFC3B mode MPLS QoS at the ingress to PE2 supports matching on the EXP value and the **police** command.

PFC3BXL or PFC3B 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.

MPLS-to-IP Ingress Port Classification

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.

MPLS-to-IP Egress Port Classification



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 PFC3BXL or PFC3B 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 egress interface if 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

In addition to the information above, the following information 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 [“Recirculation” section on page 25-4](#).

The PFC3BXL or PFC3B propagates the EXP value if all interfaces in the VPN have EXP propagation enabled.

The following PE MPLS QoS features are supported:

- General MPLS QoS features for IP packets
- Classification, policing, or marking of CE-to-PE IP traffic through the VPN subinterface
- Per-VPN QoS (per-port, per-VLAN, or per-subinterface)

LSRs at the MPLS Core

This section provides information about MPLS QoS features for LSRs at the core (MPLS-to-MPLS) for MPLS and MPLS VPN networks. Ingress features, egress interface, and PFC3BXL or PFC3B features for Carrier Supporting Carrier (CsC) QoS features are similar to those used with MPLS to MPLS described in the next section. A difference between CsC and MPLS to MPLS is that with CsC labels can be imposed inside the MPLS domain.

MPLS to MPLS

PFC3BXL or PFC3B 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 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 PFC3BXL or PFC3B mode MPLS QoS classification. Additionally, the section provides information about the capabilities provided by the ingress and egress modules.

MPLS-to-MPLS Classification

For received MPLS packets, the PFC3BXL or PFC3B ignores the port trust state, the ingress CoS, and any policy-map **trust** commands. Instead, the PFC3BXL or PFC3B trusts the EXP value in the topmost label.



Note

The PFC3BXL or PFC3B 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.

PFC3BXL or PFC3B mode MPLS QoS maps the EXP value to the internal DSCP using the EXP-to-DSCP global map. What the PFC3BXL or PFC3B does next depends on whether it is swapping labels, imposing a new label, or popping a label:

- Swapping labels—When swapping labels, the PFC3BXL or PFC3B preserves the EXP value in the received topmost label and copies it to the EXP value in the outgoing topmost label. The PFC3BXL or PFC3B assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP global maps are consistent, then the egress CoS is based on the EXP in the outgoing topmost label.

The PFC3BXL or PFC3B can mark down out-of-profile traffic using the **police** command's **exceed** and **violate** actions. It does not mark in-profile traffic, so the **conform** action must be transmitted and the **set** command cannot be used. If the PFC3BXL or PFC3B is performing a markdown, it uses the internal DSCP as an index into the internal DSCP markdown map. The PFC3BXL or PFC3B maps the result of the internal DSCP markdown to an EXP value using the internal DSCP-to-EXP global map. The PFC3BXL or PFC3B rewrites the new EXP value to the topmost outgoing label and does not copy the new EXP value to the other labels in the stack. The PFC3BXL or PFC3B assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, then the egress CoS is based on the EXP value in the topmost outgoing label.

- Imposing an additional label—When imposing a new label onto an existing label stack, the PFC3BXL or PFC3B maps the internal DSCP to the EXP value in the imposed label using the internal DSCP-to-EXP map. It then copies the EXP value in the imposed label to the underlying swapped label. The PFC3BXL or PFC3B assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, the egress CoS is based on the EXP value in the imposed label.

The PFC3BXL or PFC3B can mark in-profile and mark down out-of-profile traffic. After it marks the internal DSCP, the PFC3BXL or PFC3B uses the internal DSCP-to-EXP global map to map the internal DSCP to the EXP value in the newly imposed label. The PFC3BXL or PFC3B then copies

the EXP in the imposed label to the underlying swapped label. The PFC3BXL or PFC3B assigns the egress CoS using the internal DSCP-to-CoS global map. Therefore, the egress CoS is based on the EXP in the imposed label.

- Popping a label—When popping a label from a multilabel stack, the PFC3BXL or PFC3B preserves the EXP value in the exposed label. The PFC3BXL or PFC3B assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, then the egress CoS is based on the EXP value in the popped label.
- If EXP propagation is configured for the egress interface, the PFC3BXL or PFC3B maps the internal DSCP to the EXP value in the exposed label using the DSCP-to-EXP global map. The PFC3BXL or PFC3B assigns the egress CoS using the internal DSCP-to-CoS global map. If the DSCP maps are consistent, the egress CoS is based on the EXP value in the exposed label.

MPLS-to-MPLS PFC3BXL or PFC3B Mode MPLS QoS Classification

PFC3BXL or PFC3B 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

PFC3BXL or PFC3B mode MPLS QoS at the egress of P1 or P2 supports matching with the **mpls experimental topmost** command.

MPLS-to-MPLS Ingress Port Classification

LAN port classification is based on the egress CoS from the PFC3BXL or PFC3B. OSM and FlexWAN interfaces classify traffic using the **match mpls experimental** command. The **match mpls experimental** command matches on the EXP value in the received topmost label.

MPLS-to-MPLS Egress Port Classification

LAN port classification is based on the egress CoS value from the PFC3BXL or PFC3B. OSM and FlexWAN interfaces classify traffic using the **match mpls experimental** command. The **match mpls experimental** command matches on the egress CoS; it does not match on the EXP in the topmost label.

If the egress port is a trunk, LAN ports and OSM GE-WAN ports copy the egress CoS into the egress 802.1Q field.

PFC3BXL or PFC3B MPLS QoS Default Configuration

This section describes the PFC3BXL or PFC3B MPLS QoS default configuration. The following global PFC3BXL or PFC3B MPLS QoS settings apply:

Feature	Default Value
PFC QoS global enable state	<p>Note With PFC QoS disabled and all other PFC QoS parameters at default values, default EXP is mapped from IP precedence.</p> <p>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.</p>
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	<p>There are four possibilities at the MPLS core QoS:</p> <ul style="list-style-type: none"> 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. <p>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.</p> <ul style="list-style-type: none"> 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

PFC3BXL or PFC3B MPLS QoS on the Catalyst 6500 series switches supports the following MPLS QoS commands:

- **match mpls experimental topmost**
- **set mpls experimental imposition**
- **police**
- **mls qos map exp-dscp**
- **mls qos map dscp-exp**
- **mls qos map exp-mutation**
- **mls qos exp-mutation**
- **show mls qos mpls**



Note

For information about supported non-MPLS QoS commands, see [“Configuring PFC QoS” section on page 42-41](#).

The following commands are not supported:

- **set qos-group**
- **set discard-class**

PFC3BXL or PFC3B Mode MPLS QoS Restrictions and Guidelines

When configuring PFC3BXL or PFC3B mode MPLS QoS, follow these guidelines and restrictions:

- For IP-to-MPLS or EoMPLS imposition when the received packet is an IP packet:
 - When QoS is disabled, the EXP value is based on the received IP ToS.
 - When QoS is queuing only, the EXP value is based on the received IP ToS.
- For EoMPLS imposition when the received packet is a non-IP packet:
 - When QoS is disabled, the EXP value is based on the ingress CoS.
 - When QoS is queuing only, the EXP value is based on the received IP ToS.
- For MPLS-to-MPLS operations:
 - Swapping when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation)
 - Swapping when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation)
 - Imposing additional label when QoS is disabled, the EXP value is based on the original EXP value (in the absence of EXP mutation)
 - Imposing an additional label when QoS is queuing only, the EXP value is based on the original EXP value (in the absence of EXP mutation)
 - Popping one label when QoS is disabled, the EXP value is based on the underlying EXP value
 - Popping one label when QoS is queuing only, the EXP value is based on the underlying EXP value
- EXP value is irrelevant to MPLS-to-IP disposition.
- The **no mls qos rewrite ip dscp** command is incompatible with MPLS. The default **mls qos rewrite ip dscp** command must remain enabled in order for the PFC3BXL or PFC3B to assign the correct EXP value for the labels that it imposes.

Configuring PFC3BXL or PFC3B Mode MPLS QoS

These sections describe how to configure PFC3BXL or PFC3B mode MPLS QoS:

- [Enabling QoS Globally, page 43-18](#)
- [Enabling Queueing-Only Mode, page 43-19](#)
- [Configuring a Class Map to Classify MPLS Packets, page 43-20](#)
- [Configuring a Policy Map, page 43-22](#)
- [Configuring a Policy Map Using the Police Command, page 43-24](#)
- [Displaying a Policy Map, page 43-27](#)
- [Configuring PFC3BXL or PFC3B mode MPLS QoS Egress EXP Mutation, page 43-28](#)
- [Configuring EXP Value Maps, page 43-29](#)

Enabling QoS Globally

Before you can configure QoS on the PFC3BXL or PFC3B, you must enable the QoS functionality globally using the **mls qos** command. This command enables default QoS conditioning of traffic.

When the **mls qos** command is enabled, the PFC3BXL or PFC3B assigns a priority value to each frame. This value is the internal DSCP. The internal DSCP is assigned based on the contents of the received frame and the QoS configuration. This value is rewritten to the egress frame’s CoS and ToS fields.

To enable QoS globally, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos	Enables PFC QoS globally on the switch.
	Router(config)# no mls qos	Disables PFC QoS globally on the switch.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos	Verifies the configuration.

This example shows how to enable QoS globally:

```
Router(config)# mls qos
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos
QoS is enabled globally
  Microflow policing is enabled globally
  QoS ip packet dscp rewrite enabled globally

Qos trust state is DSCP on the following interfaces:
  Gi4/1 Gi4/1.12

Qos trust state is IP Precedence on the following interfaces:
  Gi4/2 Gi4/2.42
Vlan or Portchannel(Multi-Earl) policies supported: Yes
Egress policies supported: Yes

----- Module [5] -----
QoS global counters:
  Total packets: 5957870
  IP shortcut packets: 0
  Packets dropped by policing: 0
  IP packets with TOS changed by policing: 6
  IP packets with COS changed by policing: 0
  Non-IP packets with COS changed by policing: 3
  MPLS packets with EXP changed by policing: 0
```


Enabling Queueing-Only Mode

To enable queueing-only mode, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos queueing-only	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 switch does the following:

- Disables marking and policing globally
- Configures all ports to trust Layer 2 CoS



Note The switch applies the port CoS value to untagged ingress traffic and to traffic that is received through ports that cannot be configured to trust CoS.

This example shows how to enable queueing-only mode:

```
Router# configure terminal
Router(config)# mls qos queueing-only
Router(config)# end
Router#
```

Restrictions and Usage Guidelines

If QoS is disabled (**no mls qos**) for the PFC3BXL or PFC3B, 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



Note

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 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 <i>class_name</i>	Specifies the class map to which packets will be matched.
Step 2	Router(config-cmap)# match mpls experimental topmost <i>value</i>	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
      -----
      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#
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
      -----
      V1300  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 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. PFC3BXL or PFC3B MPLS QoS does not attempt to apply commands from more than one policy map class to matched traffic.

Configuring a Policy Map to Set the EXP Value on All Imposed Labels

To set the value of the MPLS EXP field on all imposed label entries, use the **set mpls experimental imposition** command in QoS policy-map class configuration mode. To disable the setting, use the no form of this command.



Note

The **set mpls experimental imposition** command replaces the **set mpls experimental** command.

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy_name</i>	Creates a policy map.
Step 2	Router(config-pmap)# class-map <i>name</i> [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-1 101 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
      -----
      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 PFC3BXL or PFC3B, you can mark the EXP field using the **set mpls experimental imposition** command.

For more information on this command, see the *Cisco IOS Switching Services Command Reference, Release 12.3* located at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgr/swtch_r/swi_n1.htm

Configuring a Policy Map Using the Police Command

Policing is a function in the PFC3BXL or PFC3B hardware that provides the ability to rate limit a particular traffic class to a specific rate. The PFC3BXL or PFC3B supports aggregate policing and microflow policing.

Aggregate policing meters all traffic that ingresses into a port, regardless of different source, destination, protocol, source port, or destination port. Microflow policing meters all traffic that ingresses into a port, on a per flow (per source, destination, protocol, source port, and destination port). For additional information on aggregate and microflow policing, see “[Policers](#)” section on page 42-17.

To configure traffic policing, use the **police** command. For information on this command, see the *Catalyst 6500 Series Switch Cisco IOS Command Reference, Release 12.2SX*.

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy_name</i>	Creates a policy map.
Step 2	Router(config-pmap)# class-map <i>name</i> [match-all match-any]	Accesses the QoS class map configuration mode to configure QoS class maps.
Step 3	Router(config-pmap-c)# police { aggregate <i>name</i> }	Adds the class to a shared aggregate policer.
Step 4	Router(config-pmap-c)# police <i>bps burst_normal burst_max conform-action action exceed-action action violate-action action</i>	Creates a per-class-per-interface policer.
Step 5	Router(config-pmap-c)# police flow { <i>bps [burst_normal]</i> [conform-action <i>action</i>] [exceed-action <i>action</i>]}	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
      -----
      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
      -----
      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 [“Configuring a Policy Map” section on page 42-57](#).
- 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 PFC3BXL or PFC3B, you can mark down out-of-profile traffic using the **police** command **exceed-action policed-dscp-transmit** and **violate-action policed-dscp-transmit** keywords. The PFC3BXL or PFC3B does not mark in-profile traffic; when marking down out-of-profile traffic, the PFC3BXL or PFC3B marks the outgoing topmost label. The PFC3BXL or PFC3B does not propagate the marking down through the label stack.
- With MPLS, the flow key is based on the label and EXP value; there is no flowmask option. Otherwise, flow key operation is similar to IP-to-IP. See [“Configuring a Policy Map” section on page 42-57](#).
- 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 PFC3BXL or PFC3B, you can mark the EXP field using the **conform-action set-mpls-exp-imposition-transmit** keywords.

- During IP-to-MPLS imposition, IP ToS marking is not supported. If you configure a policy to mark IP ToS, the PFC3BXL or PFC3B marks the EXP value.

Displaying a Policy Map

You can display a policy map with an interface summary for MPLS QoS classes or with the configuration of all classes configured for all service policies on the specified interface.

Displaying a PFC3BXL or PFC3B Mode MPLS QoS Policy Map Class Summary

To display a PFC3BXL or PFC3B mode MPLS QoS policy map class summary, perform this task:

Command	Purpose
Router# show mls qos mpls [{ interface <i>interface_type</i> <i>interface_number</i> } { module <i>slot</i> }]	Displays a PFC3BXL or PFC3B mode MPLS QoS policy map class summary.

This example shows how to display a PFC3BXL or PFC3B 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 Id	Trust	Fl Id	AgForward-By	AgPoliced-By
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 <i>interface_type</i> <i>interface_number</i>	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 PFC3BXL or PFC3B mode MPLS QoS Egress EXP Mutation

These sections describe how to configure PFC3BXL or PFC3B mode MPLS QoS egress EXP mutation:

- [Configuring Named EXP Mutation Maps, page 43-28](#)
- [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 <i>name</i> <i>mutated_exp1 mutated_exp2 mutated_exp3</i> <i>mutated_exp4 mutated_exp5 mutated_exp6</i> <i>mutated_exp7 mutated_exp8</i>	Configures a named EXP mutation map.
	Router(config)# no mls qos map exp-mutation <i>name</i>	Reverts to the default map.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos maps	Verifies the configuration.

When configuring a named EXP mutation map, note the following information:

- You can enter up to eight input EXP values that map to a mutated EXP value.
- You can enter multiple commands to map additional EXP values to a mutated EXP value.
- You can enter a separate command for each mutated EXP value.
- You can configure 15 ingress EXP mutation maps to mutate the internal EXP value before it is written as the ingress-EXP value. You can attach ingress EXP mutation maps to any interface that PFC QoS supports.
- PFC QoS derives the egress EXP value from the internal DSCP value. If you configure ingress EXP mutation, PFC QoS does not derive the ingress EXP value from the mutated EXP value.

Attaching an Egress EXP Mutation Map to an Interface

To attach an egress EXP mutation map to an interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{vlan vlan_ID} {type ¹ slot/port[.subinterface]} {port-channel number[.subinterface]}}	Selects the interface to configure.
Step 2	Router(config-if)# mls qos exp-mutation exp-mutation-table-name	Attaches an egress EXP mutation map to the interface.
	Router(config-if)# no mls qos exp-mutation	Removes the egress DSCP mutation map from the interface.
Step 3	Router(config-if)# end	Exits configuration mode.
Step 4	Router# show running-config interface {{vlan vlan_ID} {type slot/port} {port-channel number}}	Verifies the configuration.

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

This example shows how to attach the egress EXP mutation map named mutemap2:

```
Router(config)# interface fastethernet 3/26
Router(config-if)# mls qos exp-mutation mutemap2
Router(config-if)# end
```

Configuring EXP Value Maps

These sections describe how EXP values are mapped to other values:

- [Configuring an Ingress-EXP to Internal-DSCP Map, page 43-29](#)
- [Configuring a Named Egress-DSCP to Egress-EXP Map, page 43-30](#)

Configuring an Ingress-EXP to Internal-DSCP Map

To configure an ingress-EXP to internal-DSCP map, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos map exp-dscp values	Configures the ingress-EXP value to internal-DSCP map. You must enter eight DSCP values corresponding to the EXP values. Valid values are 0 through 63.
	Router(config)# no mls qos map exp-dscp	Reverts to the default map.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos maps	Verifies the configuration.

This example shows how to configure an ingress-EXP to internal-DSCP map:

```
Router(config)# mls qos map exp-dscp 43 43 43 43 43 43 43 43
Router(config)#
```

This example shows how to verify the configuration:

```
Router(config)# show mls qos map exp-dscp
Exp-dscp map:
  exp:   0   1   2   3   4   5   6   7
-----
  dscp:  43  43  43  43  43  43  43  43
```

Configuring a Named Egress-DSCP to Egress-EXP Map

To configure a named egress-DSCP to egress-EXP map, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos map dscp-exp dscp_values to exp_values	Configures a named egress-DSCP to egress-EXP map. You can enter up to eight DSCP values at one time to a single EXP value. Valid values are 0 through 7.
	Router(config)# no mls qos map dscp-exp	Reverts to the default map.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos maps	Verifies the configuration.

This example shows how to configure a named egress-DSCP to egress-EXP map:

```
Router(config)# mls qos map dscp-exp 20 25 to 3
Router(config)#
```

MPLS DiffServ Tunneling Modes

Tunneling provides QoS the ability to be transparent from one edge of a network to the other edge of the network. A tunnel starts where there is label imposition. A tunnel ends where there is label disposition; that is, where the label is removed from the stack, and the packet goes out as an MPLS packet with a different per-hop behavior (PHB) layer underneath or as an IP packet with the IP PHB layer.

For PFC3BXL or PFC3Bs, 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 43-38](#).

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 more information, see this publication:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftdtmode.htm>

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.

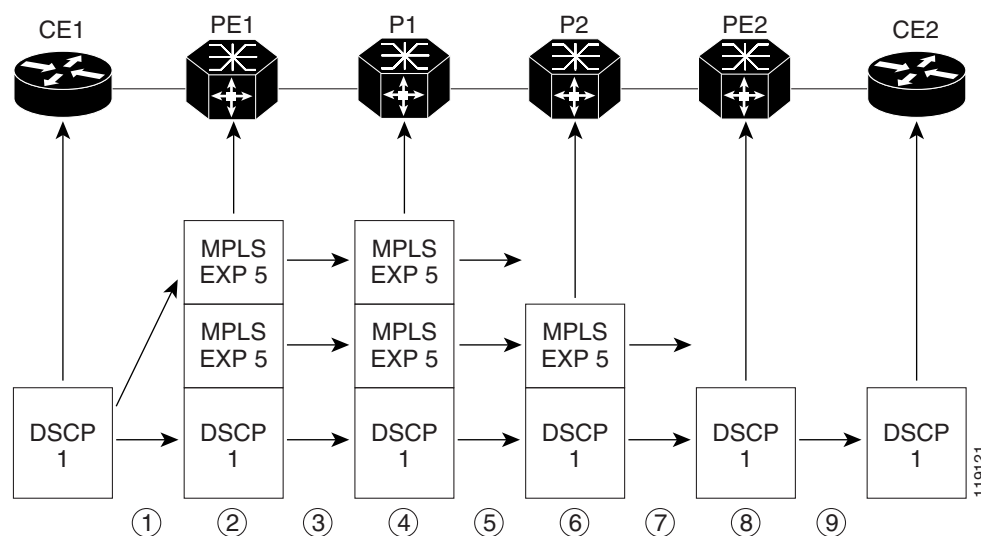
QoS policies implemented in the core do not propagate to the packet ToS byte. The classification based on MPLS EXP value ends at the customer-facing egress PE interface; classification at the customer-facing egress PE interface is based on the original IP packet header and not the MPLS header.



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/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftdtmode.htm>

Short Pipe Restrictions and Guidelines

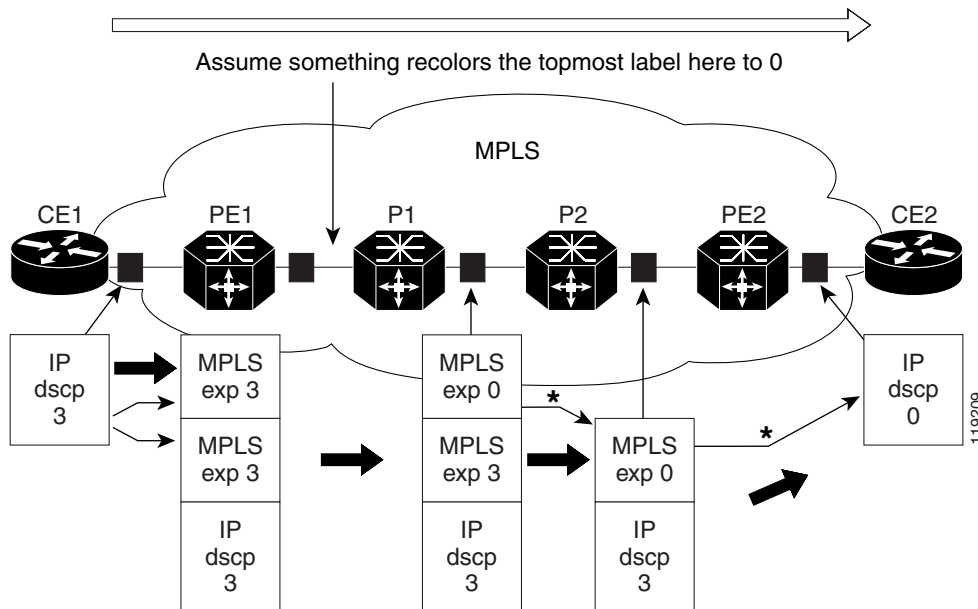
The following restriction applies to Short Pipe mode:

- Short Pipe mode is not supported if the MPLS-to-IP egress interface is EoMPLS (the adjacency has 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

This section describes how to configure the following:

- [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 43-36](#)
- [Configuring the Egress PE Router—Customer Facing Interface, page 43-37](#)



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 PFC3BXL or PFC3B 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 <i>ipv4_acl_number_or_name</i> permit any	Creates an IPv4 access list.
Step 3	Router(config)# class-map <i>class_name</i>	Creates a class map.
Step 4	Router(config-cmap)# match access-group <i>ipv4_acl_number_or_name</i>	Configures the class map to filter with the ACL created in step 2.
Step 5	Router(config)# policy-map <i>policy_map_name</i>	Creates a named QoS policy.
Step 6	Router(config-pmap)# class <i>class_name</i>	Configures the policy to use the class map created in step 3.
Step 7	Router(config-pmap-c)# police <i>bits_per_second</i> [<i>normal_burst_bytes</i>] conform-action set-mpls-exp-transmit <i>exp_value</i> exceed-action drop	Configures policing, including the following: <ul style="list-style-type: none"> • 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. <p>The <i>exp_value</i> sets the MPLS EXP field.</p>
Step 8	Router(config)# interface <i>type slot/port</i>	Selects an interface to configure.

	Command	Purpose
Step 9	Router(config-if)# no mls qos trust	Configures the interface as untrusted.
Step 10	Router(config-if)# service-policy input <i>policy_map_name</i>	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 <i>class_name</i>	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	Router(config-c-map)# match mpls experimental <i>exp_list</i>	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 <i>name</i>	Configures the QoS policy for packets that match the class or classes.
Step 5	Router(config-p-map)# class <i>class_name</i>	Associates the traffic class with the service policy.
Step 6	Router(config-p-map-c)# bandwidth { <i>bandwidth_kbps</i> 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.

	Command	Purpose
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 <i>type slot/port</i>	Selects an interface to configure.
Step 10	Router(config-if) # service-policy output <i>name</i>	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 <i>class_name</i>	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	Router(config-c-map) # match mpls experimental <i>exp_list</i>	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 <i>name</i>	Configures the QoS policy for packets that match the class or classes.
Step 5	Router(config-p-map) # class <i>class_name</i>	Associates the traffic class with the service policy.
Step 6	Router(config-p-map-c) # bandwidth { <i>bandwidth_kbps</i> percent <i>percent</i> }	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.

	Command	Purpose
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.

	Command	Purpose
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:

- [Uniform Mode Restrictions and Guidelines, page 43-33](#)
- [Configuring the Ingress PE Router—Customer Facing Interface, page 43-38](#)
- [Configuring the Ingress PE Router—P Facing Interface, page 43-40](#)
- [Configuring the Egress PE Router—Customer Facing Interface, page 43-41](#)

**Note**

The steps that follow show one way, but not the only way, to configure the Uniform mode.

Configuring the Ingress PE Router—Customer Facing Interface

For Uniform mode, setting the trust state to IP precedence or IP DSCP allows the PFC3BXL or PFC3B to copy the IP PHB into the MPLS PHB.

**Note**

This description applies to PFC3BXL or PFC3B QoS for LAN or OSM ports. For information about FlexWAN and Enhanced FlexWAN QoS, see the *Enhanced FlexWAN Module, Fabric Enabled Installation and Configuration Note* at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/core/cis7600/cfgnotes/flexport/4885_01.htm.

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 <i>ipv4_acl_number_or_name</i> permit any	Creates an IPv4 access list.
Step 3	Router(config)# class-map <i>class_name</i>	Creates a class map.
Step 4	Router(config-cmap)# match access-group <i>ipv4_acl_number_or_name</i>	Configures the class map to filter with the ACL created in step 2.
Step 5	Router(config)# policy-map <i>policy_map_name</i>	Creates a named QoS policy.
Step 6	Router(config-pmap)# class <i>class_name</i>	Configures the policy to use the class map created in step 3.
Step 7	Router(config-pmap-c)# police <i>bits_per_second</i> [<i>normal_burst_bytes</i>] conform-action transmit exceed-action drop	Configures policing, including the following: <ul style="list-style-type: none"> 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 <i>type slot/port</i>	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 <i>policy_map_name</i>	Attaches the policy map created in step 5 to the interface as an input service policy.

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-if)# 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 <i>class_name</i>	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	Router(config-c-map)# match mpls experimental <i>exp_list</i>	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 <i>name</i>	Configures the QoS policy for packets that match the class or classes.
Step 5	Router(config-p-map)# class <i>class_name</i>	Associates the traffic class with the service policy.
Step 6	Router(config-p-map-c)# bandwidth { <i>bandwidth_kbps</i> percent <i>percent</i> }	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 <i>class-default</i>	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 <i>type slot/port</i>	Selects an interface to configure.
Step 10	Router(config-if)# service-policy <i>output name</i>	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 <i>class_name</i>	Specifies the class map to which packets will be mapped (matched). Creates a traffic class.
Step 3	Router(config-c-map)# match ip precedence <i>precedence-value</i>	Identifies IP precedence values as match criteria.
Step 4	Router(config)# policy-map <i>name</i>	Configures the QoS policy for packets that match the class or classes.
Step 5	Router(config-p-map)# class <i>class_name</i>	Associates the traffic class with the service policy.
Step 6	Router(config-p-map-c)# bandwidth <i>{bandwidth_kbps percent percent}</i>	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 <i>type slot/port</i>	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 <i>name</i>	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
```




Configuring PFC QoS Statistics Data Export

This chapter describes how to configure PFC QoS statistics data export on Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter contains these sections:

- [Understanding PFC QoS Statistics Data Export, page 44-1](#)
- [PFC QoS Statistics Data Export Default Configuration, page 44-2](#)
- [Configuring PFC QoS Statistics Data Export, page 44-2](#)

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 Catalyst 6500 series switch.



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 44-1 shows the PFC QoS statistics data export default configuration.

Table 44-1 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 44-2](#)
- [Enabling PFC QoS Statistics Data Export for a Port, page 44-3](#)
- [Enabling PFC QoS Statistics Data Export for a Named Aggregate Policer, page 44-4](#)
- [Enabling PFC QoS Statistics Data Export for a Class Map, page 44-5](#)
- [Setting the PFC QoS Statistics Data Export Time Interval, page 44-6](#)
- [Configuring PFC QoS Statistics Data Export Destination Host and UDP Port, page 44-7](#)
- [Setting the PFC QoS Statistics Data Export Field Delimiter, page 44-9](#)

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 <i>type</i> ¹ <i>slot/port</i>	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 <i>aggregate_policer_name</i>	Enables PFC QoS statistics data export for a named aggregate policer.
	Router(config)# no mls qos statistics-export aggregate-policer <i>aggregate_policer_name</i>	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 <i>classmap_name</i>	Enables PFC QoS statistics data export for a class map.
	Router(config)# no mls qos statistics-export class-map <i>classmap_name</i>	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.
	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 44-2 lists the supported PFC QoS data export facility and severity parameter values.

Table 44-2 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 44-2 Supported PFC QoS Data Export Facility Parameter Values (continued)

Name	Definition	Name	Definition
lpr	line printer subsystem	local5	reserved for local use
news	netnews subsystem	local6	reserved for local use
uucp	uucp subsystem	local7	reserved for local use

Table 44-3 lists the supported PFC QoS data export severity parameter values.

Table 44-3 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 <i>delimiter_character</i>	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
```




Configuring the Cisco IOS Firewall Feature Set

This chapter describes how to configure the Cisco IOS firewall feature set on the Catalyst 6500 series switches. This chapter contains these sections:

- [Cisco IOS Firewall Feature Set Support Overview, page 45-1](#)
- [Cisco IOS Firewall Guidelines and Restrictions, page 45-2](#)
- [Additional CBAC Configuration, page 45-3](#)

Cisco IOS Firewall Feature Set Support Overview

The firewall feature set images support these Cisco IOS firewall features:

- Context-Based Access Control (CBAC) —The PFC installs entries in the Netflow table to direct flows that require CBAC to the MSFC where the CBAC is applied in software on the MSFC.
- Authentication Proxy—After authentication on the MSFC, the PFC provides TCAM support for the authentication policy.
- Port-to-Application Mapping (PAM)—PAM is done in software on the MSFC.

For more information about Cisco IOS firewall features, refer to the following publications:

- *Cisco IOS Security Configuration Guide*, Release 12.2, “Traffic Filtering and Firewalls” chapter and these sections:
 - “Cisco IOS Firewall Overview” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scffiw1.htm
 - “Configuring Context-Based Access Control” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scfcac.htm
 - “Configuring Authentication Proxy” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scfahp.htm
- *Cisco IOS Security Command Reference* publication at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_r/index.htm

The following features are supported with and without the use of a Cisco IOS firewall image:

- Standard access lists and static extended access lists
- Lock-and-key (dynamic access lists)
- IP session filtering (reflexive access lists)
- TCP intercept
- Security server support
- Network address translation
- Neighbor router authentication
- Event logging
- User authentication and authorization

**Note**

Catalyst 6500 series switches support the Intrusion Detection System Module (IDSM) (WS-X6381-IDS). Catalyst 6500 series switches do not support the Cisco IOS firewall IDS feature, which is configured with the **ip audit** command.

Cisco IOS Firewall Guidelines and Restrictions

When configuring the Cisco IOS firewall features, follow these guidelines and restrictions:

- On other platforms, if you enter the **ip inspect** command on a port, CBAC modifies ACLs on other ports to permit the inspected traffic to flow through the network device. On Catalyst 6500 series switches, you must enter the **mls ip inspect** command to permit traffic through any ACLs that would deny the traffic through other ports. Refer to the [“Additional CBAC Configuration” section on page 45-3](#) for more information.
- Reflexive ACLs and CBAC have conflicting flow mask requirements. Reflexive ACLs are processed in software on the MSFC.
- CBAC is incompatible with VACLs. You can configure CBAC and VACLs on the switch but not in the same subnet (VLAN).

**Note**

The Intrusion Detection System Module (IDSM) uses VACLs to select traffic. To use the IDSM in a subnet where CBAC is configured, enter the **mls ip ids acl_name** interface command, where *acl_name* is configured to select traffic for the IDSM.

- To inspect Microsoft NetMeeting (2.0 or greater) traffic, turn on both **h323** and **tcp** inspection.
- To inspect web traffic, turn on **tcp** inspection. To avoid reduced performance, do not turn on **http** inspection to block Java.
- QoS and CBAC do not interact or interfere with each other.
- You can configure CBAC on physical ports configured as Layer 3 interfaces and on VLAN interfaces.
- You cannot configure VACLs and CBAC on the same interface.

Additional CBAC Configuration

You need to do additional CBAC configuration on the Catalyst 6500 series switches. On a network device other than a Catalyst 6500 series switch, when ports are configured to deny traffic, CBAC permits traffic to flow bidirectionally through the port if it is configured with the **ip inspect** command. The same situation applies to any other port that the traffic needs to go through, as shown in this example:

```
Router(config)# ip inspect name permit_ftp ftp
Router(config)# interface vlan 100
Router(config-if)# ip inspect permit_ftp in
Router(config-if)# ip access-group deny_ftp_a in
Router(config-if)# ip access-group deny_ftp_b out
Router(config-if)# exit
Router(config)# interface vlan 200
Router(config-if)# ip access-group deny_ftp_c in
Router(config-if)# ip access-group deny_ftp_d out
Router(config-if)# exit
Router(config)# interface vlan 300
Router(config-if)# ip access-group deny_ftp_e in
Router(config-if)# ip access-group deny_ftp_f out
Router(config-if)# end
```

If the FTP session enters on VLAN 100 and needs to leave on VLAN 200, CBAC permits the FTP traffic through ACLs deny_ftp_a, deny_ftp_b, deny_ftp_c, and deny_ftp_d. If another FTP session enters on VLAN 100 and needs to leave on VLAN 300, CBAC permits the FTP traffic through ACLs deny_ftp_a, deny_ftp_b, deny_ftp_e, and deny_ftp_f.

On a Catalyst 6500 series switch, when ports are configured to deny traffic, CBAC permits traffic to flow bidirectionally only through the port configured with the **ip inspect** command. You must configure other ports with the **mls ip inspect** command.

If the FTP session enters on VLAN 100 and needs to leave on VLAN 200, CBAC on a Catalyst 6500 series switch permits the FTP traffic only through ACLs deny_ftp_a and deny_ftp_b. To permit the traffic through ACLs deny_ftp_c and deny_ftp_d, you must enter the **mls ip inspect deny_ftp_c** and **mls ip inspect deny_ftp_d** commands, as shown in this example:

```
Router(config)# mls ip inspect deny_ftp_c
Router(config)# mls ip inspect deny_ftp_d
```

FTP traffic cannot leave on VLAN 300 unless you enter the **mls ip inspect deny_ftp_e** and **mls ip inspect deny_ftp_f** commands. Enter the **show fm insp [detail]** command to verify the configuration.

The **show fm insp [detail]** command displays the list of ACLs and ports on which CBAC is configured and the status (**ACTIVE** or **INACTIVE**), as shown in this example:

```
Router# show fm insp
      interface:Vlan305(in) status :ACTIVE
      acl name:deny
      interfaces:
        Vlan305(out):status ACTIVE
```

On VLAN 305, inspection is active in the inbound direction and no ACL exists. ACL **deny** is applied on VLAN 305 in the outbound direction and inspection is active.

To display all of the flow information, use the **detail** keyword.

If a VACL is configured on the port before configuring CBAC, the status displayed is **INACTIVE**; otherwise, it is **ACTIVE**. If PFC resources are exhausted, the command displays the word “**BRIDGE**” followed by the number of currently active NetFlow requests that failed, which have been sent to the MSFC for processing.



Configuring IEEE 802.1X Port-Based Authentication

This chapter describes how to configure IEEE 802.1X port-based authentication to prevent unauthorized devices (clients) from gaining access to the network.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding 802.1X Port-Based Authentication, page 46-1](#)
- [Default 802.1X Port-Based Authentication Configuration, page 46-5](#)
- [802.1X Port-Based Authentication Guidelines and Restrictions, page 46-6](#)
- [Configuring 802.1X Port-Based Authentication, page 46-7](#)
- [Displaying 802.1X Status, page 46-15](#)

Understanding 802.1X Port-Based Authentication

The IEEE 802.1X standard defines a client-server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a switch port and assigns the port to a VLAN before making available any services offered by the switch or the LAN.

Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

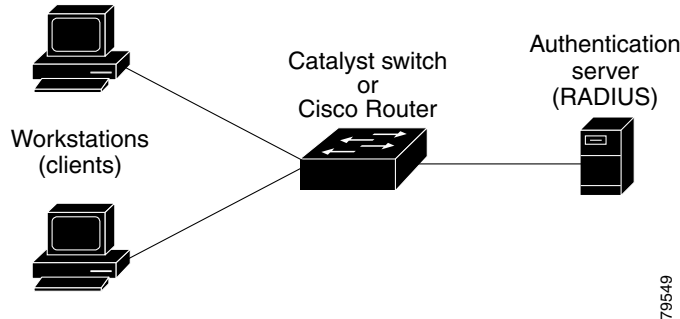
These sections describe IEEE 802.1X port-based authentication:

- [Device Roles, page 46-2](#)
- [Authentication Initiation and Message Exchange, page 46-3](#)
- [Ports in Authorized and Unauthorized States, page 46-4](#)
- [Supported Topologies, page 46-4](#)

Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles as shown in Figure 46-1.

Figure 46-1 802.1X Device Roles



The specific roles shown in Figure 46-1 are as follows:

- **Client**—The device (workstation) that requests access to the LAN and switch services and responds to requests from the switch. The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating system. (The client is the *supplicant* in the IEEE 802.1X specification.)



Note

To resolve Windows XP network connectivity and 802.1X port-based authentication issues, read the Microsoft Knowledge Base article at this URL:

<http://support.microsoft.com/support/kb/articles/Q303/5/97.ASP>

- **Authentication server**—Performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch whether or not the client is authorized to access the LAN and switch services. Because the switch acts as the proxy, the authentication service is transparent to the client. The Remote Authentication Dial-In User Service (RADIUS) security system with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server, version 3.0. RADIUS uses a client-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.
- **Switch** (also called the *authenticator* and *back-end authenticator*)—Controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client. The switch includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

When the switch receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is reencapsulated in the RADIUS format. The EAP frames are not modified or examined during encapsulation, and the authentication server must support EAP within the native frame format. When the switch receives frames from the authentication server, the server's frame header is removed, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client.

Authentication Initiation and Message Exchange

The switch or the client can initiate authentication. If you enable authentication on a port by using the **dot1x port-control auto** interface configuration command, the switch must initiate authentication when it determines that the port link state transitions from down to up. The switch then sends an EAP-request/identity frame to the client to request its identity (typically, the switch sends an initial identity/request frame followed by one or more requests for authentication information). When the client receives the frame, it responds with an EAP-response/identity frame.

If the client does not receive an EAP-request/identity frame from the switch during bootup, the client can initiate authentication by sending an EAPOL-start frame, which prompts the switch to request the client's identity.



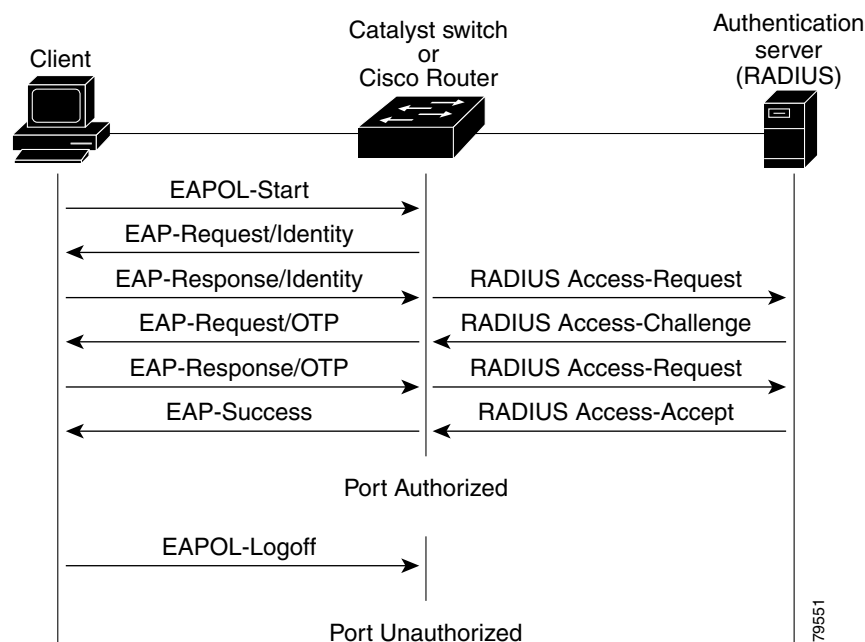
Note

If 802.1X is not enabled or supported on the network access device, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client transmits frames as if the port is in the authorized state. A port in the authorized state effectively means that the client has been successfully authenticated. For more information, see the [“Ports in Authorized and Unauthorized States”](#) section on page 46-4.

When the client supplies its identity, the switch begins its role as the intermediary, passing EAP frames between the client and the authentication server until authentication succeeds or fails. If the authentication succeeds, the switch port becomes authorized. For more information, see the [“Ports in Authorized and Unauthorized States”](#) section on page 46-4.

The specific exchange of EAP frames depends on the authentication method being used. [Figure 46-2](#) shows a message exchange initiated by the client using the One-Time-Password (OTP) authentication method with a RADIUS server.

Figure 46-2 Message Exchange



Ports in Authorized and Unauthorized States

The switch port state determines whether or not the client is granted access to the network. The port starts in the *unauthorized* state. While in this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a client is successfully authenticated, the port transitions to the *authorized* state, allowing all traffic for the client to flow normally.

If a client that does not support 802.1X is connected to an unauthorized 802.1X port, the switch requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

You control the port authorization state by using the **dot1x port-control** interface configuration command and these keywords:

- **force-authorized**—Disables 802.1X port-based authentication and causes the port to transition to the authorized state without any authentication exchange required. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This is the default setting.
- **force-unauthorized**—Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The switch cannot provide authentication services to the client through the interface.
- **auto**—Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received. The switch requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each client attempting to access the network is uniquely identified by the switch by using the client's MAC address.

If the client is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated client are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the switch can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and network access is not granted.

When a client logs off, it sends an EAPOL-logoff message, causing the switch port to transition to the unauthorized state.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

Supported Topologies

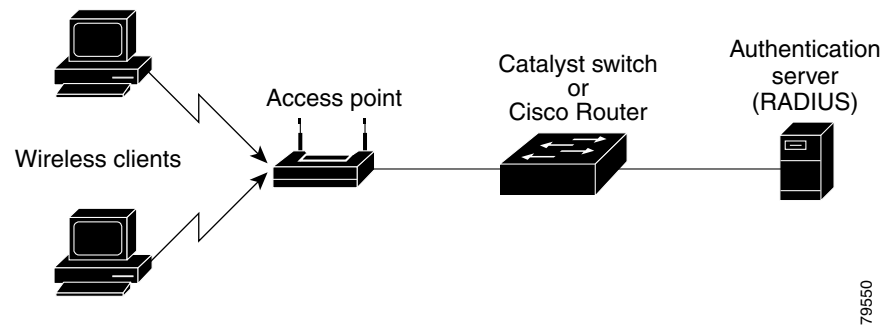
The 802.1X port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration (see [Figure 46-1 on page 46-2](#)), only one client can be connected to the 802.1X-enabled switch port. The switch detects the client when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

[Figure 46-3](#) shows 802.1X port-based authentication in a wireless LAN. The 802.1X port is configured as a multiple-host port that becomes authorized as soon as one client is authenticated. When the port is authorized, all other hosts indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the switch denies access to the network to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and the wireless access point acts as a client to the switch.

Figure 46-3 Wireless LAN Example



Default 802.1X Port-Based Authentication Configuration

[Table 46-1](#) shows the default 802.1X configuration.

Table 46-1 Default 802.1X Configuration

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled
RADIUS server IP address	None specified
RADIUS server UDP authentication port	1812
RADIUS server key	None specified
Per-interface 802.1X protocol enable state	Disabled (force-authorized) Note The port transmits and receives normal traffic without 802.1X-based authentication of the client.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet period	60 seconds (number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client)

Table 46-1 Default 802.1X Configuration (continued)

Feature	Default Setting
Retransmission time	30 seconds (number of seconds that the switch should wait for a response to an EAP request/identity frame from the client before retransmitting the request)
Maximum retransmission number	2 times (number of times that the switch will send an EAP-request/identity frame before restarting the authentication process)
Multiple host support	Disabled
Client timeout period	30 seconds (when relaying a request from the authentication server to the client, the amount of time the switch waits for a response before retransmitting the request to the client)
Authentication server timeout period	30 seconds (when relaying a response from the client to the authentication server, the amount of time the switch waits for a reply before retransmitting the response to the server)

802.1X Port-Based Authentication Guidelines and Restrictions

When configuring 802.1X port-based authentication, follow these guidelines and restrictions:

- When 802.1X is enabled, ports are authenticated before any other Layer 2 or Layer 3 features are enabled.
- The 802.1X protocol is supported on both Layer 2 static-access ports and Layer 3 routed ports, but it is not supported on these port types:
 - Trunk port—If you try to enable 802.1X on a trunk port, an error message appears, and 802.1X is not enabled. If you try to change the mode of an 802.1X-enabled port to trunk, the port mode is not changed.
 - EtherChannel port—Before enabling 802.1X on the port, you must first remove it from the EtherChannel port-channel interface. If you try to enable 802.1X on an EtherChannel port-channel interface or on an individual active port in an EtherChannel, an error message appears, and 802.1X is not enabled. If you enable 802.1X on a not-yet active individual port of an EtherChannel, the port does not join the EtherChannel.
 - Secure port—You cannot configure a secure port as an 802.1X port. If you try to enable 802.1X on a secure port, an error message appears, and 802.1X is not enabled. If you try to change an 802.1X-enabled port to a secure port, an error message appears, and the security settings are not changed.
 - Switch Port Analyzer (SPAN) destination port—You can enable 802.1X on a port that is a SPAN destination port; however, 802.1X is disabled until the port is removed as a SPAN destination port. You can enable 802.1X on a SPAN source port.

Configuring 802.1X Port-Based Authentication

These sections describe how to configure 802.1X port-based authentication:

- [Enabling 802.1X Port-Based Authentication, page 46-7](#)
- [Configuring Switch-to-RADIUS-Server Communication, page 46-8](#)
- [Enabling Periodic Reauthentication, page 46-10](#)
- [Manually Reauthenticating the Client Connected to a Port, page 46-11](#)
- [Initializing Authentication for the Client Connected to a Port, page 46-11](#)
- [Changing the Quiet Period, page 46-11](#)
- [Changing the Switch-to-Client Retransmission Time, page 46-12](#)
- [Setting the Switch-to-Client Frame Retransmission Number, page 46-14](#)
- [Enabling Multiple Hosts, page 46-14](#)
- [Resetting the 802.1X Configuration to the Default Values, page 46-15](#)

Enabling 802.1X Port-Based Authentication

To enable 802.1X port-based authentication, you must enable AAA and specify the authentication method list. A method list describes the sequence and authentication methods to be queried to authenticate a user.

The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle, the authentication process stops, and no other authentication methods are attempted.

To configure 802.1X port-based authentication, perform this task:

	Command	Purpose
Step 1	Router(config)# aaa new-model	Enables AAA.
	Router(config)# no aaa new-model	Disables AAA.
Step 2	Router(config)# aaa authentication dot1x {default} method1 [method2...]	Creates an 802.1X port-based authentication method list.
	Router(config)# no aaa authentication dot1x {default list_name}	Clears the configured method list.
Step 3	Router(config)# dot1x system-auth-control	Globally enables 802.1X port-based authentication.
	Router(config)# no dot1x system-auth-control	Globally disables 802.1X port-based authentication.
Step 4	Router(config)# interface type ¹ slot/port	Enters interface configuration mode and specifies the interface to be enabled for 802.1X port-based authentication.
Step 5	Router(config-if)# dot1x port-control auto	Enables 802.1X port-based authentication on the interface.
	Router(config-if)# no dot1x port-control auto	Disables 802.1X port-based authentication on the interface.

	Command	Purpose
Step 6	Router(config)# end	Returns to privileged EXEC mode.
Step 7	Router# show dot1x all	Verifies your entries. Check the Status column in the 802.1X Port Summary section of the display. An <i>enabled</i> status means the port-control value is set either to auto or to force-unauthorized .

1. *type* = **ethernet**, **fastethernet**, **gigabithernet**, or **tengigabithernet**

When you enable 802.1X port-based authentication, note the following information:

- To create a default list that is used when a named list is *not* specified in the **authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces.
- Enter at least one of these keywords:
 - **group radius**—Use the list of all RADIUS servers for authentication.
 - **none**—Use no authentication. The client is automatically authenticated by the switch without using the information supplied by the client.

This example shows how to enable AAA and 802.1X on Fast Ethernet port 5/1:

```
Router# configure terminal
Router(config)# aaa new-model
Router(config)# aaa authentication dot1x default group radius
Router(config)# dot1x system-auth-control
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# end
```

This example shows how to verify the configuration:

```
Router# show dot1x all
```

```
Dot1x Info for interface FastEthernet5/1
```

```
-----
AuthSM State      = FORCE UNAUTHORIZED
BendSM State      = IDLE
PortStatus        = UNAUTHORIZED
MaxReq            = 2
MultiHosts        = Disabled
Port Control      = Force Unauthorized
QuietPeriod       = 60 Seconds
Re-authentication = Disabled
ReAuthPeriod      = 3600 Seconds
ServerTimeout     = 30 Seconds
SuppTimeout       = 30 Seconds
TxPeriod          = 30 Seconds
```

Configuring Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by any of the following:

- Host name
- Host IP address

- Host name and specific UDP port numbers
- IP address and specific UDP port numbers

The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service (for example, authentication) the second host entry configured acts as the failover backup to the first one. The RADIUS host entries are tried in the order that they were configured.

To configure the RADIUS server parameters, perform this task:

	Command	Purpose
Step 1	Router(config)# ip radius source-interface <i>interface_name</i>	Specifies that the RADIUS packets have the IP address of the indicated interface.
	Router(config)# no ip radius source-interface	Prevents the RADIUS packets from having the IP address of the previously indicated interface.
Step 2	Router(config)# radius-server host { <i>hostname</i> <i>ip_address</i> }	Configures the RADIUS server host name or IP address on the switch. If you want to use multiple RADIUS servers, reenter this command.
	Router(config)# no radius-server host { <i>hostname</i> <i>ip_address</i> }	Deletes the specified RADIUS server.
Step 3	Router(config)# radius-server key <i>string</i>	Configures the authorization and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 4	Router(config)# end	Returns to privileged EXEC mode.

When you configure the RADIUS server parameters, note the following information:

- For *hostname* or *ip_address*, specify the host name or IP address of the remote RADIUS server.
- Specify the **key string** on a separate command line.
- For **key string**, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server.
- When you specify the **key string**, spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon.
- You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius-server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and the **radius-server key** global configuration commands. For more information, refer to the *Cisco IOS Security Configuration Guide*, Release 12.2, publication and the *Cisco IOS Security Command Reference*, Release 12.2, publication at this URL:

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



Note

You also need to configure some settings on the RADIUS server. These settings include the IP address of the switch and the key string to be shared by both the server and the switch. For more information, refer to the RADIUS server documentation.

This example shows how to configure the RADIUS server parameters on the switch:

```
Router# configure terminal
Router(config)# ip radius source-interface Vlan80
Router(config)# radius-server host 172.120.39.46
Router(config)# radius-server key rad123
Router(config)# end
```

Enabling Periodic Reauthentication

You can enable periodic 802.1X client reauthentication and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication attempts is 3600.

Automatic 802.1X client reauthentication is a global setting and cannot be set for clients connected to individual ports. To manually reauthenticate the client connected to a specific port, see the [“Manually Reauthenticating the Client Connected to a Port” section on page 46-11](#).

To enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x reauthentication	Enables periodic reauthentication of the client, which is disabled by default.
	Router(config-if)# no dot1x reauthentication	Disables periodic reauthentication of the client.
Step 3	Router(config-if)# dot1x timeout re-authperiod <i>seconds</i>	Sets the number of seconds between reauthentication attempts. The range is 1 to 4294967295; the default is 3600 seconds. This command affects the behavior of the switch only if periodic reauthentication is enabled.
	Router(config-if)# no dot1x timeout re-authperiod	Returns to the default reauthorization period.
	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

This example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000:

```
Router(config-if)# dot1x reauthentication
Router(config-if)# dot1x timeout re-authperiod 4000
```


Manually Reauthenticating the Client Connected to a Port



Note

Reauthentication does not disturb the status of an already authorized port.

To manually reauthenticate the client connected to a port, perform this task:

	Command	Purpose
Step 1	Router# dot1x re-authenticate interface <i>type</i> ¹ <i>slot/port</i>	Manually reauthenticates the client connected to a port.
Step 2	Router# show dot1x all	Verifies your entries.

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

This example shows how to manually reauthenticate the client connected to Fast Ethernet port 5/1:

```
Router# dot1x re-authenticate interface fastethernet 5/1
Starting reauthentication on FastEthernet 5/1
```

Initializing Authentication for the Client Connected to a Port



Note

Initializing authentication disables any existing authentication before authenticating the client connected to the port.

To initialize the authentication for the client connected to a port, perform this task:

	Command	Purpose
Step 1	Router# dot1x initialize interface <i>type</i> ¹ <i>slot/port</i>	Initializes the authentication for the client connected to a port.
Step 2	Router# show dot1x all	Verifies your entries.

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

This example shows how to initialize the authentication for the client connected to Fast Ethernet port 5/1:

```
Router# dot1x initialize interface fastethernet 5/1
Starting reauthentication on FastEthernet 5/1
```

Changing the Quiet Period

When the switch cannot authenticate the client, the switch remains idle for a set period of time, and then tries again. The idle time is determined by the quiet-period value. A failed authentication of the client might occur because the client provided an invalid password. You can provide a faster response time to the user by entering a smaller number than the default.

To change the quiet period, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout quiet-period <i>seconds</i>	Sets the number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client. The range is 0 to 65535 seconds; the default is 60.
	Router(config-if)# no dot1x timeout quiet-period	Returns to the default quiet time.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

This example shows how to set the quiet time on the switch to 30 seconds:

```
Router(config-if)# dot1x timeout quiet-period 30
```

Changing the Switch-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the switch with an EAP-response/identity frame. If the switch does not receive this response, it waits a set period of time (known as the retransmission time), and then retransmits the frame.



Note

You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

To change the amount of time that the switch waits for client notification, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout tx-period <i>seconds</i>	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request. The range is 1 to 65535 seconds; the default is 30.
	Router(config-if)# dot1x timeout tx-period	Returns to the default retransmission time.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

This example shows how to set 60 as the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request:

```
Router(config)# dot1x timeout tx-period 60
```

Setting the Switch-to-Client Retransmission Time for EAP-Request Frames

The client notifies the switch that it received the EAP-request frame. If the switch does not receive this notification, the switch waits a set period of time, and then retransmits the frame. You may set the amount of time that the switch waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the switch-to-client retransmission time for the EAP-request frames, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout supp-timeout <i>seconds</i> Router(config-if)# no dot1x timeout supp-timeout	Sets the switch-to-client retransmission time for the EAP-request frame. Returns to the default retransmission time.
Step 3	Router# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

This example shows how to set the switch-to-client retransmission time for the EAP-request frame to 25 seconds:

```
Router(config-if)# dot1x timeout supp-timeout 25
```

Setting the Switch-to-Authentication-Server Retransmission Time for Layer 4 Packets

The authentication server notifies the switch each time it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the switch waits a set period of time and then retransmits the packet. You may set the amount of time that the switch waits for notification from 1 to 65535 seconds. (The default is 30 seconds.)

To set the value for the retransmission of Layer 4 packets from the switch to the authentication server, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x timeout server-timeout <i>seconds</i> Router(config-if)# no dot1x timeout server-timeout	Sets the switch-to-authentication-server retransmission time for Layer 4 packets. Returns to the default retransmission time.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

This example shows how to set the switch-to-authentication-server retransmission time for Layer 4 packets to 25 seconds:

```
Router(config-if)# dot1x timeout server-timeout 25
```

Setting the Switch-to-Client Frame Retransmission Number

In addition to changing the switch-to-client retransmission time, you can change the number of times that the switch sends an EAP-request/identity frame (assuming no response is received) to the client before restarting the authentication process.



Note

You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

To set the switch-to-client frame retransmission number, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x max-req <i>count</i>	Sets the number of times that the switch sends an EAP-request/identity frame to the client before restarting the authentication process. The range is 1 to 10; the default is 2.
	Router(config-if)# no dot1x max-req	Returns to the default retransmission number.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

This example shows how to set 5 as the number of times that the switch sends an EAP-request/identity request before restarting the authentication process:

```
Router(config-if)# dot1x max-req 5
```

Enabling Multiple Hosts

You can attach multiple hosts to a single 802.1X-enabled port as shown in [Figure 46-3 on page 46-5](#). In this mode, only one of the attached hosts must be successfully authorized for all hosts to be granted network access. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), all attached clients are denied access to the network.

To allow multiple hosts (clients) on an 802.1X-authorized port that has the **dot1x port-control** interface configuration command set to **auto**, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x host-mode multi-host	Allows multiple hosts (clients) on an 802.1X-authorized port.
	Router(config-if)# dot1x host-mode single-host	Disables multiple hosts on the port.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x interface <i>type</i> ¹ <i>slot/port</i>	Verifies your entries.

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

This example shows how to enable 802.1X on Fast Ethernet interface 5/1 and to allow multiple hosts:

```
Router(config)# interface fastethernet 5/1
Router(config-if)# dot1x port-control auto
Router(config-if)# dot1x multi-hosts
```

Resetting the 802.1X Configuration to the Default Values

To reset the 802.1X configuration to the default values, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects an interface to configure.
Step 2	Router(config-if)# dot1x default	Resets the configurable 802.1X parameters to the default values.
Step 3	Router(config-if)# end	Returns to privileged EXEC mode.
Step 4	Router# show dot1x all	Verifies your entries.

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

Displaying 802.1X Status

To display global 802.1X administrative and operational status for the switch, use the **show dot1x** privileged EXEC command. To display the 802.1X administrative and operational status for a specific interface, use the **show dot1x interface interface-id** privileged EXEC command.

For detailed information about the fields in these displays, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX.



Configuring Port Security

This chapter describes how to configure the port security feature.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding Port Security, page 47-1](#)
- [Default Port Security Configuration, page 47-2](#)
- [Port Security Guidelines and Restrictions, page 47-3](#)
- [Configuring Port Security, page 47-3](#)
- [Displaying Port Security Settings, page 47-11](#)

Understanding Port Security

These sections describe port security:

- [Port Security with Dynamically Learned and Static MAC Addresses, page 47-1](#)
- [Port Security with Sticky MAC Addresses, page 47-2](#)

Port Security with Dynamically Learned and Static MAC Addresses

You can use the port security with dynamically learned and static MAC addresses to restrict ingress traffic to an interface by limiting the MAC addresses that are allowed to send traffic into a port. When you assign secure MAC addresses to a secure port, the port does not forward packets with 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 workstation attached to that port has the full bandwidth of the port.

If a port is configured as a secure port and the maximum number of secure MAC addresses is reached, when the source MAC address of ingress traffic is different from any of the identified secure MAC addresses, a security violation occurs. If traffic with a secure MAC address that is configured or learned on one secure port attempts to access another secure port, a violation occurs.

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 47-3](#).

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

Release 12.2(18)SXE and later releases support 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 47-1](#) shows the default port security configuration for an interface.

Table 47-1 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 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.
- With Release 12.2(18)SXE and later releases, port security supports private VLAN (PVLAN) ports.
- With releases earlier than Release 12.2(18)SXE, port security does not support PVLAN ports.
- With Release 12.2(18)SXE and later releases, 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.

- With releases earlier than Release 12.2(18)SXE, port security does not support trunks.
- With Release 12.2(18)SXE and later releases, port security supports IEEE 802.1Q tunnel ports.
- With releases earlier than Release 12.2(18)SXE, port security does not support 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, page 47-4](#)
- [Configuring the Port Security Violation Mode on a Port, page 47-6](#)

- [Configuring the Maximum Number of Secure MAC Addresses on a Port, page 47-7](#)
- [Enabling Port Security with Sticky MAC Addresses on a Port, page 47-8](#)
- [Configuring a Static Secure MAC Address on a Port, page 47-9](#)
- [Configuring Secure MAC Address Aging on a Port, page 47-10](#)

Enabling Port Security

When enabling port security on access ports or trunks, follow these guidelines:

- When you enable port security, if an address learned or configured on one secure port is seen on another secure port in the same VLAN, port security puts the port into the error-disabled state immediately.
- 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 reenabling it by entering the **shutdown** and **no shutdown** interface configuration commands.
- Enter the **clear port-security dynamic** global configuration command to clear all dynamically learned secure addresses. See the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX, for complete syntax information.

These sections describe how to enable port security:

- [Enabling Port Security on a Trunk, page 47-4](#)
- [Enabling Port Security on an Access Port, page 47-5](#)

Enabling Port Security on a Trunk

With Release 12.2(18)SXE and later releases, 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 47-7](#)).

To enable port security on a trunk, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	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 <i>type</i> ¹ <i>slot/port</i> 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 <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure. Note With Release 12.2(18)SX and later releases, 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 <i>type</i> ¹ <i>slot/port</i> 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 <i>type</i> ¹ <i>slot/port</i>	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 <i>type</i> ¹ <i>slot/port</i> 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 reen able 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 <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport port-security maximum <i>number_of_addresses</i> vlan { <i>vlan_ID</i> <i>vlan_range</i> }	Sets the maximum number of secure MAC addresses for the port (default is 1).
	Router(config-if)# no switchport port-security maximum	Reverts to the default configuration.
Step 3	Router(config-if)# do show port-security interface <i>type</i> ¹ <i>slot/port</i> 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:

- With Release 12.2(18)SXE and later releases, the range for *number_of_addresses* is 1 to 4,097.
- With releases earlier than Release 12.2(18)SXE, the range for *number_of_addresses* is 1 to 1,024.
- With Release 12.2(18)SXE and later releases, 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

Release 12.2(18)SXE and later releases support port security with sticky MAC addresses. To enable port security with sticky MAC addresses on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	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 <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport port-security mac-address [sticky] <i>mac_address</i> [vlan <i>vlan_ID</i>] Router(config-if)# no switchport port-security mac-address [sticky] <i>mac_address</i>	Configures a static MAC address as secure on the port. Note Per-VLAN configuration is supported only on trunks. 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:

- With Release 12.2(18)SX and later releases, 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 47-8).
- 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.
- With Release 12.2(18)SX and later releases, 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 any VLAN on the port.

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 47-10](#)
- [Configuring Secure MAC Address Aging Time on a Port, page 47-11](#)

Configuring the Secure MAC Address Aging Type on a Port

With a PFC3 and Release 12.2(18)SX and later releases, you can configure the secure MAC address aging type on a port. With a PFC2, you cannot configure the secure MAC address aging type. The PFC2 supports only absolute aging.

To configure the secure MAC address aging type on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	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 <i>type</i> ¹ <i>slot/port</i> 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 <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# switchport port-security aging time <i>aging_time</i>	Configures the secure MAC address aging time on the port. The <i>aging_time</i> 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 <i>type</i> ¹ <i>slot/port</i> 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 <i>vlan_ID</i> {{ <i>type</i> ¹ <i>slot/port</i> }}}}] [address]	Displays port security settings for the switch or for the specified interface.

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

When displaying port security settings, note the following information:

- Port security supports the **vlan** keyword only on trunks.
- Enter the **address** keyword to display secure MAC addresses, with aging information for each address, globally for the switch or per interface.
- The display includes these values:
 - The maximum allowed number of secure MAC addresses for each interface
 - The number of secure MAC addresses on the interface
 - The number of security violations that have occurred
 - The violation mode.

This example displays output from the **show port-security** command when you do not enter an interface:

```
Router# show port-security
Secure Port      MaxSecureAddr  CurrentAddr  SecurityViolation  Security
Action
                (Count)          (Count)      (Count)
-----
      Fa5/1         11             11           0             Shutdown
      Fa5/5         15             5            0             Restrict
      Fa5/11        5              4            0             Protect
-----

Total Addresses in System: 21
Max Addresses limit in System: 128
```

This example displays output from the **show port-security** command for a specified interface:

```
Router# show port-security interface fastethernet 5/1
Port Security: Enabled
Port status: SecureUp
Violation mode: Shutdown
Maximum MAC Addresses: 11
Total MAC Addresses: 11
Configured MAC Addresses: 3
Aging time: 20 mins
Aging type: Inactivity
SecureStatic address aging: Enabled
Security Violation count: 0
```

This example displays the output from the **show port-security address** privileged EXEC command:

```
Router# show port-security address
      Secure Mac Address Table
-----
Vlan    Mac Address      Type                Ports    Remaining Age
      (mins)
-----
      1    0001.0001.0001    SecureDynamic        Fa5/1     15 (I)
      1    0001.0001.0002    SecureDynamic        Fa5/1     15 (I)
      1    0001.0001.1111    SecureConfigured     Fa5/1     16 (I)
      1    0001.0001.1112    SecureConfigured     Fa5/1      -
      1    0001.0001.1113    SecureConfigured     Fa5/1      -
      1    0005.0005.0001    SecureConfigured     Fa5/5     23
      1    0005.0005.0002    SecureConfigured     Fa5/5     23
      1    0005.0005.0003    SecureConfigured     Fa5/5     23
      1    0011.0011.0001    SecureConfigured     Fa5/11    25 (I)
      1    0011.0011.0002    SecureConfigured     Fa5/11    25 (I)
-----

Total Addresses in System: 10
Max Addresses limit in System: 128
```



Configuring CDP

This chapter contains information about how to configure Cisco Discovery Protocol (CDP) on the Catalyst 6500 series switches, which supplements the information in these publications:

- The *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2, “System Management,” “Configuring Cisco Discovery Protocol (CDP)” at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/fcfprt3/fcf015.htm
- The *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.2, “System Management Commands,” “CDP Commands” publication at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_r/ffrprt3/frf015.htm

This chapter consists of these sections:

- [Understanding How CDP Works, page 48-1](#)
- [Configuring CDP, page 48-1](#)

Understanding How CDP Works

CDP is a protocol that runs over Layer 2 (the data link layer) on all Cisco routers, bridges, access servers, and switches. CDP allows network management applications to discover Cisco devices that are neighbors of already known devices, in particular, neighbors running lower-layer, transparent protocols. With CDP, network management applications can learn the device type and the SNMP agent address of neighboring devices. This feature enables applications to send SNMP queries to neighboring devices.

CDP runs on all LAN and WAN media that support Subnetwork Access Protocol (SNAP).

Each CDP-configured device sends periodic messages to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain the time-to-live, or holdtime information, which indicates the length of time a receiving device should hold CDP information before discarding it.

Configuring CDP

These sections describe how to configure CDP:

- [Enabling CDP Globally, page 48-2](#)
- [Displaying the CDP Global Configuration, page 48-2](#)
- [Enabling CDP on a Port, page 48-2](#)

- [Displaying the CDP Interface Configuration, page 48-3](#)
- [Monitoring and Maintaining CDP, page 48-3](#)

Enabling CDP Globally

To enable CDP globally, perform this task:

Command	Purpose
Router(config)# cdp run	Enables CDP globally.
Router(config)# no cdp run	Disables CDP globally.

This example shows how to enable CDP globally:

```
Router(config)# cdp run
```

Displaying the CDP Global Configuration

To display the CDP configuration, perform this task:

Command	Purpose
Router# show cdp	Displays global CDP information.

This example shows how to display the CDP configuration:

```
Router# show cdp
Global CDP information:
  Sending CDP packets every 120 seconds
  Sending a holdtime value of 180 seconds
  Sending CDPv2 advertisements is enabled
Router#
```

For additional CDP show commands, see the [“Monitoring and Maintaining CDP”](#) section on page 48-3.

Enabling CDP on a Port

To enable CDP on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# interface {{type ¹ slot/port} {port-channel number}}	Selects the port to configure.
Step 2	Router(config-if)# cdp enable	Enables CDP on the port.
	Router(config-if)# no cdp enable	Disables CDP on the port.

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

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

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

Displaying the CDP Interface Configuration

To display the CDP configuration for a port, perform this task:

Command	Purpose
Router# show cdp interface [{type ¹ slot/port} {port-channel number}]]	Displays information about ports where CDP is enabled.
1. type = ethernet, fastethernet, gigabitethernet, or tengigabitethernet	

This example shows how to display the CDP configuration of Fast Ethernet port 5/1:

```
Router# show cdp interface fastethernet 5/1
FastEthernet5/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 120 seconds
  Holdtime is 180 seconds
Router#
```

Monitoring and Maintaining CDP

To monitor and maintain CDP on your device, perform one or more of these tasks:

Command	Purpose
Router# clear cdp counters	Resets the traffic counters to zero.
Router# clear cdp table	Clears information about neighbors from the CDP table.
Router# show cdp	Displays global information such as frequency of transmissions and the holdtime for packets being transmitted.
Router# show cdp entry entry_name [protocol version]	Displays information about a specific neighbor. The display can be limited to protocol or version information.
Router# show cdp interface [type ¹ slot/port]]	Displays information about interfaces on which CDP is enabled.
Router# show cdp neighbors [type ¹ slot/port] [detail]	Displays information about neighbors. The display can be limited to neighbors on a specific interface and expanded to provide more detailed information.
Router# show cdp traffic	Displays CDP counters, including the number of packets sent and received and checksum errors.
Router# show debugging	Displays information about the types of debugging that are enabled. Refer to the <i>Debug Command Reference</i> for more information about CDP debug commands.

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

This example shows how to clear CDP counter configuration:

```
Router# clear cdp counters
```

This example shows how to display information about the neighboring equipment:

```
Router# show cdp neighbors
```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
JAB023807H1	Fas 5/3	127	T S	WS-C2948	2/46
JAB023807H1	Fas 5/2	127	T S	WS-C2948	2/45
JAB023807H1	Fas 5/1	127	T S	WS-C2948	2/44
JAB023807H1	Gig 1/2	122	T S	WS-C2948	2/50
JAB023807H1	Gig 1/1	122	T S	WS-C2948	2/49
JAB03130104	Fas 5/8	167	T S	WS-C4003	2/47
JAB03130104	Fas 5/9	152	T S	WS-C4003	2/48



Configuring UDLD

This chapter describes how to configure the UniDirectional Link Detection (UDLD) protocol on the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How UDLD Works, page 49-1](#)
- [Default UDLD Configuration, page 49-3](#)
- [Configuring UDLD, page 49-3](#)

Understanding How UDLD Works

These sections describe how UDLD works:

- [UDLD Overview, page 49-1](#)
- [UDLD Aggressive Mode, page 49-2](#)

UDLD Overview

The Cisco-proprietary UDLD protocol allows devices connected through fiber-optic or copper (for example, Category 5 cabling) Ethernet cables connected to LAN ports to monitor the physical configuration of the cables and detect when a unidirectional link exists. When a unidirectional link is detected, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems, including spanning tree topology loops.

UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both autonegotiation and UDLD, Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally at Layer 1, then UDLD at Layer 2 determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation, because autonegotiation operates at Layer 1.

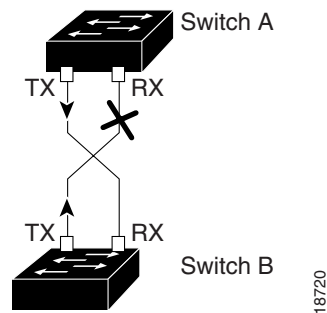
The Catalyst 6500 series switch periodically transmits UDLD packets to neighbor devices on LAN ports with UDLD enabled. If the packets are echoed back within a specific time frame and they are lacking a specific acknowledgment (echo), the link is flagged as unidirectional and the LAN port is shut down. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links.


Note

By default, UDLD is locally disabled on copper LAN ports to avoid sending unnecessary control traffic on this type of media since it is often used for access ports.

Figure 49-1 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 49-1 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 discarded.

Default UDLD Configuration

Table 49-1 shows the default UDLD configuration.

Table 49-1 UDLD Default Configuration

Feature	Default Value
UDLD global enable state	Globally disabled
UDLD aggressive mode	Disabled
UDLD per-port enable state for fiber-optic media	Enabled on all Ethernet fiber-optic LAN ports
UDLD per-port enable state for twisted-pair (copper) media	Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports

Configuring UDLD

These sections describe how to configure UDLD:

- [Enabling UDLD Globally, page 49-3](#)
- [Enabling UDLD on Individual LAN Interfaces, page 49-4](#)
- [Disabling UDLD on Fiber-Optic LAN Interfaces, page 49-4](#)
- [Configuring the UDLD Probe Message Interval, page 49-5](#)
- [Resetting Disabled LAN Interfaces, page 49-5](#)

Enabling UDLD Globally

To enable UDLD globally on all fiber-optic LAN ports, perform this task:

Command	Purpose
Router(config)# udld {enable aggressive}	Enables UDLD globally on fiber-optic LAN ports. Note This command only configures fiber-optic LAN ports. Individual LAN port configuration overrides the setting of this command.
Router(config)# no udld {enable aggressive}	Disables UDLD globally on fiber-optic LAN ports.

Enabling UDLD on Individual LAN Interfaces

To enable UDLD on individual LAN ports, perform this task:

	Command	Purpose
Step 1	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# udld port [aggressive] Router(config-if)# no 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. 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 <i>type</i> ¹ <i>slot/number</i>	Verifies the configuration.
	1. <i>type</i> = 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 <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 2	Router(config-if)# udld port disable Router(config-if)# no udld port disable	Disables UDLD on a fiber-optic LAN port. 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 <i>type</i> ¹ <i>slot/number</i>	Verifies the configuration.
	1. <i>type</i> = 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 type¹ slot/number	Verifies the configuration.
	1. <i>type</i> = 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.



Configuring NetFlow and NDE

This chapter describes how to configure NetFlow statistics collection and NetFlow Data Export (NDE) on the Catalyst 6500 series switches.



Note

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

- The *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- The Release 12.2 publications at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/index.htm>
- Release 12.2(18)SXF and later releases support NetFlow version 9—See this publication:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s24/nfv9expf.htm>
- Release 12.2(18)SXF and later releases have NetFlow multicast support with the NetFlow v9 export format feature:
 - See this publication:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123_1/nfmultic.htm
 - The NetFlow Multicast Support document contains a prerequisite that does not apply when configuring NetFlow multicast support with Release 12.2(18)SXF and later 12.2SX releases:
You do not need to configure multicast fast switching or multicast distributed fast switching (MDFS); multicast CEF switching is supported with Release 12.2(18)SXF and later 12.2SX releases.

This chapter consists of these sections:

- [Understanding How NetFlow and NDE Work](#), page 50-2
- [Default NetFlow and NDE Configuration](#), page 50-8
- [NetFlow and NDE Configuration Guidelines and Restrictions](#), page 50-9
- [Configuring NetFlow and NDE](#), page 50-9

Understanding How NetFlow and NDE Work

These sections describe how NetFlow and NDE work:

- [NetFlow and NDE Overview, page 50-2](#)
- [NetFlow and NDE on the MSFC, page 50-2](#)
- [NetFlow and NDE on the PFC, page 50-2](#)

NetFlow and NDE Overview

NetFlow collects statistics globally from traffic that flows through the switch and stores the statistics in the NetFlow table. In PFC3B or PFC3BXL mode with Release 12.2(18)SXE and later releases, you can configure NDE to collect statistics for both routed and bridged traffic. In PFC3A mode or with releases earlier than Release 12.2(18)SXE, NDE collects statistics only for routed traffic.

Two NetFlow options reduce the volume of statistics collected:

- Sampled NetFlow, which reduces the number of statistics collected
- NetFlow aggregation, which merges collected statistics



Note NetFlow aggregation uses NDE version 8.

NDE makes routed-traffic statistics available for analysis by an external data collector. You can use NDE for all IP unicast traffic. NDE exports global statistics.

You can configure two external data collector addresses, which improves the probability of receiving complete NetFlow data by providing redundant data streams with these releases and hardware:

- A PFC2 and Release 12.2(18)SXD and later releases
- A PFC3 and Release 12.2(18)SXE and later releases

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, refer to this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt2/index.htm

The MSFC supports NetFlow ToS-based router aggregation, described at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s15/dtnfltoss.htm>

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, page 50-3](#)

- [NDE Versions, page 50-3](#)
- [MLS Cache Entries, page 50-7](#)
- [Sampled NetFlow, page 50-7](#)
- [NetFlow Aggregation, page 50-8](#)

Flow Masks

The PFC uses one of these 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.

NetFlow uses only one flow mask for all statistics.

NDE Versions



Note

Release 12.2(18)SXF and later releases support NetFlow version 9—See this publication:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s24/nfv9expf.htm>

NDE on the PFC supports NDE versions 5 and 7 for the statistics captured on the PFC. NetFlow aggregation uses version 8.

Some fields in the flow records might not have values, depending on the current flow mask. Unsupported fields contain a zero (0).

The following tables describe the supported NDE fields:

- [Table 50-1](#)—Version 5 header format
- [Table 50-2](#)—Version 5 flow record format
- [Table 50-3](#)—Version 7 header format
- [Table 50-4](#)—Version 7 flow record format

Table 50-1 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 50-2 NDE Version 5 Flow Record Format

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field (see the “Populating Additional NDE Fields” section on page 50-15)					
			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	0	A ¹	A	A	A	A
12–13	input	Ingress interface SNMP ifIndex	0	0	0	X	0	X
14–15	output	Egress interface SNMP ifIndex	0	A ¹	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 ²	X ²
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 ³	0	0	0	0	0	0
38	prot	Layer 4 protocol (for example, 6=TCP, 17=UDP)	0	0	0	0	X	X
39	tos	IP type-of-service byte	X ⁴	X ⁴	X ⁴	X ⁴	X ⁴	X ⁴
40–41	src_as	Autonomous system number of the source, either origin or peer	X	0	X	X	X	X

Table 50-2 NDE Version 5 Flow Record Format (continued)

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field (see the “Populating Additional NDE Fields” section on page 50-15)					
			Source	Destination	Destination Source	Destination Source Interface	Full	Full Interface
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. 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.
2. In PFC3BXL or PFC3B mode, for ICMP traffic, contains the ICMP code and type values.
3. Always zero for hardware-switched flows.
4. Populated in PFC3BXL or PFC3B mode.

Table 50-3 NDE Version 7 Header Format

Bytes	Content	Description
0–1	version	NetFlow export format version number
2–3	count	Number of flows exported in this packet (1–30)
4–7	SysUptime	Current time in milliseconds since router booted
8–11	unix_secs	Current seconds since 0000 UTC 1970
12–15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970
16–19	flow_sequence	Sequence counter of total flows seen
20–23	reserved	Unused (zero) bytes

Table 50-4 NDE Version 7 Flow Record Format

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field (see the “Populating Additional NDE Fields” section on page 50-15)					
			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

Table 50-4 NDE Version 7 Flow Record Format (continued)

Bytes	Content	Description	Flow masks: • X=Populated • A=Additional field (see the “Populating Additional NDE Fields” section on page 50-15)					
			Source	Destination	Destination Source	Destination Source Interface	Full	Full Interface
8–11	nexthop	Next hop router’s IP address	0	A ¹	A	A	A	A
12–13	input	Ingress interface SNMP ifIndex	0	0	0	X	0	X
14–15	output	Egress interface SNMP ifIndex	0	A ¹	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 ²	X ²
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 ³	0	0	0	0	0	0
38	prot	Layer 4 protocol (for example, 6=TCP, 17=UDP)	0	0	0	0	X	X
39	tos	IP type-of-service byte	X ⁴	X ⁴	X ⁴	X ⁴	X ⁴	X ⁴
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. 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.
2. In PFC3BXL or PFC3B mode, for ICMP traffic, contains the ICMP code and type values.
3. Always zero for hardware-switched flows.
4. Populated with Release 12.2(17b)SXA and later releases in PFC3BXL or PFC3B mode.

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 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.

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 50-19](#) for NDE filter configuration procedures.

Sampled NetFlow

On Layer 3 interfaces, the Sampled NetFlow feature captures data for a subset of traffic in a flow instead of for all packets in the flow. Sampled NetFlow substantially decreases the supervisor engine CPU utilization.

With a Supervisor Engine 720, sampled NetFlow always uses the full-interface flow mask. With a Supervisor Engine 2, sampled NetFlow uses the full-interface or destination-source-interface flow masks.

With either the full-interface or destination-source-interface flow masks, you can enable or disable sampled NetFlow on each Layer 3 interface.

With a Supervisor Engine 2 and other flow masks, sampled Netflow is enabled or disabled globally.

You can configure sampled NetFlow to use time-based sampling or packet-based sampling.

[Table 50-5](#) lists the time-based sampling rates and export intervals.

Table 50-5 Time-Based Sampling Rates, Sampling Times, and Export Intervals

Sampling Rate	Sampling Time (Milliseconds)	Export Interval (Milliseconds)
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

Table 50-5 Time-Based Sampling Rates, Sampling Times, and Export Intervals

Sampling Rate	Sampling Time (Milliseconds)	Export Interval (Milliseconds)
1 in 2048	4	8192
1 in 4096	4	16384
1 in 8192	4	32768

If you configure 64 as the rate, the sampled NetFlow feature uses traffic from the first 64 milliseconds of a flow every 4096 milliseconds. If the rate is 2048, the sampled NetFlow feature uses traffic from the first 4 milliseconds of a flow every 8192 milliseconds. With time-based sampled NetFlow, the export interval is not configurable.

Packet-based sampled NetFlow uses this formula to sample a flow: the number of times sampled is approximately the length divided by the rate (*packets_in_flow/sampling_rate*). For example, if the flow is 32,768 packets long and the sampling rate is 1024, the flow is sampled approximately 32 times (32,768/1,024). With packet-based sampled NetFlow, the export interval is configurable.

NetFlow Aggregation

For complete information about NetFlow aggregation as supported on the PFC and DFCs, refer to this publication:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt2/xcfnfov.htm#1001212

Default NetFlow and NDE Configuration

Table 50-6 shows the default NetFlow and NDE configuration.

Table 50-6 Default NetFlow and NDE Configuration

Feature	Default Value
NetFlow	Disabled
NDE	Disabled
NetFlow and NDE of ingress bridged IP traffic	Disabled
NDE source addresses	None
NDE data collector address and UDP port	None
NDE filters	None
Sampled NetFlow	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.
- In PFC3B or PFC3BXL mode with Release 12.2(18)SXE and later releases, NetFlow and NDE support bridged IP traffic. PFC3A mode does not support NetFlow and NDE for bridged IP traffic.
- NDE does not support Internetwork Packet Exchange (IPX) traffic.
- The Policy Feature Card 3 (PFC3) and Policy Feature Card 2 (PFC) do 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
PFC3BXL	235,520 (230K) entries	262,144 entries
PFC3B	117,760 (115K) entries	131,072 entries
PFC3A	65,536 (64K) entries	131,072 entries
PFC2	32,768 (32K) entries	65,536 entries

- No statistics are available for flows that are switched when the NetFlow table is full.

Configuring NetFlow and NDE

These sections describe how to configure NetFlow and NDE:

- [Configuring NetFlow and NDE on the PFC, page 50-10](#)
- [Configuring NetFlow and NDE on the MSFC, page 50-15](#)
- [Enabling NetFlow and NDE for Ingress Bridged IP Traffic, page 50-17](#)
- [Displaying the NDE Address and Port Configuration, page 50-18](#)
- [Configuring NDE Flow Filters, page 50-19](#)
- [Displaying the NDE Configuration, page 50-21](#)



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 on the PFC

These sections describe how to configure NetFlow and NDE on the PFC:

- [Configuring NetFlow on the PFC, page 50-10](#)
- [Configuring NDE on the PFC, page 50-14](#)

Configuring NetFlow on the PFC

These sections describe how to configure NetFlow statistics collection on the PFC:

- [Enabling NetFlow on the PFC, page 50-10](#)
- [Configuring Sampled NetFlow, page 50-10](#)
- [Configuring NetFlow Aggregation on the PFC, page 50-11](#)
- [Setting the Minimum IP MLS Flow Mask, page 50-12](#)
- [Configuring the MLS Aging Time, page 50-13](#)

Enabling NetFlow on the PFC

To enable NetFlow statistics collection on the PFC, perform this task:

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
```

Configuring Sampled NetFlow

These sections describe how to configure sampled NetFlow on the PFC:

- [Configuring Sampled NetFlow Globally, page 50-11](#)
- [Configuring Sampled NetFlow on a Layer 3 Interface, page 50-11](#)



Note

NDE on the MSFC does not support sampled NetFlow.

Configuring Sampled NetFlow Globally

To configure sampled NetFlow globally, perform this task:

	Command	Purpose
Step 1	Router(config)# mls sampling { time-based rate packet-based rate [<i>interval</i>]}	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.

See the “[Sampled NetFlow](#)” section on page 50-7 for more information.

Configuring Sampled NetFlow on a Layer 3 Interface



Note

- 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.
- The Layer 3 interface must be configured with an IP address.

To configure sampled NetFlow on a Layer 3 interface, perform this task:

	Command	Purpose
Step 1	Router(config)# interface { vlan <i>vlan_ID</i> <i>type slot/port</i> }	Selects a Layer 3 interface to configure.
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 on the PFC

NetFlow aggregation is configured automatically on the PFC and DFCs when you configure NetFlow aggregation on the MSFC (see the “[Configuring NetFlow Aggregation on the MSFC](#)” section on page 50-16).

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.

**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 aged polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
Dst If Dst Prefix Msk AS Flows Pkts B/Pk Active
Gi7/9 9.1.0.0 /16 0 3003 12M 64 1699.8
Gi7/10 11.1.0.0 /16 0 3000 9873K 64 1699.8
Router#
```

This example shows how to display the NetFlow aggregation flow mask information:

```
Router# show mls netflow aggregation flowmask
Current flowmask set for netflow aggregation : Vlan Full Flow
Netflow aggregations configured/enabled :
    AS Aggregation
    PROTOCOL-PORT Aggregation
    SOURCE-PREFIX Aggregation
    DESTINATION-PREFIX Aggregation
Router#
```

Setting the Minimum IP MLS Flow Mask

You can set the minimum specificity of the flow mask for the NetFlow cache on the PFC (see the [“Flow Masks” section on page 50-3](#)). The actual flow mask that is used will have at least the specificity configured by the **mls flow ip** command.

To set the minimum IP MLS flow mask, perform this task:

Command	Purpose
Router(config)# mls flow ip {source destination destination-source interface-destination-source full interface-full}	Sets the minimum IP MLS flow mask for the protocol.
Router(config)# no mls flow ip	Reverts to the default IP MLS flow mask (null).

This example shows how to set the minimum IP MLS flow mask:

```
Router(config)# mls flow ip destination
```


To display the IP MLS flow mask configuration, perform this task:

Command	Purpose
Router# show mls netflow flowmask	Displays the flow mask configuration.

This example shows how to display the MLS flow mask configuration:

```
Router# show mls netflow flowmask
current ip flowmask for unicast: destination address
Router#
```

Configuring the MLS Aging Time

The MLS aging time (default 300 seconds) applies to all NetFlow 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 50-9), 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 shortcut entries.
- **fast aging**—Configures an efficient process to age out entries created for flows that only switch a few packets, and then are never used again. The **fast aging** parameter uses the **time** keyword value to check if at least the **threshold** keyword value of packets have been switched for each flow. If a flow has not switched the threshold number of packets during the time interval, then the entry is aged out.
- **long**—Configures entries for deletion that have been up for the specified value even if the entry is 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 the MLS aging time, perform this task:

Command	Purpose
Router(config)# mls aging {fast [threshold {1-128} time {1-128}] long 64-1920 normal 32-4092}	Configures the MLS aging time for a NetFlow 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 the MLS aging time:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# mls aging fast threshold 64 time 30
```

To display the MLS aging-time configuration, perform this task:

Command	Purpose
Router# show mls netflow aging	Displays the MLS aging-time configuration.

This example shows how to display the MLS aging-time configuration:

```
Router# show mls netflow aging
enable timeout packet threshold
-----
normal aging true 300 N/A
fast aging true 32 100
long aging true 900 N/A
```

Configuring NDE on the PFC

- [Enabling NDE From the PFC, page 50-14](#)
- [Populating Additional NDE Fields, page 50-15](#)

Enabling NDE From the PFC

To enable NDE from the PFC, perform this task:

Command	Purpose
Router(config)# mls nde sender [version {5 7}]	Enables NDE from the PFC and (optionally) configures the NDE version.
Router(config)# no mls nde sender	Disables NDE from the PFC.
Router(config)# no mls nde sender version	Reverts to the default (version 7).



Note

- NDE from the PFC uses the source interface configured for the MSFC (see the [“Configuring the MSFC NDE Source Layer 3 Interface”](#) section on page 50-16).
- Release 12.2(18)SXF and later releases support NetFlow version 9—See this publication: <http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s24/nfv9expf.htm>

This example shows how to enable NDE from the PFC:

```
Router(config)# mls nde sender
```

This example shows how to enable NDE from the PFC and configure NDE version 5:

```
Router(config)# mls nde sender version 5
```

Populating Additional NDE Fields

You can configure NDE to populate the following additional fields in the NDE packets:

- IP address of the next hop router
- Egress interface SNMP ifIndex

Not all of the additional fields are populated with all flow masks. See the “NDE Versions” section on page 50-3 for additional information.

To populate the additional fields in NDE packets, perform this task:

Command	Purpose
Router(config)# mls nde interface	Populates additional fields in NDE packets.
Router(config)# no mls nde interface	Disables population of the additional fields.

This example shows how to populate the additional fields in NDE packets:

```
Router(config)# mls nde interface
```

Configuring NetFlow and NDE on the MSFC

This section supplements the NetFlow procedures at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/switch_r/index.htm

These sections describe how to configure NDE on the MSFC:

- [Enabling NetFlow on the MSFC, page 50-15](#)
- [Configuring NetFlow Aggregation on the MSFC, page 50-16](#)
- [Configuring the MSFC NDE Source Layer 3 Interface, page 50-16](#)
- [Configuring the NDE Destination, page 50-16](#)

Enabling NetFlow on the MSFC

To enable NetFlow on the MSFC, perform this task for each Layer 3 interface from which you want NDE:

	Command	Purpose
Step 1	Router(config)# interface { vlan <i>vlan_ID</i> } { <i>type slot/port</i> } { port-channel <i>port_channel_number</i> }	Selects a Layer 3 interface to configure.
Step 2	Router(config-if)# ip flow-export ingress ¹ Router(config-if)# ip route-cache flow ²	Enables NetFlow.

1. Supported in Release 12.2(18)SXD and later releases.

2. Deprecated in Release 12.2(18)SXD.

Configuring NetFlow Aggregation on the MSFC

To configure NetFlow aggregation on the MSFC, use the procedures at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_c/swprt2/xcfnfc.htm#1001058

To configure NetFlow ToS-based router aggregation on the MSFC, use the procedures at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s15/dtnfltoss.htm>



Note

- When you configure NetFlow aggregation on the MSFC, it is configured automatically on the PFC and DFCs (see the “Configuring NetFlow Aggregation on the PFC” section on page 50-11).
- 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 <i>vlan_ID</i> } { type <i>slot/port</i> } { port-channel <i>number</i> } { loopback <i>number</i> }}	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 <i>ip_address</i> <i>udp_port_number</i>	Configures the NDE destination IP address and UDP port.
Router(config)# no ip flow-export destination <i>ip_address udp_port_number</i>	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 and these releases support the Netflow Multiple Export Destinations feature:

- PFC3 and Release 12.2(18)SXE and later releases
- PFC2 and Release 12.2(18)SXD and later releases

This example shows how to configure the NDE flow destination IP address and UDP port:

```
Router(config)# ip flow-export destination 172.20.52.37 200
```

**Note**

The destination address and UDP port number are saved in NVRAM and are preserved if NDE is disabled and reenabled or if the switch is power cycled. If you are using the NetFlow FlowCollector application for data collection, verify that the UDP port number you configure is the same port number shown in the FlowCollector's /opt/csconfc/config/nfconfig.file file.

Enabling NetFlow and NDE for Ingress Bridged IP Traffic

In PFC3B or PFC3BXL mode with Release 12.2(18)SXE and later releases, NetFlow and NDE support ingress bridged IP traffic. PFC3A mode does not support NetFlow and NDE for bridged IP traffic. These sections describe how to enable NetFlow and NDE for ingress bridged IP traffic:

- [Enabling NetFlow for Ingress Bridged IP Traffic in VLANs, page 50-17](#)
- [Enabling NDE for Ingress Bridged IP Traffic in VLANs, page 50-18](#)

**Note**

- When you enable NetFlow for ingress bridged IP traffic, the statistics are available to the Sampled Netflow feature (see the [“Sampled NetFlow” section on page 50-7](#)).
- You must configure an IP address on the VLAN interface of each VLAN where you want to enable NetFlow and NDE.

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 <i>vlan_ID</i> [- <i>vlan_ID</i>] [, <i>vlan_ID</i> [- <i>vlan_ID</i>]]	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 <i>vlan_ID</i> [- <i>vlan_ID</i>] [, <i>vlan_ID</i> [- <i>vlan_ID</i>]]	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 <i>vlan_ID</i> [- <i>vlan_ID</i>] [<i>,</i> <i>vlan_ID</i> [- <i>vlan_ID</i>]]	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).
Router(config)# no ip flow export layer2-switched vlan <i>vlan_ID</i> [- <i>vlan_ID</i>] [<i>,</i> <i>vlan_ID</i> [- <i>vlan_ID</i>]]	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, page 50-19](#)
- [Configuring a Port Flow Filter, page 50-20](#)
- [Configuring a Host and Port Filter, page 50-20](#)
- [Configuring a Host Flow Filter, page 50-20](#)
- [Configuring a Protocol Flow Filter, page 50-21](#)

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 50-21](#).

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 <i>number</i> src-port <i>number</i> }	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 35
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 <i>ip_address mask</i> source <i>ip_address mask</i> { dest-port <i>number</i> src-port <i>number</i> }}	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 exclude destination 2.2.2.2 255.255.255.0 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 <i>ip_address mask</i> source <i>ip_address mask</i> protocol { tcp { dest-port <i>number</i> src-port <i>number</i> } udp { dest-port <i>number</i> src-port <i>number</i> }}	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 include and export only destinations to host 172.20.52.37:

```
Router(config)# mls nde flow include destination 172.20.52.37 255.255.255.224
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 <i>number</i> src-port <i>number</i> } udp { dest-port <i>number</i> src-port <i>number</i> }}	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 50-21](#).

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#
```




Configuring Local SPAN, RSPAN, and ERSPAN

This chapter describes how to configure local Switched Port Analyzer (SPAN), remote SPAN (RSPAN), and Encapsulated RSPAN (ERSPAN) on the Catalyst 6500 series switches. Release 12.2(18)SX and later releases support ERSPAN with Supervisor Engine 720 (see the “[ERSPAN Guidelines and Restrictions](#)” section on page 51-11).



Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- OSM WAN ports and FlexWAN ports do not support SPAN, RSPAN or ERSPAN.
- Supervisor Engine 2 does not support ERSPAN.

This chapter consists of these sections:

- [Understanding How Local SPAN, RSPAN, and ERSPAN Work](#), page 51-1
- [Local SPAN, RSPAN, and ERSPAN Configuration Guidelines and Restrictions](#), page 51-6
- [Configuring Local SPAN, RSPAN, and ERSPAN](#), page 51-12

Understanding How Local SPAN, RSPAN, and ERSPAN Work

These sections describe how local SPAN, RSPAN, and ERSPAN work:

- [Local SPAN, RSPAN, and ERSPAN Overview](#), page 51-2
- [Local SPAN, RSPAN, and ERSPAN Sources](#), page 51-5
- [Local SPAN, RSPAN, and ERSPAN Destination Ports](#), page 51-6

Local SPAN, RSPAN, and ERSPAN Overview

Local SPAN, RSPAN, and ERSPAN sessions allow you to monitor traffic on one or more ports, or one or more VLANs, and send the monitored traffic to one or more destination ports. With Release 12.2(18)SXD and later releases, you can configure per-VLAN filtering on destination trunk ports.

Local SPAN, RSPAN, and ERSPAN all send traffic to a network analyzer such as a SwitchProbe device or other Remote Monitoring (RMON) probe. SPAN does not affect the switching of traffic on source ports or VLANs. SPAN sends a copy of the packets received or transmitted by the source ports and VLANs to the destination port. You must dedicate the destination port for SPAN use.

These sections provide an overview of local SPAN, RSPAN, and ERSPAN:

- [Local SPAN Overview, page 51-2](#)
- [RSPAN Overview, page 51-3](#)
- [ERSPAN Overview, page 51-4](#)
- [Monitored Traffic, page 51-4](#)

Local SPAN Overview

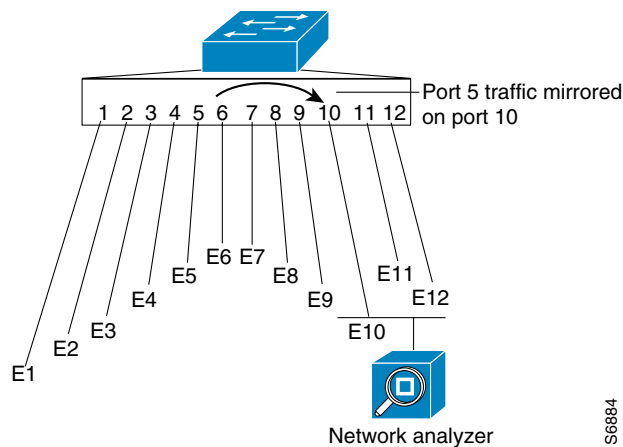
A local SPAN session is an association of source ports and source VLANs with one or more destination ports. You configure a local SPAN session on a single switch. Local SPAN does not have separate source and destination sessions.

Local SPAN sessions do not copy locally sourced RSPAN VLAN traffic from source trunk ports that carry RSPAN VLANs. Local SPAN sessions do not copy locally sourced RSPAN GRE-encapsulated traffic from source ports.

Each local SPAN session can have either ports or VLANs as sources, but not both.

Local SPAN copies traffic from one or more source ports in any VLAN or from one or more VLANs to a destination port for analysis (see [Figure 51-1](#)). For example, as shown in [Figure 51-1](#), 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 51-1 Example SPAN Configuration



56884

RSPAN Overview

RSPAN supports source ports, source VLANs, and destination ports on different switches, which provides remote monitoring of multiple switches across your network (see [Figure 51-2](#)).

RSPAN consists of an RSPAN source session, an RSPAN VLAN, and an RSPAN destination session. You separately configure RSPAN source sessions and destination sessions on different switches. To configure an RSPAN source session on one switch, you associate a set of source ports or VLANs with an RSPAN VLAN. To configure an RSPAN destination session on another switch, you associate the destination ports with the RSPAN VLAN.

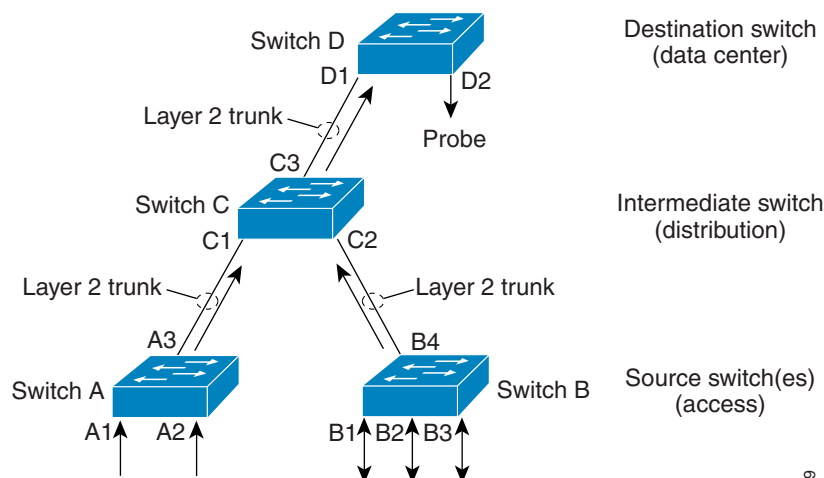
The traffic for each RSPAN session is carried as Layer 2 nonroutable traffic over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches. All participating switches must be trunk-connected at Layer 2.

RSPAN source sessions do not copy locally sourced RSPAN VLAN traffic from source trunk ports that carry RSPAN VLANs. RSPAN source sessions do not copy locally sourced RSPAN GRE-encapsulated traffic from source ports.

Each RSPAN source session can have either ports or VLANs as sources, but not both.

The RSPAN source session copies traffic from the source ports or source VLANs and switches the traffic over the RSPAN VLAN to the RSPAN destination session. The RSPAN destination session switches the traffic to the destination ports.

Figure 51-2 RSPAN Configuration



27389

ERSPAN Overview

ERSPAN supports source ports, source VLANs, and destination ports on different switches, which provides remote monitoring of multiple switches across your network (see [Figure 51-3](#)).

ERSPAN consists of an ERSPAN source session, routable ERSPAN GRE-encapsulated traffic, and an ERSPAN destination session. You separately configure ERSPAN source sessions and destination sessions on different switches.

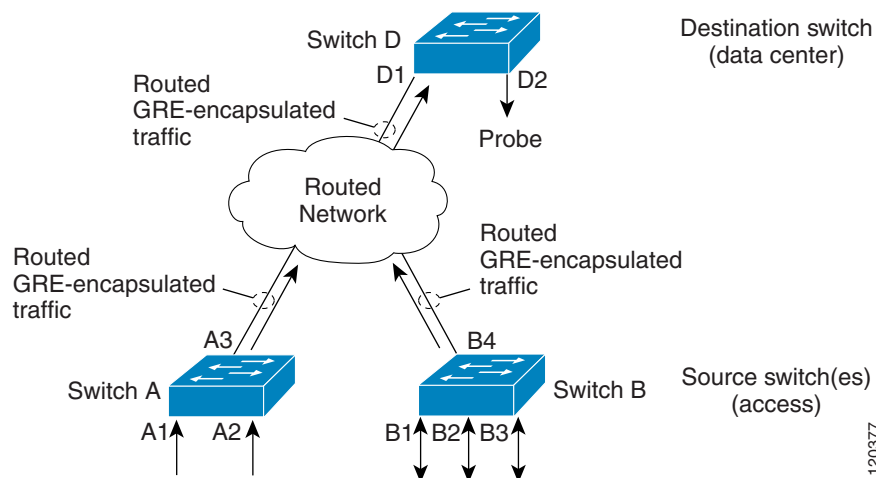
To configure an ERSPAN source session on one switch, you associate a set of source ports or VLANs with a destination IP address, ERSPAN ID number, and optionally with a VRF name. To configure an ERSPAN destination session on another switch, you associate the destination ports with the source IP address, ERSPAN ID number, and optionally with a VRF name.

ERSPAN source sessions do not copy locally sourced RSPAN VLAN traffic from source trunk ports that carry RSPAN VLANs. ERSPAN source sessions do not copy locally sourced ERSPAN GRE-encapsulated traffic from source ports.

Each ERSPAN source session can have either ports or VLANs as sources, but not both.

The ERSPAN source session copies traffic from the source ports or source VLANs and forwards the traffic using routable GRE-encapsulated packets to the ERSPAN destination session. The ERSPAN destination session switches the traffic to the destination ports.

Figure 51-3 ERSPAN Configuration



Monitored Traffic

These sections describe the traffic that local SPAN, RSPAN, and ERSPAN can monitor:

- [Monitored Traffic Direction, page 51-5](#)
- [Monitored Traffic, page 51-5](#)
- [Duplicate Traffic, page 51-5](#)

Monitored Traffic Direction

You can configure local SPAN sessions, RSPAN source sessions, and ERSPAN source sessions to monitor ingress traffic (called ingress SPAN), or to monitor egress traffic (called egress SPAN), or to monitor traffic flowing in both directions.

Ingress SPAN copies traffic received by the source ports and VLANs for analysis at the destination port. Egress SPAN copies traffic transmitted from the source ports and VLANs. When you enter the **both** keyword, SPAN copies the traffic received and transmitted by the source ports and VLANs to the destination port.

Monitored Traffic

By default, local SPAN and ERSPAN monitor all traffic, including multicast and bridge protocol data unit (BPDU) frames. RSPAN does not support BPDU monitoring.

Duplicate Traffic

In some configurations, SPAN sends multiple copies of the same source traffic to the destination port. For example, in a configuration with a bidirectional SPAN session (both ingress and egress) for two SPAN sources, called s1 and s2, to a SPAN destination port, called d1, if a packet enters the switch through s1 and is sent for egress from the switch to s2, ingress SPAN at s1 sends a copy of the packet to SPAN destination d1 and egress SPAN at s2 sends a copy of the packet to SPAN destination d1. If the packet was Layer 2 switched from s1 to s2, both SPAN packets would be the same. If the packet was Layer 3 switched from s1 to s2, the Layer 3 rewrite would alter the source and destination Layer 2 addresses, in which case the SPAN packets would be different.

Local SPAN, RSPAN, and ERSPAN Sources

These sections describe local SPAN, RSPAN, and ERSPAN sources:

- [Source Ports, page 51-5](#)
- [Source VLANs, page 51-5](#)

Source Ports

A source port is a port monitored for traffic analysis. You can configure both switched and routed ports as SPAN source ports. SPAN can monitor one or more source ports in a single SPAN session. You can configure source ports in any VLAN. Trunk ports can be configured as source ports and mixed with nontrunk source ports. SPAN does not copy the encapsulation from a source trunk port.

Source VLANs

A source VLAN is a VLAN monitored for traffic analysis. VLAN-based SPAN (VSPAN) uses a VLAN as the SPAN source. All the ports in the source VLANs become source ports.

Local SPAN, RSPAN, and ERSPAN Destination Ports

A destination port is a Layer 2 or Layer 3 LAN port to which local SPAN, RSPAN, or ERSPAN sends traffic for analysis.

When you configure a port as a destination port, it can no longer receive any traffic. When you configure a port as a destination port, the port is dedicated for use only by the SPAN feature. A SPAN destination port does not forward any traffic except that required for the SPAN session.

You can configure trunk ports as destination ports, which allows destination trunk ports to transmit encapsulated traffic. With Release 12.2(18)SXD and later releases, for local SPAN, you can configure per-VLAN filtering on destination trunk ports using allowed VLAN lists (see the [“Configuring Destination Trunk Port VLAN Filtering”](#) section on page 51-23).

Local SPAN, RSPAN, and ERSPAN Configuration Guidelines and Restrictions

These sections describe local SPAN, RSPAN, and ERSPAN configuration guidelines and restrictions:

- [Feature Incompatibilities, page 51-6](#)
- [Local SPAN, RSPAN, and ERSPAN Session Limits, page 51-7](#)
- [Local SPAN, RSPAN, and ERSPAN Guidelines and Restrictions, page 51-8](#)
- [VSPAN Guidelines and Restrictions, page 51-10](#)
- [RSPAN Guidelines and Restrictions, page 51-10](#)
- [ERSPAN Guidelines and Restrictions, page 51-11](#)



Note

Release 12.2(18)SXE and later releases support ERSPAN.

Feature Incompatibilities

These feature incompatibilities exist with local SPAN, RSPAN, and ERSPAN:

- With a PFC3, EoMPLS ports cannot be SPAN sources. (CSCed51245)
- A port-channel interface (an EtherChannel) can be a SPAN source, but you cannot configure active member ports of an EtherChannel as SPAN source ports. Inactive member ports of an EtherChannel can be configured as SPAN sources but they are put into the suspended state and carry no traffic.
- A port-channel interface (an EtherChannel) cannot be a SPAN destination.
- You cannot configure active member ports of an EtherChannel as SPAN destination ports. Inactive member ports of an EtherChannel can be configured as SPAN destination ports but they are put into the suspended state and carry no traffic.
- Because SPAN destination ports drop ingress traffic, these features are incompatible with SPAN destination ports:
 - Private VLANs
 - IEEE 802.1X port-based authentication
 - Port security

- Spanning tree protocol (STP) and related features (PortFast, PortFast BPDU Filtering, BPDU Guard, UplinkFast, BackboneFast, EtherChannel Guard, Root Guard, Loop Guard)
- VLAN trunk protocol (VTP)
- Dynamic trunking protocol (DTP)
- IEEE 802.1Q tunneling


Note

SPAN destination ports can participate in IEEE 802.3Z Flow Control.

Local SPAN, RSPAN, and ERSPAN Session Limits

These sections describe the local SPAN, RSPAN, and ERSPAN session limits:

- [Supervisor Engine 720, page 51-7](#)
- [Supervisor Engine 2, page 51-8](#)

Supervisor Engine 720

These are the Supervisor Engine 720 local SPAN, RSPAN, and ERSPAN session limits:

Total Sessions	Local SPAN, RSPAN Source, or ERSPAN Source Sessions	RSPAN Destination Sessions	ERSPAN Destination Sessions
66	2 (ingress or egress or both)	64	23

These are the Supervisor Engine 720 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				—	—
With releases earlier than Release 12.2(18)SXE	1	1	1		
Release 12.2(18)SXE and later releases	128	128	128		
Ingress sources				—	—
With releases earlier than Release 12.2(18)SXD	64	64	64		
Release 12.2(18)SXD and later releases	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

Supervisor Engine 2



Note

- Supervisor Engine 2 does not support RSPAN if you configure an egress SPAN source for a local SPAN session.
- Supervisor Engine 2 does not support egress SPAN sources for local SPAN if you configure RSPAN.

These are the Supervisor Engine 2 local SPAN and RSPAN session limits:

Total Sessions	Local SPAN Sessions	RSPAN Source Sessions	RSPAN Destination Sessions
66	2 (ingress or egress or both)	0	64
	1 ingress	1 (ingress or egress or both)	64
	1 or 2 egress	0	64

These are the Supervisor Engine 2 local SPAN and RSPAN source and destination limits:

	In Each Local SPAN Session	In Each RSPAN Source Session	In Each RSPAN Destination Session
Egress or “both” sources	1 (0 with a remote SPAN source session configured)	1 (0 with a local SPAN egress source session configured)	—
Ingress sources			—
With releases earlier than Release 12.2(18)SXD	64	64	
Release 12.2(18)SXD and later releases	128	128	
RSPAN destination session source	—	—	1 RSPAN VLAN
Destinations per session	64	1 RSPAN VLAN	64

Local SPAN, RSPAN, and ERSPAN Guidelines and Restrictions

These guidelines and restrictions apply to local SPAN, RSPAN, and ERSPAN:

- A SPAN destination port that is copying traffic from a single egress SPAN source port sends only egress traffic to the network analyzer. However, in Release 12.2(18)SXE and later releases, if you configure more than one egress SPAN source port, the traffic that is sent to the network analyzer also includes these types of ingress traffic that were received from the egress SPAN source ports:
 - Any unicast traffic that is flooded on the VLAN
 - Broadcast and multicast traffic

This situation occurs because an egress SPAN source port receives these types of traffic from the VLAN but then recognizes itself as the source of the traffic and drops it instead of sending it back to the source from which it was received. Before the traffic is dropped, SPAN copies the traffic and sends it to the SPAN destination port. (CSCds22021)

- Entering additional **monitor session** commands does not clear previously configured SPAN parameters. You must enter the **no monitor session** command to clear configured SPAN parameters.
- Connect a network analyzer to the SPAN destination ports.
- All the SPAN destination ports receive all of the traffic from all the SPAN sources.



Note With Release 12.2(18)SXD and later releases, you can configure destination trunk port VLAN filtering using allowed VLAN lists (see the [“Configuring Destination Trunk Port VLAN Filtering”](#) section on page 51-23).

With Release 12.2(18)SXE and later releases, for local SPAN and RSPAN, you can configure Source VLAN Filtering (see the [“Configuring Source VLAN Filtering for Local SPAN and RSPAN”](#) section on page 51-22).

- You can configure both Layer 2 LAN ports (LAN ports configured with the **switchport** command) and Layer 3 LAN ports (LAN ports not configured with the **switchport** command) as sources or destinations.
- You cannot mix individual source ports and source VLANs within a single session.
- If you specify multiple ingress source ports, the ports can belong to different VLANs.
- You cannot mix source VLANs and filter VLANs within a session. You can have source VLANs or filter VLANs, but not both at the same time.
- When enabled, local SPAN, RSPAN, and ERSPAN use any previously entered configuration.
- When you specify sources and do not specify a traffic direction (ingress, egress, or both), “both” is used by default.
- SPAN copies Layer 2 Ethernet frames, but SPAN does not copy source trunk port ISL or 802.1Q tags. You can configure destination ports as trunks to send locally tagged traffic to the traffic analyzer.



Note A destination port configured as a trunk tags traffic from a Layer 3 LAN source port with the internal VLAN used by the Layer 3 LAN port.

- Local SPAN sessions, RSPAN source sessions, and ERSPAN source sessions do not copy locally sourced RSPAN VLAN traffic from source trunk ports that carry RSPAN VLANs.
- Local SPAN sessions, RSPAN source sessions, and ERSPAN source sessions do not copy locally sourced ERSPAN GRE-encapsulated traffic from source ports.
- A port specified as a destination port in one SPAN session cannot be a destination port for another SPAN session.
- A port configured as a destination port cannot be configured as a source port.
- Destination ports never participate in any spanning tree instance. Local SPAN includes BPDUs in the monitored traffic, so any BPDUs seen on the destination port are from the source port. RSPAN does not support BPDU monitoring.
- All packets sent through the switch for transmission from a port configured as an egress source are copied to the destination port, including packets that do not exit the switch through the port because STP has put the port into the blocking state, or on a trunk port because STP has put the VLAN into the blocking state on the trunk port.

VSPAN Guidelines and Restrictions


Note

Local SPAN, RSPAN, and ERSPAN all support VSPAN.

These are VSPAN guidelines and restrictions:

- For VSPAN sessions with both ingress and egress configured, two packets are forwarded from the destination port if the packets get switched on the same VLAN (one as ingress traffic from the ingress port and one as egress traffic from the egress port).
- VSPAN only monitors traffic that leaves or enters Layer 2 ports in the VLAN.
 - If you configure a VLAN as an ingress source and traffic gets routed into the monitored VLAN, the routed traffic is not monitored because it never appears as ingress traffic entering a Layer 2 port in the VLAN.
 - If you configure a VLAN as an egress source and traffic gets routed out of the monitored VLAN, the routed traffic is not monitored because it never appears as egress traffic leaving a Layer 2 port in the VLAN.

RSPAN Guidelines and Restrictions

These are RSPAN guidelines and restrictions:

- Supervisor Engine 2 does not support RSPAN if you configure an egress SPAN source for a local SPAN session.
- Supervisor Engine 2 does not support egress SPAN sources for local SPAN if you configure RSPAN.
- All participating switches must be trunk-connected at Layer 2.
- Any network device that supports RSPAN VLANs can be an RSPAN intermediate device.
- Networks impose no limit on the number of RSPAN VLANs that the networks carry.
- Intermediate network devices might impose limits on the number of RSPAN VLANs that they can support.
- You must configure the RSPAN VLANs in all source, intermediate, and destination network devices. If enabled, the VLAN Trunking Protocol (VTP) can propagate configuration of VLANs numbered 1 through 1024 as RSPAN VLANs. You must manually configure VLANs numbered higher than 1024 as RSPAN VLANs on all source, intermediate, and destination network devices.
- If you enable VTP and VTP pruning, RSPAN traffic is pruned in the trunks to prevent the unwanted flooding of RSPAN traffic across the network.
- RSPAN VLANs can be used only for RSPAN traffic.
- Do not configure a VLAN used to carry management traffic as an RSPAN VLAN.
- Do not assign access ports to RSPAN VLANs. RSPAN puts access ports in an RSPAN VLAN into the suspended state.
- Do not configure any ports in an RSPAN VLAN except trunk ports selected to carry RSPAN traffic.
- MAC address learning is disabled in the RSPAN VLAN.
- You can use output access control lists (ACLs) on the RSPAN VLAN in the RSPAN source switch to filter the traffic sent to an RSPAN destination.
- RSPAN does not support BPDU monitoring.

- Do not configure RSPAN VLANs as sources in VSPAN sessions.
- You can configure any VLAN as an RSPAN VLAN as long as all participating network devices support configuration of RSPAN VLANs and you use the same RSPAN VLAN for each RSPAN session in all participating network devices.

ERSPAN Guidelines and Restrictions

These are ERSPAN guidelines and restrictions:

- ERSPAN is supported only when the switch is operating in the compact switching mode: all modules must be fabric-enabled.
- Release 12.2(18)SXE and later releases support ERSPAN.
- All versions of WS-SUP720-3B (Supervisor Engine 720 with PFC3B) and WS-SUP720-3BXL (Supervisor Engine 720 with PFC3BXL) support ERSPAN.
- WS-SUP720 (Supervisor Engine 720 with PFC3A), hardware version 3.2 or higher, supports ERSPAN. 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
```

- Supervisor Engine 2 does not support ERSPAN.
- 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 10-10](#)) set the maximum Layer 3 packet size (default is 1,500 bytes, maximum is 9,216 bytes).

- All participating switches must be connected at Layer 3 and the network path must support the size of the ERSPAN traffic.
- ERSPAN does not support packet fragmentation. The “do not fragment” bit is set in the IP header of ERSPAN packets. ERSPAN destination sessions cannot reassemble fragmented ERSPAN packets.

- ERSPAN traffic is subject to the traffic load conditions of the network. You can set the ERSPAN packet IP precedence or DSCP value to prioritize ERSPAN traffic for QoS.
- The only supported destination for ERSPAN traffic is an ERSPAN destination session on a Supervisor Engine 720.
- All ERSPAN source sessions on a switch must use the same origin IP address, configured with the **origin ip address** command (see the [“Configuring ERSPAN Source Sessions” section on page 51-18](#)).
- 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 51-20](#)).
- The ERSPAN source session’s destination IP address, which must be configured on an interface on the destination switch, is the source of traffic that an ERSPAN destination session sends to the destination ports. You configure the same address in both the source and destination sessions with the **ip address** command.
- The ERSPAN ID differentiates the ERSPAN traffic arriving at the same destination IP address from various different ERSPAN source sessions.

Configuring Local SPAN, RSPAN, and ERSPAN

These sections describe how to configure local SPAN, RSPAN, and ERSPAN:

- [Configuring Destination Port Permit Lists \(Optional\), page 51-13](#)
- [Configuring Local SPAN, page 51-13](#)
- [Configuring RSPAN, page 51-15](#)
- [Configuring ERSPAN, page 51-17](#)
- [Configuring Source VLAN Filtering for Local SPAN and RSPAN, page 51-22](#)
- [Configuring a Destination Port as an Unconditional Trunk, page 51-22](#)
- [Configuring Destination Trunk Port VLAN Filtering, page 51-23](#)
- [Verifying the Configuration, page 51-24](#)
- [Configuration Examples, page 51-25](#)

Configuring Destination Port Permit Lists (Optional)

To prevent accidental configuration of ports as destinations, you can create a permit list of the ports that are valid for use as destinations. With a destination port permit list configured, you can only configure the ports in the permit list as destinations.

To configure a destination port permit list, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# monitor permit-list	Enables use of the destination port permit list.
Step 3	Router(config)# no monitor permit-list	Disables use of the destination port permit list.
Step 4	Router(config)# monitor permit-list destination interface <i>type</i> ¹ <i>slot/port[-port]</i> [, <i>type</i> ¹ <i>slot/port - port</i>]	Configures a destination port permit list or adds to an existing destination port permit list.
Step 5	Router(config)# no monitor permit-list destination interface <i>type</i> ¹ <i>slot/port[-port]</i> [, <i>type</i> ¹ <i>slot/port - port</i>]	Deletes from or clears an existing destination port permit list.
Step 6	Router(config)# do show monitor permit-list	Verifies the configuration.

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

This example shows how to configure a destination port permit list that includes Gigabit Ethernet ports 5/1 through 5/4 and 6/1:

```
Router# configure terminal
Router(config)# monitor permit-list
Router(config)# monitor permit-list destination interface gigabitethernet 5/1-4,
gigabitethernet 6/1
```

This example shows how to verify the configuration:

```
Router(config)# do show monitor permit-list
SPAN Permit-list      :Admin Enabled
Permit-list ports     :Gi5/1-4,Gi6/1
```

Configuring Local SPAN

Local SPAN does not use separate source and destination sessions. To configure a local SPAN session, configure local SPAN sources and destinations with the same session number. To configure a local SPAN session, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# monitor session <i>local_span_session_number</i> source {{ <i>single_interface</i> <i>interface_list</i> <i>interface_range</i> <i>mixed_interface_list</i> <i>single_vlan</i> <i>vlan_list</i> <i>vlan_range</i> <i>mixed_vlan_list</i> } [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

Command	Purpose
Router(config)# monitor session <i>local_span_session_number</i> destination { <i>single_interface</i> <i>interface_list</i> <i>interface_range</i> <i>mixed_interface_list</i> }	Associates the local SPAN session number and the destination ports.
Router(config)# no monitor session { <i>session_number</i> all local range <i>session_range</i> [<i>, session_range</i>], ...]}	Clears the monitor configuration.

When configuring local SPAN sessions, note the following information:

- *local_span_session_number* can range from 1 to 66.
- *single_interface* is **interface** type *slot/port*; type is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...

**Note**

In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** type *slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- To tag the monitored traffic as it leaves a destination port, you must configure the destination port to trunk unconditionally before you configure it as a destination (see the [“Configuring a Destination Port as an Unconditional Trunk”](#) section on page 51-22).

When clearing monitor sessions, note the following information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*.

**Note**

In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

This example shows how to configure Fast Ethernet port 5/1 as a bidirectional source for session 1:

```
Router(config)# monitor session 1 source interface fastethernet 5/1
```

This example shows how to configure Fast Ethernet port 5/48 as the destination for SPAN session 1:

```
Router(config)# monitor session 1 destination interface fastethernet 5/48
```

For additional examples, see the [“Configuration Examples”](#) section on page 51-25.

Configuring RSPAN

RSPAN uses a source session on one switch and a destination session on a different switch. These sections describe how to configure RSPAN sessions:

- [Configuring RSPAN VLANs, page 51-15](#)
- [Configuring RSPAN Source Sessions, page 51-15](#)
- [Configuring RSPAN Destination Sessions, page 51-16](#)

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 <i>vlan_ID</i> { [- <i>vlan_ID</i>] [, <i>vlan_ID</i>] }	Creates or modifies an Ethernet VLAN, a range of Ethernet VLANs, or several Ethernet VLANs specified in a comma-separated list (do not enter space characters).
Step 3	Router(config-vlan)# remote-span Router(config-vlan)# no remote-span	Configures the VLAN as an RSPAN VLAN. Clears the RSPAN VLAN configuration.
Step 4	Router(config-vlan)# end	Updates the VLAN database and returns to privileged EXEC mode.

Configuring RSPAN Source Sessions

To configure an RSPAN source session, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# monitor session <i>RSPAN_source_session_number</i> source { { <i>single_interface</i> <i>interface_list</i> <i>interface_range</i> <i>mixed_interface_list</i> <i>single_vlan</i> <i>vlan_list</i> <i>vlan_range</i> <i>mixed_vlan_list</i> } [<i>rx</i> <i>tx</i> <i>both</i>] }	Associates the RSPAN source session number with the source ports or VLANs, and selects the traffic direction to be monitored.
Step 3	Router(config)# monitor session <i>RSPAN_source_session_number</i> destination remote vlan <i>rspan_vlan_ID</i>	Associates the RSPAN source session number session number with the RSPAN VLAN.
Step 4	Router(config)# no monitor session { <i>session_number</i> all range <i>session_range</i> [[, <i>session_range</i>], ...] remote }	Clears the monitor configuration.

When configuring monitor sessions, note the following information:

- To configure RSPAN VLANs, see the [“Configuring RSPAN VLANs” section on page 51-15](#).
- *RSPAN_source_span_session_number* can range from 1 to 66.
- *single_interface* is **interface type slot/port**; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.

- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...



Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** type *slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...

When clearing monitor sessions, note the following information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*.



Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

This example shows how to configure Fast Ethernet port 5/2 as the source for session 2:

```
Router(config)# monitor session 2 source interface fastethernet 5/2
```

This example shows how to configure RSPAN VLAN 200 as the destination for session 2:

```
Router(config)# monitor session 2 destination remote vlan 200
```

For additional examples, see the [“Configuration Examples” section on page 51-25](#).

Configuring RSPAN Destination Sessions



Note You can configure an RSPAN destination session on the RSPAN source session switch to monitor RSPAN traffic locally.

To configure an RSPAN destination session, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# monitor session <i>RSPAN_destination_session_number</i> source remote vlan <i>rspan_vlan_ID</i>	Associates the RSPAN destination session number with the RSPAN VLAN.

	Command	Purpose
Step 3	Router(config)# monitor session <i>RSPAN_destination_session_number</i> destination { <i>single_interface</i> <i>interface_list</i> <i>interface_range</i> <i>mixed_interface_list</i> }	Associates the RSPAN destination session number with the destination ports.
Step 4	Router(config)# no monitor session { <i>session_number</i> all range <i>session_range</i> [, <i>session_range</i>],...] remote }	Clears the monitor configuration.

When configuring monitor sessions, note the following information:

- To tag the monitored traffic, you must configure the port to trunk unconditionally before you configure it as a destination (see the [“Configuring a Destination Port as an Unconditional Trunk” section on page 51-22](#)).
- RSPAN_destination_span_session_number* can range from 1 to 66.
- single_interface* is **interface** type slot/port; type is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- interface_list* is *single_interface* , *single_interface* , *single_interface* ...



Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- interface_range* is **interface** type slot/first_port - last_port.
- mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...

When clearing monitor sessions, note the following information:

- Enter the **no monitor session** *number* command with no other parameters to clear session *session_number*.
- session_range* is *first_session_number*-*last_session_number*.



Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

This example shows how to configure RSPAN VLAN 200 as the source for session 3:

```
Router(config)# monitor session 3 source remote vlan 200
```

This example shows how to configure Fast Ethernet port 5/47 as the destination for session 3:

```
Router(config)# monitor session 3 destination interface fastethernet 5/47
```

For additional examples, see the [“Configuration Examples” section on page 51-25](#).

Configuring ERSPAN

ERSPAN uses separate source and destination sessions. You configure the source and destination sessions on different switches. These sections describe how to configure ERSPAN sessions:

- [Configuring ERSPAN Source Sessions, page 51-18](#)
- [Configuring ERSPAN Destination Sessions, page 51-20](#)

**Note**

Release 12.2(18)SX-E and later releases support ERSPAN on Supervisor Engine 720 (see the [“ERSPAN Guidelines and Restrictions”](#) section on page 51-11).

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 <i>ERSPAN_source_session_number type erspan-source</i>	Configures an ERSPAN source session number and enters ERSPAN source session configuration mode for the session.
	Router(config)# no monitor session { <i>session_number</i> all range <i>session_range</i> [<i>, session_range</i>],...}	Clears the monitor configuration.
Step 3	Router(config-mon-erspan-src)# description <i>session_description</i>	(Optional) Describes the ERSPAN source session.
Step 4	Router(config-mon-erspan-src)# shutdown	(Default) Inactivates the ERSPAN source session.
	Router(config-mon-erspan-src)# no shutdown	Activates the ERSPAN source session.
Step 5	Router(config-mon-erspan-src)# source { <i>single_interface</i> <i>interface_list</i> <i>interface_range</i> <i>mixed_interface_list</i> <i>single_vlan</i> <i>vlan_list</i> <i>vlan_range</i> <i>mixed_vlan_list</i> } [rx tx both]	Associates the ERSPAN source session number with the source ports or VLANs, and selects the traffic direction to be monitored.
Step 6	Router(config-mon-erspan-src)# filter <i>single_vlan</i> <i>vlan_list</i> <i>vlan_range</i> <i>mixed_vlan_list</i>	(Optional) Configures source VLAN filtering when the ERSPAN source is a trunk port.
Step 7	Router(config-mon-erspan-src)# destination	Enters ERSPAN source session destination configuration mode.
Step 8	Router(config-mon-erspan-src-dst)# ip address <i>ip_address</i>	Configures the ERSPAN flow destination IP address, which must also be configured on an interface on the destination switch and be entered in the ERSPAN destination session configuration (see the “Configuring ERSPAN Destination Sessions” section on page 51-20, Step 7).
Step 9	Router(config-mon-erspan-src-dst)# erspan-id <i>ERSPAN_flow_id</i>	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 51-20, Step 8).
Step 10	Router(config-mon-erspan-src-dst)# origin ip address <i>ip_address</i> [force]	Configures the IP address used as the source of the ERSPAN traffic.
Step 11	Router(config-mon-erspan-src-dst)# ip ttl <i>ttl_value</i>	(Optional) Configures the IP time-to-live (TTL) value of the packets in the ERSPAN traffic.
Step 12	Router(config-mon-erspan-src-dst)# ip prec <i>ipp_value</i>	(Optional) Configures the IP precedence value of the packets in the ERSPAN traffic.

	Command	Purpose
Step 13	Router(config-mon-erspan-src-dst)# ip dscp <i>dscp_value</i>	(Optional) Configures the IP DSCP value of the packets in the ERSPAN traffic.
Step 14	Router(config-mon-erspan-src-dst)# vrf <i>vrf_name</i>	(Optional) Configures the VRF name to use instead of the global routing table.
Step 15	Router(config-mon-erspan-src-dst)# end	Exits configuration mode.

When configuring monitor sessions, note the following information:

- *session_description* can be up to 255 characters and cannot contain special characters or spaces.
- *ERSPAN_source_span_session_number* can range from 1 to 66.
- *single_interface* is **interface** *type slot/port*; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...



Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID - last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...
- *ERSPAN_flow_id* can range from 1 to 1023.
- All ERSPAN source sessions on a switch must use the same source IP address. Enter the **origin ip address ip_address force** command to change the origin IP address configured in all ERSPAN source sessions on the switch.
- *ttl_value* can range from 1 to 255.
- *ipp_value* can range from 0 to 7.
- *dscp_value* can range from 0 to 63.

When clearing monitor sessions, note the following information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*.



Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

This example shows how to configure session 3 to monitor bidirectional traffic from Gigabit Ethernet port 4/1:

```
Router(config)# monitor session 3 type erspan-source
Router(config-mon-erspan-src)# source interface gigabitethernet 4/1
Router(config-mon-erspan-src)# destination
Router(config-mon-erspan-src-dst)# ip address 10.1.1.1
Router(config-mon-erspan-src-dst)# origin ip address 20.1.1.1
Router(config-mon-erspan-src-dst)# erspan-id 101
```

For additional examples, see the “Configuration Examples” section on page 51-25.

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 <i>ERSPAN_destination_session_number</i> type erspan-destination Router(config)# no monitor session { <i>session_number</i> all range <i>session_range</i> [, <i>session_range</i>] , ... }	Configures an ERSPAN destination session number and enters ERSPAN destination session configuration mode for the session. Clears the monitor configuration.
Step 3	Router(config-mon-erspan-dst)# description <i>session_description</i>	(Optional) Describes the ERSPAN destination session.
Step 4	Router(config-mon-erspan-dst)# shutdown Router(config-mon-erspan-dst)# no shutdown	(Default) Inactivates the ERSPAN destination session. Activates the ERSPAN destination session.
Step 5	Router(config-mon-erspan-dst)# destination { <i>single_interface</i> <i>interface_list</i> <i>interface_range</i> <i>mixed_interface_list</i> }	Associates the ERSPAN destination session number with the destination ports.
Step 6	Router(config-mon-erspan-dst)# source	Enters ERSPAN destination session source configuration mode.
Step 7	Router(config-mon-erspan-dst-src)# ip address <i>ip_address</i> [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 51-18, Step 8 .
Step 8	Router(config-mon-erspan-dst-src)# erspan-id <i>ERSPAN_flow_id</i>	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 51-18, Step 9 .
Step 9	Router(config-mon-erspan-dst-src)# vrf <i>vrf_name</i>	(Optional) Configures the VRF name used instead of the global routing table.
Step 10	Router(config-mon-erspan-dst-src)# end	Exits configuration mode.

When configuring monitor sessions, note the following information:

- *ERSPAN_destination_span_session_number* can range from 1 to 66.
- *single_interface* is **interface** *type slot/port*; *type* is **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**.
- *interface_list* is *single_interface* , *single_interface* , *single_interface* ...



Note In lists, you must enter a space before and after the comma. In ranges, you must enter a space before and after the dash.

- *interface_range* is **interface** *type slot/first_port - last_port*.
- *mixed_interface_list* is, in any order, *single_interface* , *interface_range* , ...
- All ERSPAN destination sessions on a switch must use the same IP address on the same destination interface. Enter the **ip address ip_address force** command to change the IP address configured in all ERSPAN destination sessions on the switch.



Note You must also change all ERSPAN source session destination IP addresses (see the [“Configuring ERSPAN Source Sessions”](#) section on page 51-18, Step 8).

- *ERSPAN_flow_id* can range from 1 to 1023.

When clearing monitor sessions, note the following information:

- The **no monitor session** *number* command entered with no other parameters clears session *session_number*.
- *session_range* is *first_session_number-last_session_number*.



Note In the **no monitor session range** command, do not enter spaces before or after the dash. If you enter multiple ranges, do not enter spaces before or after the commas.

This example shows how to configure an ERSPAN destination session to send ERSPAN ID 101 traffic arriving at IP address 10.1.1.1 to Gigabit Ethernet port 2/1:

```
Router(config)# monitor session 3 type erspan-destination
Router(config-erspan-dst)# destination interface gigabitethernet 2/1
Router(config-erspan-dst)# source
Router(config-erspan-dst-src)# ip address 10.1.1.1
Router(config-erspan-dst-src)# erspan-id 101
```

For additional examples, see the [“Configuration Examples”](#) section on page 51-25.

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 51-17](#).

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 <i>session_number</i> filter <i>single_vlan</i> <i>vlan_list</i> <i>vlan_range</i> <i>mixed_vlan_list</i>	Configures source VLAN filtering when the local SPAN or RSPAN source is a trunk port.
	Router(config)# no monitor session <i>session_number</i> filter <i>single_vlan</i> <i>vlan_list</i> <i>vlan_range</i> <i>mixed_vlan_list</i>	Clears source VLAN filtering.

When configuring source VLAN filtering, note the following information:

- *single_vlan* is the ID number of a single VLAN.
- *vlan_list* is *single_vlan* , *single_vlan* , *single_vlan* ...
- *vlan_range* is *first_vlan_ID* - *last_vlan_ID*.
- *mixed_vlan_list* is, in any order, *single_vlan* , *vlan_range* , ...

This example shows how to monitor VLANs 1 through 5 and VLAN 9 when the source is a trunk port:

```
Router(config)# monitor session 2 filter vlan 1 - 5 , 9
```

Configuring a Destination Port as an Unconditional Trunk

To tag the monitored traffic as it leaves a destination port, configure the destination port as a trunk.

To configure the destination port as a trunk, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the LAN port to configure.
Step 3	Router(config-if)# switchport	Configures the LAN port for Layer 2 switching (required only if the LAN port is not already configured for Layer 2 switching).
Step 4	Router(config-if)# switchport trunk encapsulation { <i>isl</i> <i>dot1q</i> }	Configures the encapsulation, which configures the Layer 2 switching port as either an ISL or 802.1Q trunk.
Step 5	Router(config-if)# switchport mode trunk	Configures the port to trunk unconditionally.
Step 6	Router(config-if)# switchport nonegotiate	Configures the trunk not to use DTP.

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

This example shows how to configure a port as an unconditional IEEE 802.1Q trunk:

```
Router(config)# interface fastethernet 5/12
Router(config-if)# switchport
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# switchport nonegotiate
```

Configuring Destination Trunk Port VLAN Filtering

With Release 12.2(18)SXD and later releases, when a destination port is a trunk, you can use the list of VLANs allowed on the trunk to filter the traffic transmitted from the destination port. (CSCeb01318)

Destination trunk port VLAN filtering removes the restriction that all destination ports receive all the traffic from all the sources. Destination trunk port VLAN filtering allows you to select, on a per-VLAN basis, the traffic that is transmitted from each destination trunk port to the network analyzer.

To configure destination trunk port VLAN filtering on a destination trunk port, perform this task:

	Command	Purpose
Step 1	Router# configure terminal	Enters global configuration mode.
Step 2	Router(config)# interface <i>type</i> ¹ <i>slot/port</i>	Selects the destination trunk port to configure.
Step 3	Router(config-if)# switchport trunk allowed vlan { add except none remove } <i>vlan</i> [, <i>vlan</i> [, <i>vlan</i> [, ...]]	Configures the list of VLANs allowed on the trunk.

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

When configuring the list of VLANs allowed on a destination trunk port, note the following information:

- The *vlan* parameter is either a single VLAN number from 1 through 4094, or a range of VLANs described by two VLAN numbers, the lesser one first, separated by a dash. Do not enter any spaces between comma-separated *vlan* parameters or in dash-specified ranges.
- All VLANs are allowed by default.
- To remove all VLANs from the allowed list, enter the **switchport trunk allowed vlan none** command.
- To add VLANs to the allowed list, enter the **switchport trunk allowed vlan add** command.
- You can modify the allowed VLAN list without removing the SPAN configuration.

This example shows the configuration of a local SPAN session that has several VLANs as sources and several trunk ports as destinations, with destination trunk port VLAN filtering that filters the SPAN traffic so that each destination trunk port transmits the traffic from one VLAN:

```
interface GigabitEthernet1/1
description SPAN destination interface for VLAN 10
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 10
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/2
description SPAN destination interface for VLAN 11
no ip address
```

```

switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 11
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/3
description SPAN destination interface for VLAN 12
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 12
switchport mode trunk
switchport nonegotiate
!
interface GigabitEthernet1/4
description SPAN destination interface for VLAN 13
no ip address
switchport
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 13
switchport mode trunk
switchport nonegotiate
!
monitor session 1 source vlan 10 - 13
monitor session 1 destination interface Gi1/1 - 4

```

Verifying the Configuration

To verify the configuration, enter the **show monitor session** command.

This example shows how to verify the configuration of session 2:

```

Router# show monitor session 2
Session 2
-----
Type : Remote Source Session

Source Ports:
    RX Only:      Fa3/1
Dest RSPAN VLAN: 901
Router#

```

This example shows how to display the full details of session 2:

```

Router# show monitor session 2 detail
Session 2
-----
Type : Remote Source Session

Source Ports:
    RX Only:      Fa1/1-3
    TX Only:      None
    Both:         None
Source VLANs:
    RX Only:      None
    TX Only:      None
    Both:         None
Source RSPAN VLAN: None
Destination Ports: None
Filter VLANs:    None
Dest RSPAN VLAN: 901

```

Configuration Examples

This example shows the configuration of RSPAN source session 2:

```
Router(config)# monitor session 2 source interface fastethernet1/1 - 3 rx
Router(config)# monitor session 2 destination remote vlan 901
```

This example shows how to clear the configuration for sessions 1 and 2:

```
Router(config)# no monitor session range 1-2
```

This example shows the configuration of an RSPAN source session with multiple sources:

```
Router(config)# monitor session 2 source interface fastethernet 5/15 , 7/3 rx
Router(config)# monitor session 2 source interface gigabitethernet 1/2 tx
Router(config)# monitor session 2 source interface port-channel 102
Router(config)# monitor session 2 source filter vlan 2 - 3
Router(config)# monitor session 2 destination remote vlan 901
```

This example shows how to remove sources for a session:

```
Router(config)# no monitor session 2 source interface fastethernet 5/15 , 7/3
```

This example shows how to remove options for sources for a session:

```
Router(config)# no monitor session 2 source interface gigabitethernet 1/2
Router(config)# no monitor session 2 source interface port-channel 102 tx
```

This example shows how to remove VLAN filtering for a session:

```
Router(config)# no monitor session 2 filter vlan 3
```

This example shows the configuration of RSPAN destination session 8:

```
Router(config)# monitor session 8 source remote vlan 901
Router(config)# monitor session 8 destination interface fastethernet 1/2 , 2/3
```

This example shows the configuration of ERSPAN source session 12:

```
monitor session 12 type erspan-source
description SOURCE_SESSION_FOR_VRF_GRAY
source interface Gi8/48 rx
destination
  erspan-id 120
  ip address 10.8.1.2
  origin ip address 32.1.1.1
  vrf gray
```

This example shows the configuration of ERSPAN destination session 12:

```
monitor session 12 type erspan-destination
description DEST_SESSION_FOR_VRF_GRAY
destination interface Gi4/48
source
  erspan-id 120
  ip address 10.8.1.2
  vrf gray
```

This example shows the configuration of ERSPAN source session 13:

```
monitor session 13 type erspan-source
source interface Gi6/1 tx
destination
  erspan-id 130
  ip address 10.11.1.1
  origin ip address 32.1.1.1
```

This example shows the configuration of ERSPAN destination session 13:

```
monitor session 13 type erspan-destination
destination interface Gi6/1
source
  erspan-id 130
  ip address 10.11.1.1
```



Configuring SNMP IfIndex Persistence

This chapter describes how to configure the SNMP ifIndex persistence feature on Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding SNMP IfIndex Persistence, page 52-1](#)
- [Configuring SNMP IfIndex Persistence, page 52-2](#)

Understanding SNMP IfIndex Persistence

The SNMP ifIndex persistence feature provides an interface index (ifIndex) value that is retained and used when the switch reboots. The ifIndex value is a unique identifying number associated with a physical or logical interface.

There is no requirement in the relevant RFCs that the correspondence between particular ifIndex values and their interfaces be maintained when the switch reboots, but many applications (for example, device inventory, billing, and fault detection) require maintenance of this correspondence.

You can poll the switch at regular intervals to correlate the interfaces to the ifIndexes, but it is not practical to poll constantly. The SNMP ifIndex persistence feature provides permanent ifIndex values, which eliminates the need to poll interfaces.

The following definitions are based on RFC 2233, “The Interfaces Group MIB using SMIv2.” The following terms are values in the Interfaces MIB (IF-MIB):

- **ifIndex**—A unique number (greater than zero) that identifies each interface for SNMP identification of that interface.
- **ifName**—The text-based name of the interface, for example, “ethernet 3/1.”
- **ifDescr**—A description of the interface. Recommended information for this description includes the name of the manufacturer, the product name, and the version of the interface hardware and software.

Configuring SNMP IfIndex Persistence

These sections describe how to configure SNMP ifIndex persistence:

- [Enabling SNMP IfIndex Persistence Globally, page 52-2](#) (Optional)
- [Enabling and Disabling SNMP IfIndex Persistence on Specific Interfaces, page 52-2](#) (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 enable SNMP 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 <i>vlan_ID</i> { <i>type</i> ¹ <i>slot/port</i> } { port-channel <i>port_channel_number</i> }	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 <i>type slot/port</i>	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
```




Power Management and Environmental Monitoring

This chapter describes the power management and environmental monitoring features in the Catalyst 6500 series switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How Power Management Works, page 53-1](#)
- [Understanding How Environmental Monitoring Works, page 53-10](#)

Understanding How Power Management Works

These sections describe power management in the Catalyst 6500 series switches:

- [Enabling or Disabling Power Redundancy, page 53-2](#)
- [Powering Modules Off and On, page 53-3](#)
- [Viewing System Power Status, page 53-4](#)
- [Power Cycling Modules, page 53-5](#)
- [Power Cycling Power Supplies, page 53-5](#)
- [Determining System Power Requirements, page 53-5](#)
- [Determining System Hardware Capacity, page 53-5](#)
- [Determining Sensor Temperature Threshold, page 53-9](#)



Note

In systems with redundant power supplies, both power supplies must be of the same wattage. The Catalyst 6500 series switches allow you to use both AC-input and DC-input power supplies in the same chassis. For detailed information on supported power supply configurations, refer to the *Catalyst 6500 Series Switch Installation Guide*.

The modules have different power requirements, and some configurations require more power than a single power supply can provide. The power management feature allows you to power all installed modules with two power supplies. However, redundancy is not supported in this configuration because the total power drawn from both power supplies is at no time greater than the capability of one supply. Redundant and nonredundant power configurations are described in the following sections.

To determine the power requirements for your system, see the [“Determining System Power Requirements” section on page 53-5](#).

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 53-4](#)).

[Table 53-1](#) describes how the system responds to changes in the power supply configuration.

Table 53-1 Effects of Power Supply Configuration Changes

Configuration Change	Effect
Redundant to nonredundant	<ul style="list-style-type: none"> System log and syslog messages are generated. System power is increased to the combined power capability of both power supplies. Modules marked <i>power-deny</i> 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)	<ul style="list-style-type: none"> 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 <i>power-deny</i> in the show power oper state field.
Equal wattage power supply is inserted with redundancy enabled	<ul style="list-style-type: none"> 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 53-1 Effects of Power Supply Configuration Changes (continued)

Configuration Change	Effect
Equal wattage power supply is inserted with redundancy disabled	<ul style="list-style-type: none"> System log and syslog messages are generated. System power is increased to the combined power capability of both power supplies. Modules marked <i>power-deny</i> 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	<ul style="list-style-type: none"> System log and syslog messages are generated. The system does not allow you to operate a power supply of different wattage even if the wattage is higher than the installed supply. The inserted supply shuts down.
Higher or lower wattage power supply is inserted with redundancy disabled	<ul style="list-style-type: none"> System log and syslog messages are generated. System power is increased to the combined power capability of both power supplies. Modules marked <i>power-deny</i> in the show power oper state field are brought up if there is sufficient power.
Power supply is removed with redundancy enabled	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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 <i>power-deny</i> in the show power oper state field.
System is booted with power supplies of different wattage installed and redundancy enabled	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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 <i>slot_number</i>	Powers a module on.
	Router(config)# no power enable module <i>slot_number</i>	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)

PS   Type                Power-Capacity PS-Fan Output Oper
Watts A @42V Status Status State
-----
1    WS-CAC-2500W         1153.32 27.46 OK      OK      on
2    none

Slot Card-Type                Pwr-Requested Pwr-Allocated Admin Oper
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

PS   Type                Power-Capacity PS-Fan Output Oper
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

With Release 12.2(18)SXF and later releases, you can determine the system hardware capacity by entering the **show platform hardware capacity** command. This command displays the current system utilization of the hardware resources and displays a list of the currently available hardware capacities, including the following:

- Hardware forwarding table utilization
- Switch fabric utilization
- CPU(s) utilization
- Memory device (flash, DRAM, NVRAM) utilization

This example shows how to display CPU capacity and utilization information for the route processor, the switch processor, and the LAN module in the Catalyst 6500 series switch:

```
Router# show platform hardware capacity cpu
CPU Resources
CPU utilization: Module          5 seconds    1 minute    5 minutes
```

```

          1  RP          0% /  0%          1%          1%
          1  SP          5% /  0%          5%          4%
          7           69% /  0%         69%         69%
          8           78% /  0%         74%         74%

Processor memory: Module  Bytes:      Total      Used      %Used
                   1  RP      176730048      51774704      29%
                   1  SP      192825092      51978936      27%
                   7           195111584      35769704      18%
                   8           195111584      35798632      18%

I/O memory: Module  Bytes:      Total      Used      %Used
                1  RP      35651584      12226672      34%
                1  SP      35651584      9747952       27%
                7           35651584      9616816      27%
                8           35651584      9616816      27%
Router#

```

This example shows how to display EOBC-related statistics for the route processor, the switch processor, and the DFCs in the Catalyst 6500 series switch:

```

Router# show platform hardware capacity eobc EOBC Resources
Module
  1  RP      Rx:          61          108982          0
           Tx:          37           77298          0
  1  SP      Rx:          34          101627          0
           Tx:          39          115417          0
  7           Rx:          5           10358          0
           Tx:          8           18543          0
  8           Rx:          5           12130          0
           Tx:         10           20317          0
Router#

```

This example shows how to display the current and peak switching utilization:

```

Router# show platform hardware capacity fabric Switch Fabric Resources
Bus utilization: current is 100%, peak was 100% at 12:34 12mar45
Fabric utilization:      ingress      egress
Module channel speed current peak      current peak
  1      0      20G  100% 100% 12:34 12mar45  100%  100% 12:34 12mar45
  1      1      20G   12%   80% 12:34 12mar45   12%   80% 12:34 12mar45
  4      0      20G   12%   80% 12:34 12mar45   12%   80% 12:34 12mar45
 13      0       8G   12%   80% 12:34 12mar45   12%   80% 12:34 12mar45
Router#

```

This example shows how to display information about the total capacity, the bytes used, and the percentage that is used for the flash and NVRAM resources present in the system:

```

Router# show platform hardware capacity flash
Flash/NVRAM Resources
Usage: Module Device      Bytes:      Total      Used      %Used
      1  RP  bootflash:      31981568      15688048      49%
      1  SP  disk0:      128577536      105621504      82%
      1  SP  sup-bootflash:      31981568      29700644      93%
      1  SP  const_nvram:      129004          856          1%
      1  SP  nvram:      391160      22065          6%
      7      dfc#7-bootflash:      15204352      616540          4%
      8      dfc#8-bootflash:      15204352          0          0%
Router#

```

This example shows how to display the capacity and utilization of the EARLs present in the system:

```

Router# show platform hardware capacity forwarding

```

```

L2 Forwarding Resources
  MAC Table usage:  Module Collisions Total      Used      %Used
                   6          0  65536      11        1%
  VPN CAM usage:           Total      Used      %Used
                   512          0        0%

L3 Forwarding Resources
  FIB TCAM usage:           Total      Used      %Used
    72 bits (IPv4, MPLS, EoM)  196608    36        1%
    144 bits (IP mcast, IPv6)  32768     7        1%

    detail:      Protocol      Used      %Used
                IPv4          36        1%
                MPLS           0        0%
                EoM            0        0%

                IPv6           4        1%
                IPv4 mcast     3        1%
                IPv6 mcast     0        0%

  Adjacency usage:           Total      Used      %Used
                   1048576    175        1%

Forwarding engine load:
  Module      pps    peak-pps    peak-time
  6            8      1972  02:02:17 UTC Thu Apr 21 2005

Netflow Resources
  TCAM utilization:  Module      Created      Failed      %Used
                   6            1          0          0%
  ICAM utilization:  Module      Created      Failed      %Used
                   6            0          0          0%

  Flowmasks:  Mask#    Type      Features
    IPv4:      0    reserved    none
    IPv4:      1  Intf FulNAT_INGRESS NAT_EGRESS FM_GUARDIAN
    IPv4:      2    unused      none
    IPv4:      3    reserved    none

    IPv6:      0    reserved    none
    IPv6:      1    unused      none
    IPv6:      2    unused      none
    IPv6:      3    reserved    none

CPU Rate Limiters Resources
  Rate limiters:      Total      Used      Reserved      %Used
    Layer 3           9          4          1          44%
    Layer 2           4          2          2          50%

ACL/QoS TCAM Resources
  Key: ACLent - ACL TCAM entries, ACLmsk - ACL TCAM masks, AND - ANDOR,
       QoSEnt - QoS TCAM entries, QoSmsk - QoS TCAM masks, OR - ORAND,
       Lbl-in - ingress label, Lbl-eg - egress label, LOUsrc - LOU source,
       LOUdst - LOU destination, ADJ - ACL adjacency

  Module ACLent ACLmsk QoSEnt QoSmsk Lbl-in Lbl-eg LOUsrc LOUdst AND OR ADJ
  6         1%      1%      1%      1%      1%      1%      0%      0%  0% 0%  1%

Router#

```

This example shows how to display the interface resources:

```

Router# show platform hardware capacity interface Interface Resources
Interface drops:

```

```

Module      Total drops:  Tx      Rx      Highest drop port: Tx  Rx
9           0             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
                               egress  egress
                               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 Catalyst 6500 series switch.

```

Router# show platform hardware capacity qos
QoS Policer Resources
Aggregate policers: Module      Total      Used      %Used
                               1          1024      102       10%
                               5          1024       1         1%
Microflow policer configurations: Module      Total      Used      %Used
                               1           64       32        50%
                               5           64       1         1%
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 53-2](#) lists the environmental indicators for the supervisor engine and switching modules.



Note

Refer to the *Catalyst 6500 Series Switch Module Installation Guide* for additional information on LEDs, including the supervisor engine SYSTEM LED.

Table 53-2 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 53-2 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 53-1 for instructions.



Configuring Online Diagnostics

This chapter describes how to configure the online diagnostics on the Catalyst 6500 series switches:



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding How Online Diagnostics Work, page 54-1](#)
- [Configuring Online Diagnostics, page 54-2](#)
- [Running Online Diagnostic Tests, page 54-6](#)
- [Performing Memory Tests, page 54-10](#)

Understanding How Online Diagnostics Work

With online diagnostics, you can test and verify the hardware functionality of the supervisor engine, modules, and switch while the switch is connected to a live network.

The online diagnostics contain packet switching tests that check different hardware components and verify the data path and control signals. Disruptive online diagnostic tests, such as the built-in self-test (BIST) and the disruptive loopback test, and nondisruptive online diagnostic tests, such as packet switching, run during bootup, line card online insertion and removal (OIR), and system reset. The nondisruptive online diagnostic tests run as part of background health monitoring or at the user's request (on-demand).

The online diagnostics detect problems in the following areas:

- Hardware components
- Interfaces (GBICs, Ethernet ports, and so forth)
- Connectors (loose connectors, bent pins, and so forth)
- Solder joints
- Memory (failure over time)

Online diagnostics is one of the requirements for the high availability feature. High availability is a set of quality standards that seek to limit the impact of equipment failures on the network. A key part of high availability is detecting hardware failures and taking corrective action while the switch runs in a live network. Online diagnostics in high availability detect hardware failures and provide feedback to high availability software components to make switchover decisions.

Online diagnostics are categorized as bootup, on-demand, schedule, or health monitoring diagnostics. Bootup diagnostics run during bootup, module OIR, or switchover to a backup supervisor engine; on-demand diagnostics run from the CLI; schedule diagnostics run at user-designated intervals or specified times when the switch is connected to a live network; and health-monitoring runs in the background.

Configuring Online Diagnostics

These sections describe how to configure online diagnostics:

- [Setting Bootup Online Diagnostics Level, page 54-2](#)
- [Configuring On-Demand Online Diagnostics, page 54-3](#)
- [Scheduling Online Diagnostics, page 54-4](#)

Setting Bootup Online Diagnostics Level

You can set the bootup diagnostics level as minimal or complete or you can bypass the bootup diagnostics entirely. Enter the **complete** keyword to run all diagnostic tests; enter the **minimal** keyword to run only EARL tests for the supervisor engine and loopback tests for all ports in the switch. Enter the **no** form of the command to bypass all diagnostic tests. The default bootup diagnostics level is minimal.

**Note**

The diagnostic level applies to the entire switch and cannot be configured on a per-module basis.

To set the bootup diagnostic level, perform this task:

Command	Purpose
Router(config)# diagnostic bootup level { minimal complete }	Sets the bootup diagnostic level.

This example shows how to set the bootup online diagnostic level:

```
Router(config)# diagnostic bootup level complete
Router(config)#
```

This example shows how to display the bootup online diagnostic level:

```
Router(config)# 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 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 (this test was the TestLinecardMemory test in releases prior to 12.2(18)SXF)

You must reboot the supervisor engine after running the exhaustive memory tests before it is operational again. You cannot run any other tests on the supervisor engine or other modules after running the exhaustive memory tests. Do not save the configuration when rebooting as it will have changed during the tests. You will need to power cycle the modules before they can be operational. After a module comes back on line, reenable the health monitoring tests using the **diagnostic monitor module *module* test all** command

To set the bootup diagnostic level, perform this task:

Command	Purpose
Router# diagnostic ondemand {iteration iteration_count} {action-on-error {continue stop} [error_count]}	Configures on-demand diagnostic tests to run, how many times to run (iterations), and what action to take when errors are found.

This example shows how to set the on-demand testing iteration count:

```
Router# diagnostic ondemand iteration 3
Router#
```

This example shows how to set the execution action when an error is detected:

```
Router# diagnostic ondemand action-on-error continue 2
Router#
```

Scheduling Online Diagnostics

You can schedule online diagnostics to run at a designated time of day or on a daily, weekly, or monthly basis for a specific module. You can schedule tests to run only once or to repeat at an interval. Use the **no** form of this command to remove the scheduling.

To schedule online diagnostics, perform this task:

Command	Purpose
Router(config)# diagnostic schedule {module num} test {test_id test_id_range all} [port {num num_range all}] {on mm dd yyyy hh:mm} {daily hh:mm} {weekly day_of_week hh:mm}	Schedules on-demand diagnostic tests for a specific date and time, how many times to run (iterations), and what action to take when errors are found.

This example shows how to schedule diagnostic testing on a specific date and time for a specific module and port:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 on january 3 2003 23:32
Router(config)#
```

This example shows how to schedule diagnostic testing to occur daily at a certain time for a specific port and module:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 daily 12:34
Router(config)#
```

This example shows how to schedule diagnostic testing to occur weekly on a certain day for a specific port and module:

```
Router(config)# diagnostic schedule module 1 test 1,2,5-9 port 3 weekly friday 09:23
Router(config)#
```

Configuring Health-Monitoring Diagnostics

You can configure health-monitoring diagnostic testing on specified modules while the switch is connected to a live network. You can configure the execution interval for each health monitoring test, whether or not to generate a system message upon test failure, or to enable or disable an individual test. Use the **no** form of this command to disable testing.

To configure health monitoring diagnostic testing, perform this task:

	Command	Purpose
Step 1	Router(config)# diagnostic monitor interval {module num} test {test_id test_id_range all} [hour hh] [min mm] [second ss] [millisec ms] [day day]	Configures the health-monitoring interval of the specified tests for the specified module. The no form of this command will change the interval to the default interval, or zero.
Step 2	Router(config)#[no] diagnostic monitor {module num} test {test_id test_id_range all}	Enables or disables health-monitoring diagnostic tests.

This example shows how to configure the specified test to run every two minutes:

```
Router(config)# diagnostic monitor interval module 1 test 1 min 2
Router(config)#
```

This example shows how to run the test on the specified module if health monitoring has not previously been enabled:

```
Router(config)# diagnostic monitor module 1 test 1
```

This example shows how to enable the generation of a syslog message when any health monitoring test fails:

```
Router(config)# diagnostic monitor syslog
Router(config)#
```

Running Online Diagnostic Tests

After you configure online diagnostics, you can start or stop diagnostic tests or display the test results. You can also see which tests are configured for each module and what diagnostic tests have already run.

These sections describe how to run online diagnostic tests after they have been configured:

- [Starting and Stopping Online Diagnostic Tests, page 54-6](#)
- [Displaying Online Diagnostic Tests and Test Results, page 54-6](#)

Starting and Stopping Online Diagnostic Tests

After you configure diagnostic tests to run on the switch or individual modules, you can use the **start** and **stop** to begin or end a diagnostic test.

To start or stop an online diagnostic command, perform one of these tasks:

Command	Purpose
<code>diagnostic start {module num} test {test_id test_id_range minimal complete basic per-port non-disruptive all} [port {num port#_range all}]</code>	Starts a diagnostic test on a specific module and port or range of ports.
<code>diagnostic stop {module num}</code>	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

```

ID	Test Name	Attributes	Testing Interval (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)	TestQoS Tcam	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) TestQoS Tcam -----> .
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	<code>show diagnostic content [module num]</code>	Displays the online diagnostics configured for a module. Use this command to obtain the test ID for the schedule switchover.
Step 2	<code>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}</code>	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 PFC3B of the supervisor engine.

- 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
<code>show diagnostic sanity</code>	Runs a set of tests on all of the Gigabit Ethernet WAN interfaces in the Catalyst 6500 series switch.

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



Configuring Web Cache Services Using WCCP

This chapter describes how to configure the Catalyst 6500 series switches 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 *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>
- Release 12.2(17d)SXB and later release support WCCP on Supervisor Engine 2.
- Release 12.2(18)SXD1 and later release support WCCP on Supervisor Engine 720.
- To use the WCCP Layer 2 PFC redirection feature, configure WCCP on the Catalyst 6500 series switch 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/ucc/acns42/cnfg42/transprt.htm#xtocid34>
- A release of Cisco Application and Content Networking System (ACNS) software later than Release 4.2.2 supports the **ip wccp service accelerated** command with a PFC2.
- 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:

- [Understanding WCCP, page 55-2](#)
- [Restrictions for WCCPv2, page 55-7](#)
- [Configuring WCCP, page 55-8](#)
- [Verifying and Monitoring WCCP Configuration Settings, page 55-12](#)
- [WCCP Configuration Examples, page 55-12](#)

**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 at <http://www.cisco.com/warp/public/cc/pd/cxsr/ces/index.shtml>.

Understanding WCCP

These sections describe WCCP:

- [WCCP Overview, page 55-2](#)
- [Hardware Acceleration, page 55-3](#)
- [Understanding WCCPv1 Configuration, page 55-4](#)
- [Understanding WCCPv2 Configuration, page 55-5](#)
- [WCCPv2 Features, page 55-6](#)

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” in 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

Catalyst 6500 series switches 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.



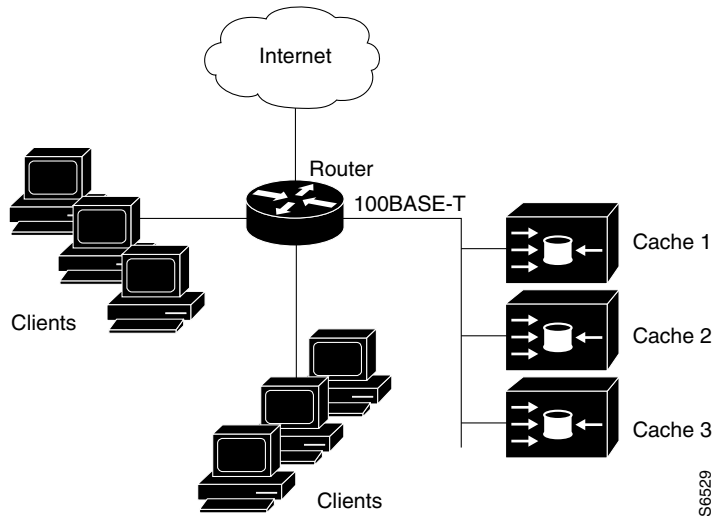
Note

- The PFC3 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 55-1](#) illustrates how this configuration appears.

Figure 55-1 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.

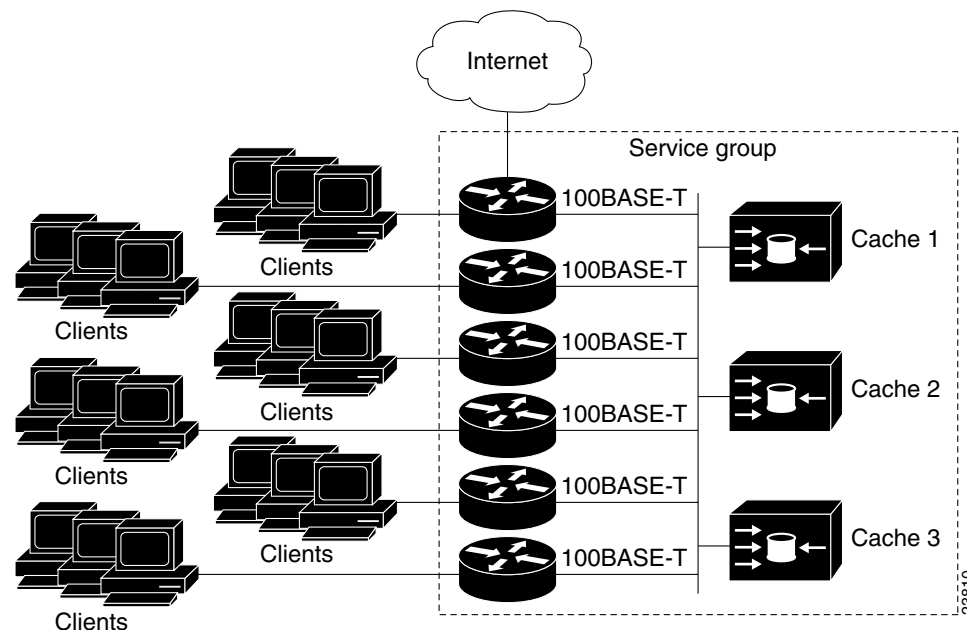
The following sequence of events details how WCCPv1 configuration works:

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 55-2 illustrates a sample configuration using multiple routers.

Figure 55-2 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 unserved. 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 User 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 55-8](#) (Optional)
- [Configuring a Service Group Using WCCPv2, page 55-8](#) (Required)
- [Excluding Traffic on a Specific Interface from Redirection, page 55-10](#) (Optional)
- [Registering a Router to a Multicast Address, page 55-10](#) (Optional)
- [Using Access Lists for a WCCP Service Group, page 55-11](#) (Optional)
- [Setting a Password for a Router and Cache Engines, page 55-11](#) (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 Catalyst 6500 series switch, perform this task:

	Command	Purpose
Step 1	Router(config)# ip wccp { web-cache <i>service-number</i> } [accelerated] [group-address <i>groupaddress</i>] [redirect-list <i>access-list</i>] [group-list <i>access-list</i>] [password <i>password</i>]	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.
Step 2	Router(config)# interface <i>type number</i>	Specifies an interface to configure and enters interface configuration mode.
Step 3	Router(config-if)# ip wccp { web-cache <i>service-number</i> } 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 <i>type number</i>	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 <i>type number</i>	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 <i>service-number</i> } group-address <i>groupaddress</i>	Specifies the multicast address for the service group.
Step 2	Router(config)# interface <i>type number</i>	Specifies the interface to be configured for multicast reception.
Step 3	Router(config-if)# ip wccp { web-cache <i>service-number</i> } 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 <i>access-list</i> permit ip host <i>host-address</i> [<i>destination-address</i> <i>destination-host</i> 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 <i>access-list</i>	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 <i>access-list</i> permit ip host <i>host-address</i> [<i>destination-address</i> <i>destination-host</i> 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 <i>access-list</i>	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 <i>password</i>	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 <i>service-number</i>]	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 <i>service-number</i> } 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 <i>service-number</i> } 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 <i>service number</i> } service command.

WCCP Configuration Examples

This section provides the following configuration examples:

- [Changing the Version of WCCP on a Router Example, page 55-13](#)
- [Performing a General WCCPv2 Configuration Example, page 55-13](#)
- [Running a Web Cache Service Example, page 55-13](#)
- [Running a Reverse Proxy Service Example, page 55-14](#)
- [Registering a Router to a Multicast Address Example, page 55-14](#)
- [Using Access Lists Example, page 55-14](#)
- [Setting a Password for a Router and Cache Engines Example, page 55-15](#)
- [Verifying WCCP Settings Example, page 55-15](#)

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 are 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
```



Using the Top N Utility

This chapter describes how to use the Top N utility on the Catalyst 6500 series switches. Release 12.2(18)SXE and later releases support the Top N utility.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter consists of these sections:

- [Understanding the Top N Utility, page 56-1](#)
- [Using the Top N Utility, page 56-2](#)

Understanding the Top N Utility

These sections describe the Top N utility:

- [Top N Utility Overview, page 56-1](#)
- [Understanding Top N Utility Operation, page 56-2](#)

Top N Utility Overview

The Top N utility allows you to collect and analyze data for each physical port on a switch. 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 56-1](#).

Table 56-1 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, page 56-3](#)
- [Displaying the Top N Utility Reports, page 56-3](#)
- [Clearing Top N Utility Reports, page 56-4](#)

Enabling Top N Utility Report Creation

To enable Top N Utility report creation, perform this task:

Command	Purpose
Router# collect top [<i>number_of_ports</i>] counters interface { <i>interface_type</i> ¹ all layer-2 layer-3 } [sort-by <i>statistic_type</i> ²] [interval <i>seconds</i>]	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 [<i>report_num</i>]	Displays the Top N Utility reports.
	Note To display information about all the reports, do not enter a <i>report_num</i> 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 width	Util	Bytes (Tx + Rx)	Packets (Tx + Rx)	Broadcast (Tx + Rx)	Multicast (Tx + Rx)	In-err	Buf-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 <i>[report_num]</i>	Clears Top N Utility report number <i>report_num</i> 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
```




Using the Layer 2 Traceroute Utility

This chapter describes how to use the Layer 2 traceroute utility. Release 12.2(18)SXE and later releases support the Layer 2 traceroute utility.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6500 Series Switch Cisco IOS Command Reference*, Release 12.2SX at this URL:

<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/cmdref/index.htm>

This chapter contains these sections:

- [Understanding the Layer 2 Traceroute Utility, page 57-1](#)
- [Usage Guidelines, page 57-1](#)
- [Using the Layer 2 Traceroute Utility, page 57-2](#)

Understanding the Layer 2 Traceroute Utility

The Layer 2 traceroute utility identifies the Layer 2 path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses. The utility determines the path by using the MAC address tables of the switches in the path. When the Layer 2 traceroute utility detects a device in the path that does not support Layer 2 traceroute, it continues to send Layer 2 trace queries and allows them to time out.

The Layer 2 traceroute utility can only identify the path from the source device to the destination device. The utility cannot identify the path that a packet takes from the source host to the source device or from the destination device to the destination host.

Usage Guidelines

When using the Layer 2 traceroute utility, follow these guidelines:

- Cisco Discovery Protocol (CDP) must be enabled on all the devices in the network. For the Layer 2 traceroute utility to function properly, do not disable CDP. If any devices in the Layer 2 path are transparent to CDP, the Layer 2 traceroute utility cannot identify these devices on the path.

**Note**

For more information about CDP, see [Chapter 48, “Configuring CDP.”](#)

- A switch is defined as reachable from another switch when you can test connectivity by using the **ping** privileged EXEC command. All devices in the Layer 2 path must be mutually reachable.
- The maximum number of hops identified in the path is ten.
- You can enter the **traceroute mac** or the **traceroute mac ip** privileged EXEC command on a switch that is not in the Layer 2 path from the source device to the destination device. All devices in the path must be reachable from this switch.
- The **traceroute mac** command output shows the Layer 2 path only when the specified source and destination MAC addresses belong to the same VLAN. If you specify source and destination MAC addresses that belong to different VLANs, the Layer 2 path is not identified, and an error message appears.
- If you specify a multicast source or destination MAC address, the path is not identified, and an error message appears.
- If the source or destination MAC address belongs to multiple VLANs, you must specify the VLAN to which both the source and destination MAC addresses belong. If the VLAN is not specified, the path is not identified, and an error message appears.
- The **traceroute mac ip** command output shows the Layer 2 path when the specified source and destination IP addresses belong to the same subnet. When you specify the IP addresses, the Layer 2 traceroute utility uses the Address Resolution Protocol (ARP) to associate the IP addresses with the corresponding MAC addresses and the VLAN IDs.
 - If an ARP entry exists for the specified IP address, the Layer 2 traceroute utility uses the associated MAC address and identifies the Layer 2 path.
 - If an ARP entry does not exist, the Layer 2 traceroute utility sends an ARP query and tries to resolve the IP address. If the IP address is not resolved, the path is not identified, and an error message appears.
- When multiple devices are attached to one port through hubs (for example, multiple CDP neighbors are detected on a port), the Layer 2 traceroute utility terminates at that hop and displays an error message.
- The Layer 2 traceroute utility is not supported in Token Ring VLANs.

Using the Layer 2 Traceroute Utility

To display the Layer 2 path that a packet takes from a source device to a destination device, perform one of these tasks in privileged EXEC mode:

Command	Purpose
Router# traceroute mac [interface type interface_number] source_mac_address [interface type interface_number] destination_mac_address [vlan vlan_id] [detail]	Uses MAC addresses to trace the path that packets take through the network.
Router# traceroute mac ip {source_ip_address source_hostname} {destination_ip_address destination_hostname} [detail]	Uses IP addresses to trace the path that packets take through the network.

These examples show how to use the **traceroute mac** and **traceroute mac ip** commands to display the physical path a packet takes through the network to reach its destination:

```
Router# traceroute mac 0000.0201.0601 0000.0201.0201
```

```
Source 0000.0201.0601 found on con6[WS-C2950G-24-EI] (2.2.6.6)
con6 (2.2.6.6) :Fa0/1 => Fa0/3
con5          (2.2.5.5       ) : Fa0/3 => Gi0/1
con1          (2.2.1.1       ) : Gi0/1 => Gi0/2
con2          (2.2.2.2       ) : Gi0/2 => Fa0/1
Destination 0000.0201.0201 found on con2[WS-C3550-24] (2.2.2.2)
Layer 2 trace completed
Router#
```

```
Router# traceroute mac 0001.0000.0204 0001.0000.0304 detail
```

```
Source 0001.0000.0204 found on VAYU[WS-C6509] (2.1.1.10)
1 VAYU / WS-C6509 / 2.1.1.10 :
    Gi6/1 [full, 1000M] => Po100 [auto, auto]
2 PANI / WS-C6509 / 2.1.1.12 :
    Po100 [auto, auto] => Po110 [auto, auto]
3 BUMI / WS-C6509 / 2.1.1.13 :
    Po110 [auto, auto] => Po120 [auto, auto]
4 AGNI / WS-C6509 / 2.1.1.11 :
    Po120 [auto, auto] => Gi8/12 [full, 1000M] Destination 0001.0000.0304
found on AGNI[WS-C6509] (2.1.1.11) Layer 2 trace completed.
Router#
```




Acronyms

[Table A-1](#) defines the acronyms used in this publication.

Table A-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 A-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 A-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
IDS	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 A-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 A-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 A-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 A-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 A-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



Symbols

- !Local Egress Replication [29-12](#)
- !Multicast Replication Mode Detection enhancement [29-11](#)

Numerics

- 4K VLANs (support for 4,096 VLANs) [15-2](#)
- 802.10 SAID (default) [15-6](#)
- 802.1Q
 - encapsulation [11-3](#)
 - Layer 2 protocol tunneling
 - See Layer 2 protocol tunneling
 - mapping to ISL VLANs [15-13](#), [15-16](#)
 - trunks [11-2](#)
 - restrictions [11-5](#)
 - tunneling
 - configuration guidelines [18-3](#)
 - configuring tunnel ports [18-6](#)
 - overview [18-1](#)
- 802.1Q Ethertype
 - specifying custom [11-15](#)
- 802.1s
 - See MST
- 802.1w
 - See MST
- 802.1X
 - See port-based authentication
- 802.3ad
 - See LACP
- 802.3Z Flow Control [10-13](#)

A

- AAA [34-1](#), [35-1](#), [37-1](#), [45-1](#)
- abbreviating commands [2-5](#)
- access control entries and lists [34-1](#), [35-1](#), [37-1](#), [45-1](#)
- access lists
 - using with WCCP [55-11](#)
- access port, configuring [11-14](#)
- ACEs and ACLs [34-1](#), [35-1](#), [37-1](#), [45-1](#)
- acronyms, list of [A-1](#)
- addresses
 - IP, see IP addresses
 - MAC, see MAC addresses
- advertisements, VTP [14-3](#)
- aggregate label [25-2](#), [25-4](#)
- aggregate policing
 - see QoS policing
- aging time
 - accelerated
 - for MSTP [20-25](#)
 - maximum
 - for MSTP [20-26](#)
- aging-time
 - IP MLS [50-13](#)
- alarms
 - major [53-12](#)
 - minor [53-12](#)
- any transport over MPLS (AToM) [25-13](#)
 - compatibility with previous releases of AToM [25-15](#)
 - Ethernet over MPLS [25-16](#)
- ARP ACL [42-53](#)
- ARP spoofing [39-1](#)
- AToM [25-13](#)

audience [i](#)

authentication
 See also port-based authentication

Authentication, Authorization, and Accounting
 See AAA

Authentication, Authorization, and Accounting
 (AAA) [37-1](#)

authorized ports with 802.1X [46-4](#)

auto-sync command [8-6, 8-7, 9-7, 9-8](#)

auxiliary VLAN
 See voice VLAN

B

BackboneFast
 See STP BackboneFast

backup interfaces
 See Flex Links

binding database, DHCP snooping
 See DHCP snooping binding database

binding table, DHCP snooping
 See DHCP snooping binding database

blocking floods [41-1](#)

blocking state, STP [21-7](#)

boot bootldr command [3-26](#)

boot command [3-22](#)

boot config command [3-26](#)

boot system command [3-21, 3-26](#)

boot system flash command [3-22](#)

BPDU
 RSTP format [20-13](#)

BPDU guard
 See STP BPDU guard

bridge groups [23-2](#)

bridge ID
 See STP bridge ID

bridge priority, STP [21-29](#)

bridge protocol data units
 see BPDUs

bridging [23-2](#)

broadcast storms
 see traffic-storm control

C

cache engine clusters [55-1](#)

cache engines [55-1](#)

cache farms
 See cache engine clusters

cautions for passwords
 encrypting [3-17](#)
 TACACS+ [3-17](#)

CDP
 configuration task lists [48-1](#)
 enabling on an interface [48-2](#)
 monitoring and maintaining [48-3](#)
 overview [48-1](#)

cdp enable command [48-2](#)

CEF [27-1](#)
 configuring
 MSFC2 [27-5](#)
 supervisor engine [27-4](#)
 examples [27-3](#)
 Layer 3 switching [27-2](#)
 packet rewrite [27-2](#)

CEF for PFC2
 See CEF

CGMP [31-8](#)

channel-group group
 command [13-8, 13-12](#)
 command example [13-8](#)

checking
 configuration, system [3-10](#)

Cisco Cache Engines [55-2](#)

Cisco Discovery Protocol
 See CDP

Cisco Express Forwarding [25-3](#)

Cisco Group Management Protocol

- See CGMP
- Cisco IOS Unicast Reverse Path Forwarding [34-2](#)
- CiscoView [1-2](#)
- CIST [21-15](#)
- CIST regional root
 - See MSTP
- CIST root
 - See MSTP
- class command [42-58](#)
- class-map command [42-49](#)
- class map configuration [42-54](#)
- clear cdp counters command [48-3](#)
- clear cdp table command [48-3](#)
- clear counters command [10-17](#)
- clear interface command [10-18](#)
- clear mls ip multicast statistics command
 - clears IP MMLS statistics [29-27](#)
- CLI
 - accessing [2-1](#)
 - backing out one level [2-5](#)
 - console configuration mode [2-5](#)
 - getting list of commands [2-5](#)
 - global configuration mode [2-5](#)
 - history substitution [2-3](#)
 - interface configuration mode [2-5](#)
 - privileged EXEC mode [2-5](#)
 - ROM monitor [2-6](#)
 - software basics [2-4](#)
- command line processing [2-3](#)
- commands, getting list of [2-5](#)
- Common and Internal Spanning Tree
 - See also CIST [21-15](#)
- Common Spanning Tree
 - See CST [21-15](#)
- community ports [16-1](#)
- community VLANs [16-2](#)
- Concurrent routing and bridging (CRB) [23-2](#)
- CONFIG_FILE environment variable
 - configuration file, viewing [3-26](#)
 - description [3-25](#)
 - config-register command [3-23](#)
 - config terminal command [3-10](#)
 - configuration
 - file, saving [3-11](#)
 - interfaces [3-8 to 3-9](#)
 - register
 - changing settings [3-23](#)
 - configuration [3-21 to 3-24](#)
 - settings at startup [3-22](#)
 - configuration example
 - EoMPLS port mode for SUP720-3BXL-based system [25-17, 25-20](#)
 - EoMPLS VLAN mode for SUP720-3BXL-based system [25-17](#)
 - configuration register boot field
 - listing value [3-24](#)
 - modification tasks [3-23](#)
 - configure command [3-9](#)
 - configure terminal command [3-23, 10-2](#)
 - configuring [42-57](#)
 - global parameters
 - procedure [3-3](#)
 - sample configuration [3-3 to 3-8](#)
 - interfaces [3-8 to 3-9](#)
 - using configuration mode [3-10](#)
 - console configuration mode [2-5](#)
 - control plane policing
 - See CoPP
 - CoPP
 - applying QoS service policy to control plane [37-31](#)
 - configuring
 - ACLs to match traffic [37-31](#)
 - enabling MLS QoS [37-31](#)
 - packet classification criteria [37-31](#)
 - service-policy map [37-31](#)
 - control plane configuration mode
 - entering [37-31](#)
 - displaying

- dynamic information [37-32](#)
- number of conforming bytes and packets [37-32](#)
- rate information [37-22](#)
- entering control plane configuration mode [37-31](#)
- monitoring statistics [37-32](#)
- overview [37-29](#)
- packet classification guidelines [37-31](#)
- traffic classification
 - defining [37-33](#)
 - guidelines [37-34](#)
 - overview [37-33](#)
 - sample ACLs [37-34](#)
 - sample classes [37-33](#)
- copy running-config startup-config command [3-11](#)
- copy system
 - running-config nvram
 - startup-config command [3-26](#)
- CoS
 - override priority [17-7, 17-8](#)
- counters
 - clearing interface [10-17, 10-18](#)
- CST [21-15](#)
 - common spanning tree [21-17](#)

D

- dCEF [27-4, 27-5](#)
- debug commands
 - IP MMLS [29-26](#)
- DEC spanning-tree protocol [23-2](#)
- default configuration
 - 802.1X [46-5](#)
 - dynamic ARP inspection [39-5](#)
 - Flex Links [12-2](#)
 - IP MMLS [29-7](#)
 - MSTP [20-16](#)
 - supervisor engine [3-1](#)
 - UDLD [49-3](#)
 - voice VLAN [17-4](#)
 - VTP [14-5](#)
- default gateway, configuring [3-11](#)
- default NDE configuration [50-8](#)
- default VLAN [11-10](#)
- deficit weighted round robin [42-86](#)
- denial of service protection
 - See DoS protection
- description command [10-16](#)
- destination-ip flow mask [50-3](#)
- destination-source-ip flow mask [50-3](#)
- DHCP binding database
 - See DHCP snooping binding database
- DHCP binding table
 - See DHCP snooping binding database
- DHCP option 82
 - circuit ID suboption [38-3](#)
 - configuration guidelines [38-5](#)
 - overview [38-2](#)
 - packet format, suboption
 - circuit ID [38-3](#)
 - remote ID [38-3](#)
 - remote ID suboption [38-3](#)
- DHCP snooping
 - binding database
 - See DHCP snooping binding database
 - configuration guidelines [38-5](#)
 - configuring [38-6](#)
 - default configuration [38-5](#)
 - displaying binding tables [38-14](#)
 - enabling [38-7, 38-8, 38-9, 38-10](#)
 - enabling the database agent [38-11](#)
 - message exchange process [38-3](#)
 - option 82 data insertion [38-2](#)
 - overview [38-1](#)
 - Snooping database agent [38-4](#)
 - trusted interface [38-2](#)
 - untrusted interface [38-2](#)
 - untrusted messages [38-1](#)
- DHCP snooping binding database

- described [38-1](#)
- entries [38-1](#)
- DHCP snooping binding table
 - See DHCP snooping binding database
- DHCP Snooping Database Agent
 - adding to the database (example) [38-14](#)
 - enabling (example) [38-11](#)
 - overview [38-4](#)
 - reading from a TFTP file (example) [38-13](#)
- differentiated services codepoint
 - See QoS DSCP
- DiffServ
 - configuring short pipe mode [43-34](#)
 - configuring uniform mode [43-38](#)
 - short pipe mode [43-31](#)
 - uniform mode [43-32](#)
- DiffServ tunneling modes [43-4](#)
- Disabling PIM Snooping Designated Router Flooding [32-6](#)
- distributed Cisco Express Forwarding
 - See dCEF
- documentation, related [iv](#)
- document organization [i](#)
- DoS protection
 - monitoring packet drop statistics [37-7](#)
 - using monitor session commands [37-25, 37-26](#)
 - using NetFlow commands [37-24](#)
 - using VACL capture [37-27](#)
- Supervisor Engine 2
 - ARP throttling [37-5](#)
 - configuration guidelines and restrictions [37-22](#)
 - FIB rate limiting [37-4](#)
 - QoS ACLs [37-3](#)
 - recommendations [37-2](#)
 - security ACLs [37-2](#)
 - traffic storm control [37-5](#)
- Supervisor Engine 720 [37-10](#)
 - default configurations [37-22](#)
 - egress ACL bridget packet rate limiters [37-8, 37-16](#)
 - FIB glean rate limiters [37-17](#)
 - FIB receive rate limiters [37-9, 37-17](#)
 - ICMP redirect rate limiters [37-18](#)
 - IGMP unreachable rate limiters [37-17](#)
 - ingress ACL bridget packet rate limiters [37-8, 37-16](#)
 - IP errors rate limiters [37-10, 37-20](#)
 - IPv4 multicast rate limiters [37-10, 37-20](#)
 - IPv6 multicast rate limiters [37-21](#)
 - Layer 2 PDU rate limiters [37-10, 37-19](#)
 - Layer 2 protocol tunneling rate limiters [37-10, 37-19](#)
 - MTU failure rate limiters [37-19](#)
 - multicast directly connected rate limiters [37-20](#)
 - multicast FIB miss rate limiters [37-20](#)
 - multicast IGMP snooping rate limiters [37-10, 37-19](#)
 - network under SYN attack [37-13](#)
 - QoS ACLs [37-12](#)
 - security ACLs [37-11](#)
 - TCP intercept [37-6, 37-13](#)
 - traffic storm control [37-13](#)
 - TTL failure rate limiter [37-17](#)
 - uRPF check [37-12](#)
 - uRPF failure rate limiters [37-16](#)
 - VACL log rate limiters [37-10, 37-18](#)
 - Supervisor Engine 720 Layer 3 security features rate limiters [37-9, 37-18](#)
 - understanding how it works [37-2](#)
- DSCP
 - See QoS DSCP
- duplex command [10-8, 10-9](#)
- duplex mode
 - configuring interface [10-7](#)
- DWRR [42-86](#)
- dynamic ARP inspection
 - ARP cache poisoning [39-2](#)
 - ARP requests, described [39-1](#)
 - ARP spoofing attack [39-2](#)
- clearing
 - log buffer [39-16](#)
 - statistics [39-15](#)

- configuration guidelines [39-5](#)
- configuring
 - log buffer [39-13, 39-14](#)
 - logging system messages [39-13](#)
 - rate limit for incoming ARP packets [39-4, 39-9](#)
- default configuration [39-5](#)
- denial-of-service attacks, preventing [39-9](#)
- described [39-1](#)
- DHCP snooping binding database [39-3](#)
- displaying
 - ARP ACLs [39-15](#)
 - configuration and operating state [39-15](#)
 - log buffer [39-16](#)
 - statistics [39-15](#)
 - trust state and rate limit [39-15](#)
- error-disabled state for exceeding rate limit [39-4](#)
- function of [39-2](#)
- interface trust states [39-3](#)
- log buffer
 - clearing [39-16](#)
 - configuring [39-13, 39-14](#)
 - displaying [39-16](#)
- logging of dropped packets, described [39-4](#)
- logging system messages
 - configuring [39-13](#)
- man-in-the middle attack, described [39-2](#)
- network security issues and interface trust states [39-3](#)
- priority of ARP ACLs and DHCP snooping entries [39-4](#)
- rate limiting of ARP packets
 - configuring [39-9](#)
 - described [39-4](#)
 - error-disabled state [39-4](#)
- statistics
 - clearing [39-15](#)
 - displaying [39-15](#)
- validation checks, performing [39-11](#)
- Dynamic Host Configuration Protocol snooping
 - See DHCP snooping

E

- Egress ACL support for remarked DSCP [42-20](#)
- egress ACL support for remarked DSCP [42-45](#)
- egress PFC QoS [42-20](#)
- egress replication performance improvement [29-12](#)
- Embedded CiscoView [1-2](#)
- enable command [3-10, 3-23](#)
- enable mode [2-5](#)
- enable sticky secure MAC address [47-8](#)
- enabling
 - IP MMLS
 - on router interfaces [29-10](#)
- encapsulation [11-3](#)
- enhanced interface range command [10-4](#)
- environmental monitoring
 - LED indications [53-12](#)
 - SNMP traps [53-12](#)
 - supervisor engine and switching modules [53-12](#)
 - Syslog messages [53-12](#)
 - using CLI commands [53-10](#)
- environment variables
 - CONFIG_FILE [3-25](#)
 - controlling [3-26](#)
 - viewing [3-26](#)
- EoMPLS [25-14](#)
 - configuring [25-16](#)
 - configuring VLAN mode [25-16](#)
 - guidelines and restrictions [25-14](#)
 - port mode [25-16](#)
 - port mode configuration guidelines [25-20](#)
 - VLAN mode [25-16](#)
- erase startup-config command
 - configuration files cleared with [3-14](#)
- ERSPAN [51-1](#)
- EtherChannel
 - channel-group group
 - command [13-8, 13-12](#)
 - command example [13-8](#)

- configuration guidelines [13-5](#)
- configuring
 - Layer 2 [13-8](#)
- configuring (tasks) [13-6](#)
- DFC restriction, see CSCdt27074 in the Release Notes
- interface port-channel
 - command example [13-7](#)
- interface port-channel (command) [13-7](#)
- lacp system-priority
 - command example [13-10](#)
- Layer 2
 - configuring [13-8](#)
- load balancing
 - configuring [13-11](#)
 - understanding [13-5](#)
- modes [13-2](#)
- PAgP
 - Understanding [13-3](#)
- port-channel interfaces [13-4](#)
- port-channel load-balance
 - command [13-10, 13-11](#)
 - command example [13-11](#)
- STP [13-4](#)
- switchport trunk encapsulation dot1q [13-5](#)
- understanding [13-1](#)

EtherChannel Guard

- See STP EtherChannel Guard

EtherChannel Min-Links [13-12](#)

Ethernet

- setting port duplex [10-14](#)

Ethernet over MPLS (EoMPLS) configuration

- EoMPLS port mode for SUP720-3BXL-based system [25-20](#)
- EoMPLS VLAN mode for SUP720-3BXL-based system [25-17](#)

examples

- configuration
 - interface [3-8 to 3-9](#)
 - software configuration register [3-21 to 3-24](#)

- configuring global parameters [3-3](#)
- EXP mutation [43-4](#)
- extended range VLANs [15-2](#)
 - See VLANs
- extended system ID
 - MSTP [20-19](#)
- Extensible Authentication Protocol over LAN [46-1](#)

F

fabric switching mode

- See switch fabric module

fabric switching-mode allow dcef-only command on Supervisor Engine 720 [7-2, 9-2, 9-5](#)

fall-back bridging [23-2](#)

fastethernet [10-2](#)

fiber-optic, detecting unidirectional links [49-1](#)

FIB TCAM [25-2](#)

filters, NDE

- destination host filter, specifying [50-20](#)
- destination TCP/UDP port, specifying [50-20](#)
- overview [50-7](#)
- protocol [50-21](#)
- source host and destination TCP/UDP port [50-20](#)

Flash memory

- configuration process [3-25](#)
- configuring router to boot from [3-25](#)
- loading system image from [3-24](#)
- security precautions [3-25](#)
- write protection [3-25](#)

Flex Links [12-1](#)

- configuration guidelines [12-2](#)
- configuring [12-3](#)
- default configuration [12-2](#)
- description [12-1](#)
- monitoring [12-3](#)

flood blocking [41-1](#)

flow control [10-13](#)

flow masks

IP MLS

- destination-ip [50-3](#)
- destination-source-ip [50-3](#)
- interface-destination-source-ip [50-3](#)
- ip-full [50-3](#)
- ip-interface-full [50-3](#)
- minimum [50-12](#)
- overview [50-3](#)

flows

- IP MMLS
 - completely and partially switched [29-3](#)

forward-delay time

- MSTP [20-25](#)

forward-delay time, STP [21-31](#)

frame distribution

- See EtherChannel load balancing

G

gateway, configuring [3-11](#)

global configuration mode [2-5](#)

global parameters, configuring [3-3](#)

H

hardware Layer 3 switching

- guidelines [27-4](#)

hello time

- MSTP [20-24](#)

hello time, STP [21-31](#)

High Capacity Power Supply Support [53-4](#)

history

- CLI [2-3](#)

I

I-BPDU [21-15](#)

ICMP unreachable messages [35-1](#)

IEEE 802.10 SAID (default) [15-6](#)

IEEE 802.1Q

- See 802.1Q

IEEE 802.1Q Ethertype

- specifying custom [11-15](#)

IEEE 802.1s

- See MST

IEEE 802.1w

- See MST
- See RSTP

IEEE 802.3ad

- See LACP

IEEE 802.3Z Flow Control [10-13](#)

IEEE bridging protocol [23-2](#)

IGMP

- configuration guidelines [30-7, 31-7](#)
- enabling [31-10](#)
- Internet Group Management Protocol [31-1](#)
- join messages [31-2](#)
- leave processing
 - enabling [31-12](#)
- queries [31-3](#)
- query interval
 - configuring [31-11](#)
- snooping
 - fast leave [31-5](#)
 - joining multicast group [31-2](#)
 - leaving multicast group [31-4](#)
 - understanding [31-2](#)
- snooping querier
 - enabling [31-9](#)
 - understanding [31-2](#)

IGMPv3 [29-9](#)

IGMP v3lite [29-9](#)

IGRP, configuring [3-7](#)

Integrated routing and bridging (IRB) [23-2](#)

interface

- command [3-10](#)
- configuration [3-8 to 3-9](#)

- configuration mode [2-5](#)
- Layer 2 modes [11-4](#)
- number [10-2](#)
- parameters, configuring [3-8](#)
- interface-destination-source-ip flow mask [50-3](#)
- interface port-channel
 - command example [13-7](#)
- interface port-channel (command) [13-7](#)
- interfaces
 - configuring [10-2](#)
 - configuring, duplex mode [10-7](#)
 - configuring, speed [10-7](#)
 - configururing, overview [10-1](#)
 - counters, clearing [10-17, 10-18](#)
 - descriptive name, adding [10-15](#)
 - displaying information about [10-17](#)
 - maintaining [10-16](#)
 - monitoring [10-16](#)
 - naming [10-15](#)
 - range of [10-4](#)
 - restarting [10-18](#)
 - shutting down
 - task [10-18](#)
- interfaces command [10-1, 10-2](#)
- interfaces range command [10-4](#)
- interfaces range macro command [10-5](#)
- Interior Gateway Routing Protocol
 - See IGRP, configuring
- Internal Sub Tree Protocol
 - See ISTP [21-15](#)
- Internet Group Management Protocol
 - See IGMP
- IP
 - default gateway, configuring [3-11](#)
 - static routes [3-12](#)
- IP accounting, IP MMLS and [29-8](#)
- IP addresses
 - assigned by BOOTP protocol [3-13](#)
 - set to default [3-14](#)
- IP CEF
 - topology (figure) [27-3](#)
- ip flow-export destination command [50-16](#)
- ip flow-export source command [50-16, 50-17, 50-18, 56-3, 56-4](#)
- ip-full flow mask [50-3](#)
- ip http server [1-1](#)
- ip-interface-full flow mask [50-3](#)
- IP MLS
 - aging-time [50-13](#)
 - flow masks
 - destination-ip [50-3](#)
 - destination-source-ip [50-3](#)
 - interface-destination-source-ip [50-3](#)
 - ip-full [50-3](#)
 - ip-interface-full [50-3](#)
 - minimum [50-12](#)
 - overview [50-3](#)
- NDE
 - See NDE
- IP MMLS
 - cache, overview [29-2](#)
 - configuration guideline [29-7](#)
 - debug commands [29-26](#)
 - default configuration [29-7](#)
 - enabling
 - on router interfaces [29-10](#)
 - flows
 - completely and partially switched [29-3](#)
 - Layer 3 MLS cache [29-2](#)
 - overview [29-2](#)
 - packet rewrite [29-3](#)
 - router
 - displaying interface information [29-17](#)
 - enabling globally [29-9](#)
 - enabling on interfaces [29-10](#)
 - multicast routing table, displaying [29-20](#)
 - PIM, enabling [29-10](#)
 - switch

- statistics, clearing [29-27](#)
 - unsupported features [29-8](#)
- IP multicast
 - IGMP snooping and [31-9](#)
 - MLDv2 snooping and [30-9](#)
 - overview [31-1](#)
- IP multicast MLS
 - See IP MMLS
- ip multicast-routing command
 - enabling IP multicast [29-9](#)
- IP phone
 - configuring [17-5](#)
- ip pim command
 - enabling IP PIM [29-10](#)
- IPsec [45-2](#)
- IP unnumbered [23-1](#)
- IPv4 Multicast over Point-to-Point GRE Tunnels [1-4](#)
- IPv4 Multicast VPN [26-1](#)
- IPv6 Multicast PFC3 and DFC3 Layer 3 Switching [28-1](#)
- IPv6 QoS [42-2, 42-3, 42-17, 42-37](#)
- ip wccp version command [55-8](#)
- ISL encapsulation [11-3](#)
- ISL trunks [11-2](#)
- isolated port [16-1](#)
- isolated VLANs [16-2](#)
- ISTP [21-15](#)

J

- join messages, IGMP [31-2](#)
- jumbo frames [10-10](#)

K

- keyboard shortcuts [2-3](#)

L

- label edge router [25-2](#)
- label switched path [25-16](#)
- label switch router [25-2, 25-3](#)
- LACP
 - system ID [13-4](#)
- Layer 2
 - configuring interfaces [11-6](#)
 - access port [11-14](#)
 - trunk [11-7](#)
 - defaults [11-5](#)
 - interface modes [11-4](#)
 - show interfaces [10-12, 10-13, 11-7, 11-12](#)
 - switching
 - understanding [11-1](#)
 - trunks
 - understanding [11-2](#)
 - VLAN
 - interface assignment [15-12](#)
- Layer 2 Interfaces
 - configuring [11-1](#)
- Layer 2 protocol tunneling
 - configuring Layer 2 tunnels [19-2](#)
 - overview [19-1](#)
- Layer 2 Traceroute [57-1](#)
- Layer 2 traceroute
 - and ARP [57-2](#)
 - and CDP [57-1](#)
 - described [57-1](#)
 - IP addresses and subnets [57-2](#)
 - MAC addresses and VLANs [57-2](#)
 - multicast traffic [57-2](#)
 - multiple devices on a port [57-2](#)
 - unicast traffic [57-1](#)
 - usage guidelines [57-1](#)
- Layer 3
 - IP MMLS and MLS cache [29-2](#)
- Layer 3 switched packet rewrite

- CEF [27-2](#)
- Layer 3 switching
 - CEF [27-2](#)
- Layer 4 port operations (ACLs) [35-6](#)
- leave processing, IGMP
 - enabling [31-12](#)
- leave processing, MLDv2
 - enabling [30-12](#)
- LERs [43-2](#), [43-6](#), [43-7](#)
- Link Failure
 - detecting unidirectional [20-8](#)
- link negotiation [10-8](#)
- link redundancy
 - See Flex Links
- Load Balancing [25-7](#)
- load balancing [21-15](#)
- logical operation unit
 - See LOU
- loop guard
 - See STP loop guard
- LOU
 - description [35-6](#)
 - determining maximum number of [35-6](#)
- LSRs [43-2](#), [43-6](#)

M

- MAC address
 - adding to BOOTP configuration file [3-14](#)
- MAC address-based blocking [34-1](#)
- main-cpu command [8-6](#), [8-7](#), [9-7](#), [9-8](#)
- mapping 802.1Q VLANs to ISL VLANs [15-13](#), [15-16](#)
- markdown
 - see QoS markdown
- marking
 - see QoS
- match protocol [42-38](#)
- maximum aging time
 - MSTP [20-26](#)
 - maximum aging time, STP [21-32](#)
 - maximum hop count, MSTP [20-26](#)
 - microflow policing
 - see QoS policing
 - Min-Links [13-12](#)
 - MLD
 - report [30-4](#)
 - MLD snooping
 - query interval
 - configuring [30-11](#)
 - MLDv2 [30-1](#)
 - enabling [30-9](#)
 - leave processing
 - enabling [30-12](#)
 - queries [30-4](#)
 - snooping
 - fast leave [30-6](#)
 - joining multicast group [30-4](#)
 - leaving multicast group [30-6](#)
 - understanding [30-1](#)
 - snooping querier
 - enabling [30-8](#)
 - understanding [30-1](#)
 - MLDv2 Snooping [30-1](#)
 - MLS
 - configuring threshold [29-14](#)
 - MSFC
 - threshold [29-14](#)
 - mls aging command
 - configuring IP MLS [50-13](#)
 - mls flow command
 - configuring IP MLS [50-12](#), [50-15](#)
 - mls ip multicast command
 - enabling IP MMLS [29-11](#), [29-12](#), [29-13](#), [29-14](#), [29-15](#), [29-16](#), [29-22](#), [29-23](#), [29-24](#)
 - mls nde flow command
 - configuring a host and port filter [50-20](#)
 - configuring a host flow filter [50-20](#)
 - configuring a port filter [50-20](#)

- configuring a protocol flow filter [50-21](#)
- mls nde sender command [50-14](#)
- monitoring
 - Flex Links [12-3](#)
- MPLS [25-2](#)
 - aggregate label [25-2](#)
 - any transport over MPLS [25-13](#)
 - basic configuration [25-8](#)
 - core [25-3](#)
 - DiffServ Tunneling Modes [43-30](#)
 - egress [25-3](#)
 - experimental field [43-3](#)
 - guidelines and restrictions [25-7](#)
 - ingress [25-3](#)
 - IP to MPLS path [25-3](#)
 - labels [25-2](#)
 - Layer 2 VPN load balancing [25-8](#)
 - MPLS to IP path [25-3](#)
 - MPLS to MPLS path [25-3](#)
 - nonaggregate lable [25-2](#)
 - QoS default configuration [43-15](#)
 - VPN [43-12](#)
 - VPN guidelines and restrictions [25-11](#)
- mpls l2 transport route command [25-15](#)
- MPLS QoS
 - Classification [43-2](#)
 - Class of Service [43-2](#)
 - commands [43-16](#)
 - configuring a class map [43-20](#)
 - configuring a policy map [43-22](#)
 - configuring egress EXP mutation [43-28](#)
 - configuring EXP Value Maps [43-29](#)
 - Differentiated Services Code Point [43-2](#)
 - displaying a policy map [43-27](#)
 - E-LSP [43-2](#)
 - enabling QoS globally [43-18](#)
 - EXP bits [43-2](#)
 - features [43-3](#)
 - IP Precedence [43-2](#)
 - QoS Tags [43-2](#)
 - queueing-only mode [43-19](#)
- MPLS QoS configuration
 - class map to classify MPLS packets [43-20](#)
- MPLS VPN
 - limitations and restrictions [25-11](#)
- MST [21-15](#)
 - boundary ports [21-19](#)
 - configuration [21-18](#)
 - configuring [21-33](#)
 - edge ports [21-19](#)
 - enabling [21-33](#)
 - hop count [21-20](#)
 - instances [21-18](#)
 - interoperability [21-16](#)
 - interoperability with PVST+ [21-15](#)
 - link type [21-20](#)
 - master [21-19](#)
 - message age [21-20](#)
 - regions [21-18](#)
- MSTP
 - boundary ports
 - configuration guidelines [20-16](#)
 - described [20-6](#)
 - CIST, described [20-3](#)
 - CIST regional root [20-3](#)
 - CIST root [20-5](#)
 - configuration guidelines [20-16](#)
 - configuring
 - forward-delay time [20-25](#)
 - hello time [20-24](#)
 - link type for rapid convergence [20-26](#)
 - maximum aging time [20-26](#)
 - maximum hop count [20-26](#)
 - MST region [20-17](#)
 - neighbor type [20-27](#)
 - path cost [20-22](#)
 - port priority [20-21](#)
 - root switch [20-19](#)

- secondary root switch [20-20](#)
- switch priority [20-23](#)
- CST
 - defined [20-3](#)
 - operations between regions [20-4](#)
- default configuration [20-16](#)
- displaying status [20-28](#)
- enabling the mode [20-17](#)
- extended system ID
 - effects on root switch [20-19](#)
 - effects on secondary root switch [20-20](#)
 - unexpected behavior [20-19](#)
- IEEE 802.1s
 - implementation [20-7](#)
 - port role naming change [20-7](#)
 - terminology [20-5](#)
- interoperability with IEEE 802.1D
 - described [20-9](#)
 - restarting migration process [20-28](#)
- IST
 - defined [20-3](#)
 - master [20-3](#)
 - operations within a region [20-3](#)
- mapping VLANs to MST instance [20-17](#)
- M-record [21-15](#)
- MST region
 - CIST [20-3](#)
 - configuring [20-17](#)
 - described [20-2](#)
 - hop-count mechanism [20-6](#)
 - IST [20-3](#)
 - supported spanning-tree instances [20-2](#)
- M-tree [21-15](#)
- overview [20-1](#)
- root switch
 - configuring [20-19](#)
 - effects of extended system ID [20-19](#)
 - unexpected behavior [20-19](#)
- status, displaying [20-28](#)
- MTU size (default) [15-6](#)
- multicast
 - IGMP snooping and [31-9](#)
 - MLDv2 snooping and [30-9](#)
 - NetFlow statistics [50-9](#)
 - non-RPF [29-5](#)
 - overview [31-1](#)
 - PIM snooping [32-4](#)
 - RGMP [33-1](#)
- multicast, displaying routing table [29-20](#)
- Multicast enhancement - egress replication performance improvement [29-12](#)
- Multicast Enhancement - Replication Mode Detection [29-11](#)
- multicast flood blocking [41-1](#)
- multicast groups
 - joining [31-2](#)
 - leaving [30-6, 31-4](#)
- multicast groups, IPv6
 - joining [30-4](#)
- Multicast Listener Discovery version 2
 - See MLDv2
- multicast multilayer switching
 - See IPv4 MMLS
- multicast RPF [29-2](#)
- multicast storms
 - see traffic-storm control
- Multilayer MAC ACL QoS Filtering [42-50](#)
- multilayer switch feature card
 - see MSFC
- multiple forwarding paths [21-15](#)
- multiple path RPF check [34-2](#)
- Multiple Spanning Tree
 - See MST
- Multiple Spanning Tree Protocol
 - See MSTP [21-15](#)

N

native VLAN [11-10](#)

NBAR [42-1](#), [42-38](#)

NDE

configuration, displaying [50-21](#)

displaying configuration [50-21](#)

enabling [50-9](#)

filters

destination host, specifying [50-20](#)

destination TCP/UDP port, specifying [50-20](#)

overview [50-7](#)

protocol, specifying [50-21](#)

source host and destination TCP/UDP port,
specifying [50-20](#)

multicast [50-9](#)

overview [50-2](#)

specifying

destination host filters [50-20](#)

destination TCP/UDP port filters [50-20](#)

protocol filters [50-21](#)

NDE configuration, default [50-8](#)

NDE version 8 [50-2](#)

NetFlow and NDE for Ingress Bridged IP Traffic [50-17](#)

NetFlow Data Export

See NDE

Netflow Multiple Export Destinations [50-17](#)

NetFlow version 9 [50-3](#)

Network-Based Application Recognition [42-1](#)

network fault tolerance [21-15](#)

network management

configuring [48-1](#)

nonaggregate label [25-2](#), [25-4](#)

non-RPF multicast [29-5](#)

Nonstop Forwarding

See NSF

nonvolatile random-access memory

See NVRAM

normal-range VLANs

See VLANs

NSF [7-1](#)

NSF with SSO does not support IPv6 multicast
traffic. [7-1](#)

NVRAM

saving settings [3-11](#)

O

OIR [10-16](#)

online diagnostics

configuring [54-2](#)

diagnostic sanity check [54-11](#)

memory tests [54-10](#)

overview [54-1](#)

running tests [54-6](#)

schedule switchover [54-10](#)

understanding [54-1](#)

online insertion and removal

See OIR

operating system image

See system image

out of profile

see QoS out of profile

P

packet burst [37-16](#)

packet rewrite

CEF [27-2](#)

IP MMLS and [29-3](#)

packets

multicast [36-4](#)

PAGP

understanding [13-3](#)

passwords

configuring

enable password [3-15](#)

enable secret [3-15](#)

- line password [3-16](#)
- static enable password [3-15](#)
- TACACS+ [3-16](#)
- TACACS+ (caution) [3-17](#)
- encrypting [3-17](#)
 - (caution) [3-17](#)
- recovering lost enable passwords [3-19](#)
- path cost
 - MSTP [20-22](#)
- PBR [1-4, 23-4](#)
- PFC2
 - NetFlow
 - table, displaying entries [27-5](#)
- PFC3BXL
 - hardware features [25-4](#)
 - MPLS guidelines and restrictions [25-7](#)
 - MPLS label switching [25-1](#)
 - MPLS supported commands [25-7](#)
 - recirculation [25-4](#)
 - supported Cisco IOS features [25-5](#)
 - VPN supported commands [25-11](#)
 - VPN switching [25-9](#)
- PIM, IP MMLS and [29-10](#)
- PIM snooping
 - designated router flooding [32-6](#)
 - enabling globally [32-5](#)
 - enabling in a VLAN [32-5](#)
 - overview [32-4](#)
- police command [42-60](#)
- policing
 - See QoS policing
- policy [42-49](#)
- policy-based routing
 - See PBR
- policy map [42-57](#)
 - attaching to an interface [42-64](#)
- policy-map command [42-50, 42-57](#)
- Port Aggregation Protocol
 - see PAgP
- port-based authentication
 - authentication server
 - defined [46-2](#)
 - RADIUS server [46-2](#)
 - client, defined [46-2](#)
 - configuration guidelines [46-6](#)
 - configuring
 - initializing authentication of a client [46-11](#)
 - manual reauthentication of a client [46-11](#)
 - quiet period [46-11](#)
 - RADIUS server [46-10](#)
 - RADIUS server parameters on the switch [46-8](#)
 - switch-to-authentication-server retransmission time [46-13](#)
 - switch-to-client EAP-request frame retransmission time [46-13](#)
 - switch-to-client frame-retransmission number [46-14](#)
 - switch-to-client retransmission time [46-12](#)
 - default configuration [46-5](#)
 - described [46-1](#)
 - device roles [46-2](#)
 - displaying statistics [46-15](#)
 - EAPOL-start frame [46-3](#)
 - EAP-request/identity frame [46-3](#)
 - EAP-response/identity frame [46-3](#)
 - enabling
 - 802.1X authentication [46-7, 46-8](#)
 - periodic reauthentication [46-10](#)
 - encapsulation [46-2](#)
 - initiation and message exchange [46-3](#)
 - method lists [46-7](#)
 - ports
 - authorization state and dot1x port-control command [46-4](#)
 - authorized and unauthorized [46-4](#)
 - resetting to default values [46-15](#)
 - switch
 - as proxy [46-2](#)
 - RADIUS client [46-2](#)
 - topologies, supported [46-4](#)

- port-based QoS features
 - see QoS
- port channel
 - switchport trunk encapsulation dot1q [13-5](#)
- port-channel
 - see EtherChannel
- port-channel load-balance
 - command [13-10, 13-11](#)
 - command example [13-10, 13-11](#)
- port cost, STP [21-28](#)
- port debounce timer
 - disabling [10-14](#)
 - displaying [10-14](#)
 - enabling [10-14](#)
- PortFast
 - See STP PortFast
- PortFast BPDU filtering
 - See STP PortFast BPDU filtering
- port mode [25-16](#)
- port negotiation [10-8](#)
- port priority
 - MSTP [20-21](#)
- port priority, STP [21-26](#)
- ports
 - secure [47-2](#)
 - setting the debounce timer [10-14](#)
- port security
 - aging [47-10, 47-11](#)
 - configuring [47-3](#)
 - default configuration [47-2](#)
 - described [47-1](#)
 - displaying [47-11](#)
 - enable sticky secure MAC address [47-8](#)
 - sticky MAC address [47-2](#)
 - violations [47-2](#)
- Port Security is supported on trunks [47-3, 47-4, 47-7, 47-9](#)
- port security on PVLAN ports [47-3](#)
- Port Security with Sticky Secure MAC Addresses [47-2](#)
- power management
 - enabling/disabling redundancy [53-2](#)
 - overview [53-1](#)
 - powering modules up or down [53-3](#)
 - system power requirements, nine-slot chassis [53-5](#)
- primary links [12-1](#)
- primary VLANs [16-2](#)
- priority
 - overriding CoS [17-7, 17-8](#)
- private VLANs [16-1](#)
 - community VLANs [16-2](#)
 - configuration guidelines [16-2](#)
 - configuring [16-5](#)
 - host ports [16-9](#)
 - promiscuous ports [16-10](#)
 - routing secondary VLAN ingress traffic [16-8](#)
 - secondary VLANs with primary VLANs [16-7](#)
 - VLANs as private [16-6](#)
- isolated VLANs [16-2](#)
- ports
 - community [16-1](#)
 - isolated [16-1](#)
 - promiscuous [16-1](#)
 - primary VLANs [16-2](#)
 - secondary VLANs [16-2](#)
 - trunks [16-2](#)
- privileged EXEC mode [2-5](#)
- privileges
 - changing default [3-18](#)
 - configuring
 - multiple levels [3-17](#)
 - privilege level [3-18](#)
 - exiting [3-19](#)
 - logging in [3-18](#)
- procedures
 - global parameters, configuring [3-3 to 3-8](#)
 - interfaces, configuring [3-8 to 3-9](#)
 - using configuration mode [3-10](#)
- promiscuous ports [16-1](#)
- protocol tunneling

See Layer 2 protocol tunneling [19-1](#)

pruning, VTP

See VTP, pruning

PVLANs

See private VLANs

PVRST

See Rapid-PVST [21-14](#)

Q

QoS

IPv6 [42-2, 42-3, 42-17, 42-37](#)

QoS ACL [42-16](#)

attaching [42-20](#)

QoS classification (definition) [42-3](#)

QoS congestion avoidance

definition [42-4](#)

QoS CoS

and ToS final L3 Switching Engine values [42-21](#)

and ToS final values from L3 Switching Engine [42-21](#)

definition [42-4](#)

port value, configuring [42-75](#)

QoS default configuration [42-24, 44-2](#)

QoS definitions [42-3](#)

QoS DSCP

definition [42-4](#)

internal values [42-16](#)

maps, configuring [42-70](#)

QoS dual transmit queue

thresholds

configuring [42-76, 42-80](#)

QoS Ethernet egress port

feature summary [42-12](#)

scheduling [42-23](#)

scheduling, congestion avoidance, and marking [42-10, 42-22](#)

QoS Ethernet ingress port

classification, marking, scheduling, and congestion avoidance [42-7](#)

feature summary [42-11](#)

marking, scheduling, congestion avoidance, and classification [42-12](#)

QoS feature set summary [42-11](#)

QoS final L3 Switching Engine CoS and ToS values [42-21](#)

QoS internal DSCP values [42-16](#)

QoS L3 Switching Engine

classification, marking, and policing [42-8, 42-15](#)

feature summary [42-11](#)

QoS labels (definition) [42-4](#)

QoS mapping

CoS values to DSCP values [42-67, 42-71](#)

DSCP markdown values [42-25, 42-72, 43-16](#)

DSCP mutation [42-66, 43-28](#)

DSCP values to CoS values [42-73](#)

IP precedence values to DSCP values [42-71](#)

QoS markdown [42-18](#)

QoS marking [42-23](#)

definition [42-3](#)

trusted ports [42-13](#)

untrusted ports [42-13](#)

QoS MSFC

marking [42-9](#)

QoS multilayer switch feature card [42-12](#)

QoS OSM egress port

feature summary [42-12](#)

QoS out of profile [42-18](#)

QoS policing

definition [42-4](#)

microflow, enabling for nonrouted traffic [42-43](#)

QoS policing rule [42-17](#)

aggregate [42-18](#)

creating [42-48](#)

microflow [42-18](#)

QoS port

trust state [42-74](#)

QoS port-based or VLAN-based [42-44](#)

QoS port keywords [42-12](#)

QoS queues

- transmit, allocating bandwidth between [42-86](#)
- QoS receive queue [42-13, 42-84](#)
 - drop thresholds [42-15](#)
- QoS scheduling (definition) [42-3](#)
- QoS single-receive, dual-transmit queue ports
 - configuring [42-82](#)
- QoS statistics data export [44-1](#)
 - configuring [44-2](#)
 - configuring destination host [44-7](#)
 - configuring time interval [44-6, 44-9](#)
- QoS strict priority receive queue [42-14](#)
- QoS ToS
 - and CoS final values from L3 Switching Engine [42-21](#)
 - definition [42-4](#)
- QoS traffic flow through QoS features [42-6](#)
- QoS transmit queue
 - size ratio [42-87, 42-88](#)
- QoS transmit queues [42-22, 42-82, 42-83](#)
- QoS trust-cos
 - port keyword [42-11, 42-12](#)
- QoS trust-dscp
 - port keyword [42-11, 42-12](#)
- QoS trust-ipprec
 - port keyword [42-11, 42-12](#)
- QoS untrusted port keyword [42-11, 42-12](#)
- QoS VLAN-based or port-based [42-20, 42-44](#)
- queries, IGMP [31-3](#)
- queries, MLDv2 [30-4](#)

R

- range
 - command [10-4](#)
 - macro [10-5](#)
 - of interfaces [10-4](#)
- rapid convergence [20-11](#)
- Rapid-PVST
 - enabling [21-32](#)
 - overview [21-14](#)
- Rapid Spanning Tree
 - See RSTP [21-13](#)
- Rapid Spanning Tree Protocol
 - See RSTP
- rapid spanning tree protocol [21-15](#)
- receive queues
 - see QoS receive queues
- recirculation [25-4](#)
- reduced MAC address [21-2](#)
- redundancy (NSF) [7-1](#)
 - configuring
 - BGP [7-14](#)
 - CEF [7-13](#)
 - EIGRP [7-19](#)
 - IS-IS [7-16](#)
 - OSPF [7-15](#)
 - configuring multicast NSF with SSO [7-12](#)
 - configuring supervisor engine [7-10](#)
 - routing protocols [7-4](#)
- redundancy (RPR+) [8-1, 9-1](#)
 - configuring [8-6, 9-7](#)
 - configuring supervisor engine [8-5, 9-6](#)
 - displaying supervisor engine configuration [8-8, 9-9](#)
 - redundancy command [8-6, 8-7, 9-7, 9-8](#)
 - route processor redundancy plus [9-3](#)
- redundancy (SSO)
 - redundancy command [7-11](#)
- related documentation [iv](#)
- reload command [3-23](#)
- Remote source-route bridging (RSRB) [23-2](#)
- Replication Mode Detection [29-11](#)
- report, MLD [30-4](#)
- reserved-range VLANs
 - See VLANs
- rewrite, packet
 - CEF [27-2](#)
 - IP MMLS [29-3](#)
- RGMP [33-1](#)
 - overview [33-1](#)

- packet types [33-2](#)
- RIF cache monitoring [10-17](#)
- rommon command [3-24](#)
- ROM monitor
 - boot process and [3-20](#)
 - CLI [2-6](#)
- root bridge, STP [21-24](#)
- root guard
 - See STP root guard
- root switch
 - MSTP [20-19](#)
- route processor redundancy
 - See redundancy (RPR+)
- router-port group management protocol
 - See RGMP
- routing table, multicast [29-20](#)
- RPF
 - failure [29-5](#)
 - multicast [29-2](#)
 - non-RPF multicast [29-5](#)
 - unicast [34-2](#)
- RPR+
 - See redundancy (RPR+)
- RPR and RPR+ support IPv6 multicast traffic [9-1](#)
- RSTP [21-15](#)
 - active topology [20-10](#)
 - BPDUs
 - format [20-13](#)
 - processing [20-14](#)
 - designated port, defined [20-10](#)
 - designated switch, defined [20-10](#)
 - interoperability with IEEE 802.1D
 - described [20-9](#)
 - restarting migration process [20-28](#)
 - topology changes [20-15](#)
 - overview [20-9](#)
 - port roles [21-13](#)
 - described [20-10](#)
 - synchronized [20-12](#)

- port states [21-14](#)
- proposal-agreement handshake process [20-11](#)
- rapid convergence
 - described [20-11](#)
 - edge ports and Port Fast [20-11](#)
 - point-to-point links [20-11, 20-26](#)
 - root ports [20-11](#)
- root port, defined [20-10](#)
- See also MSTP

S

- SAID [15-6](#)
- sample configuration [3-2 to 3-10](#)
- Sampled NetFlow
 - description [50-7](#)
- saving the configuration file [3-11](#)
- scheduling
 - see QoS
- secondary VLANs [16-2](#)
- Secure MAC Address Aging Type [47-10](#)
- secure ports, configuring [47-2](#)
- security
 - configuring [34-1, 35-1, 37-1, 45-1](#)
- security, port [47-1](#)
- security precautions with Flash memory card [3-25](#)
- serial interfaces
 - clearing [10-18](#)
 - synchronous
 - maintaining [10-18](#)
- service-policy command [42-50](#)
- service-policy input command [42-45, 42-64, 42-67, 42-69, 43-29](#)
- service-provider network, MSTP and RSTP [20-2](#)
- set power redundancy enable/disable command [53-2](#)
- setup command [3-2](#)
- short pipe mode
 - configuring [43-34](#)
- show boot command [3-26](#)

- show catalyst6000 chassis-mac-address command [21-3](#)
- show cdp command [48-2, 48-3](#)
- show cdp entry command [48-3](#)
- show cdp interface command [48-3](#)
- show cdp neighbors command [48-3](#)
- show cdp traffic command [48-3](#)
- show ciscoview package command [1-3](#)
- show ciscoview version command [1-3](#)
- show configuration command [10-15](#)
- show debugging command [48-3](#)
- show eobc command [10-17](#)
- show hardware command [10-3](#)
- show history command [2-4](#)
- show ibc command [10-17](#)
- show interfaces command [10-2, 10-12, 10-13, 10-15, 10-17, 11-7, 11-12](#)
 - clearing interface counters [10-17](#)
 - displaying, interface type numbers [10-2](#)
 - displaying, speed and duplex mode [10-9](#)
- show ip flow export command
 - displaying NDE export flow IP address and UDP port [50-18](#)
- show ip interface command
 - displaying IP MMLS interfaces [29-17](#)
- show ip mroute command
 - displaying IP multicast routing table [29-20](#)
- show ip pim interface command
 - displaying IP MMLS router configuration [29-17](#)
- show mls aging command [50-14](#)
- show mls entry command [27-5](#)
- show mls ip multicast group command
 - displaying IP MMLS group [29-21, 29-24](#)
- show mls ip multicast interface command
 - displaying IP MMLS interface [29-21, 29-24](#)
- show mls ip multicast source command
 - displaying IP MMLS source [29-21, 29-24](#)
- show mls ip multicast statistics command
 - displaying IP MMLS statistics [29-21, 29-24](#)
- show mls ip multicast summary
 - displaying IP MMLS configuration [29-21, 29-24](#)
- show mls nde command [50-21](#)
 - displaying NDE flow IP address [50-18](#)
- show mls rp command
 - displaying IP MLS configuration [50-13](#)
- show module command [8-8, 9-9](#)
- show protocols command [10-17](#)
- show rif command [10-17](#)
- show running-config command [3-10, 10-15, 10-17](#)
- show startup-config command [3-11](#)
- show version command [3-9, 3-23, 3-24, 10-17](#)
- shutdown command [10-18](#)
- shutdown interfaces
 - result [10-18](#)
- Single Spanning Tree
 - See SST [21-15](#)
- slot number, description [10-2](#)
- SNMP
 - support and documentation [1-1](#)
- snooping
 - See IGMP snooping
 - See MLDv2 snooping
- software configuration register functions [3-21 to 3-24](#)
- source-only-ip flow mask [50-3](#)
- source specific multicast with IGMPv3, IGMP v3lite, and URD [29-9](#)
- SPAN
 - configuration guidelines [51-6](#)
 - configuring [51-12](#)
 - sources [51-13, 51-15, 51-16, 51-18, 51-20](#)
 - VLAN filtering [51-22](#)
 - overview [51-1](#)
- SPAN Destination Port Permit Lists [51-13](#)
- spanning-tree backbonefast
 - command [22-13, 22-14](#)
 - command example [22-13, 22-14](#)
- spanning-tree cost
 - command [21-28](#)
 - command example [21-28, 21-29](#)

- spanning-tree portfast
 - command [22-8, 22-9](#)
 - command example [22-8](#)
- spanning-tree portfast bpdu-guard
 - command [22-11](#)
- spanning-tree port-priority
 - command [21-26, 21-27](#)
- spanning-tree protocol for bridging [23-2](#)
- spanning-tree uplinkfast
 - command [22-12](#)
 - command example [22-12](#)
- spanning-tree vlan
 - command [21-22, 21-23, 21-25, 21-26, 22-14](#)
 - command example [21-22, 21-24, 21-25, 21-26](#)
- spanning-tree vlan cost
 - command [21-28](#)
- spanning-tree vlan forward-time
 - command [21-31](#)
 - command example [21-32](#)
- spanning-tree vlan hello-time
 - command [21-31](#)
 - command example [21-31](#)
- spanning-tree vlan max-age
 - command [21-32](#)
 - command example [21-32](#)
- spanning-tree vlan port-priority
 - command [21-26](#)
 - command example [21-27](#)
- spanning-tree vlan priority
 - command [21-30](#)
 - command example [21-30](#)
- speed
 - configuring interface [10-7](#)
- speed command [4-2, 10-7](#)
- SST [21-15](#)
 - interoperability [21-16](#)
- Standard-Compliant IEEE 802.1s MST [20-1](#)
- standby link [12-1](#)
- standby links [12-1](#)
- static route, configuring [3-12](#)
- statistics
 - 802.1X [46-15](#)
- Sticky ARP [37-36](#)
- sticky ARP [37-36](#)
- sticky MAC address [47-2](#)
- Sticky secure MAC addresses [47-8, 47-9](#)
- storm control
 - see traffic-storm control
- STP
 - configuring [21-21](#)
 - bridge priority [21-29](#)
 - enabling [21-22, 21-23](#)
 - forward-delay time [21-31](#)
 - hello time [21-31](#)
 - maximum aging time [21-32](#)
 - port cost [21-28](#)
 - port priority [21-26](#)
 - root bridge [21-24](#)
 - secondary root switch [21-25](#)
 - defaults [21-20](#)
 - EtherChannel [13-4](#)
 - understanding [21-1](#)
 - 802.1Q Trunks [21-12](#)
 - Blocking State [21-7](#)
 - BPDUs [21-4](#)
 - disabled state [21-11](#)
 - forwarding state [21-10](#)
 - learning state [21-9](#)
 - listening state [21-8](#)
 - overview [21-2](#)
 - port states [21-6](#)
 - protocol timers [21-5](#)
 - root bridge election [21-4](#)
 - topology [21-5](#)
- STP BackboneFast
 - and MST [21-16](#)
 - configuring [22-13](#)
 - figure

- adding a switch [22-7](#)
- spanning-tree backbonefast
 - command [22-13, 22-14](#)
 - command example [22-13, 22-14](#)
 - understanding [22-4](#)
- STP BPDU Guard
 - and MST [21-16](#)
 - configuring [22-11](#)
 - spanning-tree portfast bpdu-guard
 - command [22-11](#)
 - understanding [22-2](#)
- STP bridge ID [21-2](#)
- STP EtherChannel guard [22-6](#)
- STP loop guard
 - and MST [21-16](#)
 - configuring [22-15](#)
 - overview [22-6](#)
- STP PortFast
 - and MST [21-16](#)
 - BPDU filter
 - configuring [22-10](#)
 - BPDU filtering [22-2](#)
 - configuring [22-8](#)
 - spanning-tree portfast
 - command [22-8, 22-9](#)
 - command example [22-8](#)
 - understanding [22-2](#)
- STP Portfast BPDU filtering
 - and MST [21-16](#)
- STP root guard [22-6, 22-14](#)
 - and MST [21-16](#)
- STP UplinkFast
 - and MST [21-16](#)
 - configuring [22-12](#)
 - spanning-tree uplinkfast
 - command [22-12](#)
 - command example [22-12](#)
 - understanding [22-3](#)
- strict-priority queue
 - see QoS strict priority
- supervisor engine
 - configuring [3-1](#)
 - default configuration [3-1](#)
 - default gateways [3-11](#)
 - environmental monitoring [53-10](#)
 - redundancy [7-1, 8-1, 9-1](#)
 - ROM monitor [3-20](#)
 - startup configuration [3-20](#)
 - static routes [3-12](#)
 - synchronizing configurations [7-20, 8-8, 9-8](#)
- Supervisor Engine 32 [5-1](#)
- supervisor engine redundancy
 - configuring [7-10, 8-5, 9-6](#)
- supervisor engines
 - displaying redundancy configuration [8-8, 9-9](#)
- Switched Port Analyzer
 - See SPAN
- switch fabric functionality [4-2](#)
 - configuring [4-4](#)
 - monitoring [4-4](#)
- switch fabric module [6-1](#)
 - configuring [6-3](#)
 - monitoring [6-5](#)
 - slot locations [6-2](#)
- switchport
 - configuring [11-14](#)
 - example [11-13](#)
 - show interfaces [10-12, 10-13, 11-7, 11-12](#)
- switchport access vlan [11-10, 11-14](#)
 - example [11-14](#)
- switchport mode access [11-4, 11-14](#)
 - example [11-14](#)
- switchport mode dynamic [11-9](#)
- switchport mode dynamic auto [11-4](#)
- switchport mode dynamic desirable [11-4](#)
 - default [11-5](#)
 - example [11-13](#)
- switchport mode trunk [11-4, 11-9](#)

- switchport nonegotiate [11-4](#)
- switchport trunk allowed vlan [11-11](#)
- switchport trunk encapsulation [11-8](#)
- switchport trunk encapsulation dot1q [11-3](#)
 - example [11-13](#)
- switchport trunk encapsulation isl [11-3](#)
- switchport trunk encapsulation negotiate [11-3](#)
 - default [11-5](#)
- switchport trunk native vlan [11-10](#)
- switchport trunk pruning vlan [11-11](#)
- switch priority
 - MSTP [20-23](#)
- switch TopN reports
 - foreground execution [56-2](#)
 - overview [56-1](#)
 - running [56-2](#)
 - viewing [56-2](#)
- system
 - configuration register
 - configuration [3-21 to 3-24](#)
 - settings at startup [3-22](#)
 - configuring global parameters [3-3 to 3-8](#)
- System Hardware Capacity [53-5](#)
- system image
 - determining if and how to load [3-22](#)
 - loading from Flash [3-24](#)

T

- TACACS+ [34-1, 35-1, 37-1, 45-1](#)
- TCP Intercept [34-2](#)
- TDR
 - checking cable connectivity [10-19](#)
 - enabling and disabling test [10-19](#)
 - guidelines [10-19](#)
- Telnet
 - accessing CLI [2-2](#)
- Time Domain Reflectometer
 - See TDR
- TopN reports
 - See switch TopN reports
- traceroute, Layer 2
 - and ARP [57-2](#)
 - and CDP [57-1](#)
 - described [57-1](#)
 - IP addresses and subnets [57-2](#)
 - MAC addresses and VLANs [57-2](#)
 - multicast traffic [57-2](#)
 - multiple devices on a port [57-2](#)
 - unicast traffic [57-1](#)
 - usage guidelines [57-1](#)
- traffic flood blocking [41-1](#)
- traffic-storm control
 - command
 - broadcast [40-3](#)
 - described [40-1](#)
 - monitoring [40-5](#)
 - thresholds [40-1](#)
- traffic suppression
 - see traffic-storm control
- translational bridge numbers (defaults) [15-6](#)
- transmit queues
 - see QoS transmit queues
- trunks [11-2](#)
 - 802.1Q Restrictions [11-5](#)
 - allowed VLANs [11-11](#)
 - configuring [11-7](#)
 - default interface configuration [11-7](#)
 - default VLAN [11-10](#)
 - different VTP domains [11-3](#)
 - encapsulation [11-3](#)
 - native VLAN [11-10](#)
 - private VLANs [16-2](#)
 - to non-DTP device [11-4](#)
 - VLAN 1 minimization [11-11](#)
- trust-dscp
 - see QoS trust-dscp
- trust-ipprec

see QoS trust-ipprec
 tunneling [43-4, 43-30](#)
 tunneling, 802.1Q
 See 802.1Q [18-1](#)

U

UDE [24-1](#)
 configuration [24-3](#)
 overview [24-2](#)
 UDE and UDLR [24-1](#)
 UDLD
 default configuration [49-3](#)
 enabling
 globally [49-3](#)
 on ports [49-4](#)
 overview [49-1](#)
 UDLR [24-1](#)
 back channel [24-1](#)
 configuration [24-6](#)
 tunnel
 (example) [24-7](#)
 ARP and NHRP [24-3](#)
 UDLR (unidirectional link routing)
 See UDLR
 unauthorized ports with 802.1X [46-4](#)
 Unicast and Multicast Flood Blocking [41-1](#)
 unicast flood blocking [41-1](#)
 unicast RPF [34-2](#)
 unicast storms
 see traffic-storm control
 Unidirectional Ethernet
 see UDE
 unidirectional ethernet
 example of setting [24-5](#)
 UniDirectional Link Detection Protocol
 see UDLD
 uniform mode
 configuring [43-38](#)

untrusted
 see QoS trust-cos
 see QoS untrusted
 upgrade guidelines [25-15](#)
 UplinkFast
 See STP UplinkFast
 URD [29-9](#)
 User-Based Rate Limiting [42-19, 42-61](#)
 user EXEC mode [2-5](#)

V

VACLs [36-2](#)
 configuring [36-4](#)
 examples [36-9](#)
 Layer 3 VLAN interfaces [36-8](#)
 Layer 4 port operations [35-5](#)
 logging
 configuration example [36-11](#)
 configuring [36-11](#)
 restrictions [36-11](#)
 MAC address based [36-5](#)
 multicast packets [36-4](#)
 overview [36-2](#)
 SVIs [36-8](#)
 WAN interfaces [36-2](#)
 version 8 (NDE) [50-2](#)
 virtual LAN
 See VLANs
 vlan
 command [15-10, 15-12, 50-11, 51-15](#)
 command example [15-11](#)
 VLAN-based QoS filtering [42-51](#)
 VLAN-bridge spanning-tree protocol [23-2](#)
 vlan database
 command [15-10, 15-12, 50-11, 51-15](#)
 example [15-11](#)
 vlan mapping dot1q
 command [15-15, 15-16](#)

- command example [15-17](#)
 - VLAN mode [25-16](#)
 - VLANs
 - allowed on trunk [11-11](#)
 - configuration guidelines [15-8](#)
 - configuration options
 - global configuration mode [15-9](#)
 - VLAN database mode [15-9](#)
 - configuring [15-1](#)
 - configuring (tasks) [15-9](#)
 - defaults [15-6](#)
 - extended range [15-2](#)
 - ID (default) [15-6](#)
 - interface assignment [15-12](#)
 - name (default) [15-6](#)
 - normal range [15-2](#)
 - private
 - See private VLANs
 - reserved range [15-2](#)
 - support for 4,096 VLANs [15-2](#)
 - token ring [15-3](#)
 - trunks
 - understanding [11-2](#)
 - understanding [15-1](#)
 - VLAN 1 minimization [11-11](#)
 - VTP domain [15-3](#)
 - VLAN translation
 - command example [15-15, 15-16](#)
 - VLAN Trunking Protocol
 - See VTP
 - voice VLAN
 - Cisco 7960 phone, port connections [17-1](#)
 - configuration guidelines [17-4](#)
 - configuring IP phone for data traffic
 - override CoS of incoming frame [17-7, 17-8](#)
 - configuring ports for voice traffic in
 - 802.1Q frames [17-5](#)
 - connecting to an IP phone [17-5](#)
 - default configuration [17-4](#)
 - overview [17-1](#)
 - VPN
 - configuration example [25-12](#)
 - guidelines and restrictions [25-11](#)
 - VTP
 - advertisements [14-3](#)
 - client, configuring [14-8](#)
 - configuration guidelines [14-5](#)
 - default configuration [14-5](#)
 - disabling [14-8](#)
 - domains [14-2](#)
 - VLANs [15-3](#)
 - modes
 - client [14-2](#)
 - server [14-2](#)
 - transparent [14-2](#)
 - monitoring [14-10](#)
 - overview [14-1](#)
 - pruning
 - configuration [11-11](#)
 - configuring [14-7](#)
 - overview [14-3](#)
 - server, configuring [14-8](#)
 - statistics [14-10](#)
 - transparent mode, configuring [14-8](#)
 - version 2
 - enabling [14-7](#)
 - overview [14-3](#)
-
- ## W
- WCCP
 - configuring on a router [55-2, 55-15](#)
 - service groups [55-8](#)
 - specifying protocol version [55-8](#)
 - web browser interface [1-1](#)
 - Web Cache Communication Protocol
 - See WCCP [55-1](#)
 - web caches

See cache engines

web cache services

- description [55-6](#)

web caching

- See web cache services
- See also WCCP

web scaling [55-2](#)

weighted round robin [42-86](#)

WRR [42-86](#)

X

xconnect command [25-15](#)