Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide

Release 12.2(40)SG

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

Customer Order Number: DOC-OL14303=
Text Part Number: OL-14303-01

Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide

CONTENTS

CHAPTER 13

CHAPTER 29

CHAPTER 30

Preface

This preface describes who should read this document, how it is organized, and its conventions. The preface also tells you how to obtain Cisco documents, as well as how to obtain technical assistance.

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining Catalyst 4500 series switches.

Organization

This guide is organized into the following chapters:

Chapter	Title	Description
Chapter 1	Product Overview	Presents an overview of the Cisco IOS software for the Catalyst 4500 series switches
Chapter 2	Command-Line Interfaces	Describes how to use the CLI
Chapter 3	Configuring the Switch for the First Time	Describes how to perform a baseline configuration of the switch
Chapter 4	Administering the Switch	Describes how to administer the switch.
Chapter 5	Configuring the Cisco IOS In Service Software Upgrade Process	Describes how to configure ISSU on the switch.
Chapter 6	Configuring Interfaces	Describes how to configure non-layer-specific features on Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces
Chapter 7	Checking Port Status and Connectivity	Describes how to check module and interface status
Chapter 8	Configuring Supervisor Engine Redundancy Using RPR and SSO	Describes how to configure RPR and SSO on the Catalyst 4507R and 4510R switches
Chapter 9	Configuring Cisco NSF with SSO Supervisor Engine Redundancy	Describes how to configure supervisor engine redundancy using Cisco nonstop forwarding (NSF) with stateful switchover (SSO).

Chapter	Title	Description
Chapter 10	Environmental Monitoring and Power Management	Describes how to configure power management and environmental monitoring features
Chapter 11	Configuring Power over Ethernet	Describes how to configure Power over Ethernet (PoE)
Chapter 12	Configuring the Catalyst 4500 Series Switch with Cisco Network Assistant	Describes how to install and configure Network Assistant and Embedded CiscoView
Chapter 13	Configuring VLANs, VTP, and VMPS	Describes how to configure VLANs, VTP, and VMPS.
Chapter 14	Configuring IP Unnumbered Interface	Describes how to configure IP Unnumbered support.
Chapter 15	Configuring Layer 2 Ethernet Interfaces	Describes how to configure interfaces to support Layer 2 features, including VLAN trunks
Chapter 16	Configuring SmartPort Macros	Describes how to configure SmartPort macros
Chapter 17	Configuring STP and MST	Describes how to configure the Spanning Tree Protocol (STP) and the Multiple Spanning Tree (MST) protocol and explains how they work.
Chapter 18	Configuring Optional STP Features	Describes how to configure the spanning-tree PortFast, UplinkFast, BackboneFast, and other STP features
Chapter 19	Configuring EtherChannel	Describes how to configure Layer 2 and Layer 3 EtherChannel port bundles
Chapter 20	Configuring IGMP Snooping and Filtering	Describes how to configure Internet Group Management Protocol (IGMP) snooping
Chapter 21	Configuring IPv6 MLD Snooping	Describes how to configure IPv6 MLD Snooping.
Chapter 22	Configuring 802.1Q and Layer 2 Protocol Tunneling	Describes how to configure 802.1Q and Layer 2 protocol Tunneling
Chapter 23	Configuring CDP	Describes how to configure the Cisco Discovery Protocol (CDP)
Chapter 24	Configuring UDLD	Describes how to configure the UniDirectional Link Detection (UDLD) protocol
Chapter 25	Configuring Unidirectional Ethernet	Describes how to configure unidirectional Ethernet
Chapter 26	Configuring Layer 3 Interfaces	Describes how to configure interfaces to support Layer 3 features
Chapter 27	Configuring Cisco Express Forwarding	Describes how to configure Cisco Express Forwarding (CEF) for IP unicast traffic
Chapter 28	Configuring Unicast Reverse Path Forwarding	Describes how to configure Unicast Reverse Path Forwarding.
Chapter 29	Configuring IP Multicast	Describes how to configure IP Multicast Multilayer Switching (MMLS)
Chapter 30	Configuring Policy-Based Routing	Describes how to configure policy-based routing

Chapter	Title	Description
Chapter 31	Configuring VRF-lite	Describes how to configure multiple VPN routing/forwarding (multi-VRF) instances in customer edge (CE) devices
Chapter 32	Configuring Quality of Service	Describes how to configure quality of service (QoS).
Chapter 33	Configuring Voice Interfaces	Describes how to configure multi-VLAN access ports for use with Cisco IP phones
Chapter 34	Configuring 802.1X Port-Based Authentication	Describes how to configure 802.1X port-based authentication
Chapter 35	Configuring Port Security	Describes how to configure port security and trunk port security.
Chapter 36	Configuring Control Plane Policing	Describes how to protect your Catalyst 4500 series switch using control plane policing (CoPP).
Chapter 37	Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts	Describes how to configure DHCP snooping and IP Source Guard
Chapter 38	Configuring Dynamic ARP Inspection	Describes how to configure Dynamic ARP Inspection
Chapter 39	Configuring Network Security with ACLs	Describes how to configure ACLS, VACLs, and MACLs
Chapter 40	Configuring Private VLANs	Describes how to set up and modify private VLANs
Chapter 41	Port Unicast and Multicast Flood Blocking	Describes how to configure unicast flood blocking
Chapter 42	Configuring Storm Control	Describes how to configure storm control suppression
Chapter 43	Configuring SPAN and RSPAN	Describes how to configure the Switched Port Analyzer (SPAN)
Chapter 44	Configuring System Message Logging	Describes how to configure system message logging.
Chapter 45	Configuring SNMP	Describes how to configure the Simple Network Management Protocol (SNMP).
Chapter 46	Configuring NetFlow	Describes how to configure NetFlow statistics gathering
Chapter 47	Configuring RMON	Describes how to configure Remote Network Monitoring (RMON).
Chapter 48	Performing Diagnostics	Describes vaious types of diagnostics on the Catalyst 4500 series switch.
Chapter 49	Configuring WCCP Version 2 Services	Describes how to configure the Catalyst 4500 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).
Chapter 50	Configuring MIB Support	Describes how to configure configure SNMP and MIB support.

Chapter	Title	Description
Chapter 51	ROM Monitor	Describes the ROM Monitor.
Appendix A	Acronyms and Abbreviations	Defines acronyms and abbreviations used in this book

Related Documentation

The following publications are available for the Catalyst 4500 series switches:

Catalyst 4500 Series Switch Documentation Home

- http://www.cisco.com/en/US/products/hw/switches/ps4324/tsd_products_support_series_home.html

Catalyst 4500 Series Switches Installation Guide (DOC-7814409=)

- http://www.cisco.com/en/US/products/hw/switches/ps4324/products_installation_guide_book09186a0080126d3d.html

Catalyst 4500 Series Module Installation Guide (DOC-786444=)

- http://www.cisco.com/en/US/products/hw/switches/ps4324/products_module_installation_guide_book09186a008009c17d.html

Catalyst 4500 Series Regulatory Compliance and Safety Information (DOC-7813233=)

- http://www.cisco.com/en/US/products/hw/switches/ps4324/products_regulatory_approvals_and_compliance09186a00800d7676.html

Installation notes for specific supervisor engines or for accessory hardware are available at:

- http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_installation_guides_list.html

Cisco IOS configuration guides and command references—Use these publications to help you configure Cisco IOS software features not described in the preceding publications:

- *Configuration Fundamentals Configuration Guide*
- *Configuration Fundamentals 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 Fax Applications Configuration Guide*
- *Voice, Video, and Fax Applications Command Reference*
- *Cisco IOS IP Configuration Guide*
- Cisco IOS IP Command Reference

The Cisco IOS configuration guides and command references are at http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/index.htm

For information about MIBs, refer to http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Software Documentation

The abilities of your switch and the modules supported depend greatly on the software you have installed. Each software release typically has each of the following:

- Release Note
  http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_release_notes_list.html
- Configuration Guide
  http://www.cisco.com/en/US/products/hw/switches/ps4324/products_installation_and_configuration_guides_list.html
- Command Reference
  http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_command_reference_list.html
- System Message Guide
  http://www.cisco.com/en/US/products/hw/switches/ps4324/products_system_message_guides_list.html

You may want to bookmark the guides appropriate to your software release.

- For information about MIBs, go to the following URL:

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

Conventions

This document uses the following typographical conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in **boldface**.
italic font	Command arguments for which you supply values are in *italics*.
[ ]	Command elements in square brackets are optional.
{ x | y | z }	Alternative keywords in command lines 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 because the string will include the quotation marks.
`screen` font	System displays are in `screen` font.
`boldface screen` font	Information you must enter verbatim is in **`boldface screen`** font.
`italic screen` font	Arguments for which you supply values are in *`italic screen`* font.

Convention	Description
⟶	This pointer highlights an important line of text in an example.
^	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.

Commands in Task Tables

Commands listed in task tables show only the relevant information for completing the task and not all available options for the command. For a complete description of a command, refer to the command in the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

Notices

The following notices pertain to this software license.

OpenSSL/Open SSL Project

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

License Issues

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.
4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

   “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS”' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young’s, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Obtaining Documentation, Obtaining Support, and Security Guidelines

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

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

CHAPTER 1

Product Overview

This chapter provides an overview of Catalyst 4500 series switches and includes the following major sections:

- Layer 2 Software Features, page 1-1
- Layer 3 Software Features, page 1-6
- Management Features, page 1-12
- Security Features, page 1-15

Note For more information about the chassis, modules, and software features supported by the Catalyst 4500 series switch, refer to the *Release Notes for the Catalyst 4500 Series Switch* at this location:

http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_release_note09186a0080758ff3.html

Layer 2 Software Features

The following subsections describe the key Layer 2 switching software features on the Catalyst 4500 series switch:

- 802.1Q and Layer 2 Protocol Tunneling, page 1-2
- CDP, page 1-2
- EtherChannel Bundles, page 1-2
- Jumbo Frames, page 1-2
- MST, page 1-3
- PVRST+, page 1-3
- QoS, page 1-3
- Spanning Tree Protocol, page 1-4
- SSO, page 1-4
- SVI Autostate, page 1-5

802.1Q and Layer 2 Protocol Tunneling

802.1Q tunneling is a Q-in-Q technique that expands the VLAN space by retagging the tagged packets that enter the service provider infrastructure. 802.1Q tunneling allows service providers to assign a VLAN to each customer without losing the original customer VLAN IDs inside the tunnel. All data traffic that enters the tunnel is encapsulated with the tunnel VLAN ID. Layer 2 Protocol Tunneling is a similar technique for all Layer 2 control traffic. 802.1Q tunneling and Layer 2 Protocol Tunneling are supported on Supervisor Engine V only.

For information on configuring 802.1Q tunneling, see Chapter 22, “Configuring 802.1Q and Layer 2 Protocol Tunneling.”

CDP

The Cisco Discovery Protocol (CDP) is a device-discovery protocol that is both media- and protocol-independent. CDP is available on all Cisco products, including routers, switches, bridges, and access servers. Using CDP, a device can advertise its existence to other devices and receive information about other devices on the same LAN. CDP enables Cisco switches and routers to exchange information, such as their MAC addresses, IP addresses, and outgoing interfaces. CDP runs over the data-link layer only, allowing two systems that support different network-layer protocols to learn about each other. Each device configured for CDP sends periodic messages to a multicast address. Each device advertises at least one address at which it can receive Simple Network Management Protocol (SNMP) messages.

For information on configuring CDP, see Chapter 23, “Configuring CDP.”

EtherChannel Bundles

EtherChannel port bundles allow you to create high-bandwidth connections between two switches by grouping multiple ports into a single logical transmission path.

For information on configuring EtherChannel, see Chapter 19, “Configuring EtherChannel.”

Jumbo Frames

The jumbo frames feature allows the switch to forward packets as large as 9216 bytes (larger than the IEEE Ethernet MTU), rather than declare those frames “oversize” and discard them. This feature is typically used for large data transfers. The jumbo feature can be configured on on WS-X4306-GB: all ports, WS-X4232-GB-RJ: ports 1-2, WS-X4418-GB: ports 1-2, WS-X4412-2GB-TX: ports 13-14, 4648-GB-RJ45V, WS-X4648-GB+RJ45V, WS-X4706-10GE linecards, and on.Supervisor uplink ports.

For information on Jumbo Frames, see Chapter 6, “Configuring Interfaces.”

MST

IEEE 802.1s Multiple Spanning Tree (MST) allows for multiple spanning tree instances within a single 802.1Q or Inter-Switch Link (ISL) VLAN trunk. MST extends the IEEE 802.1w Rapid Spanning Tree (RST) algorithm to multiple spanning trees. This extension provides both rapid convergence and load balancing within a VLAN environment.

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).

For information on configuring MST, see Chapter 17, “Configuring STP and MST.”

PVRST+

Per-VLAN Rapid Spanning Tree (PVRST+) is the implementation of 802.1w on a per-VLAN basis. It is the same as PVST+ with respect to STP mode and runs RSTP protocol based on 802.1w.

For information on configuring PVRST+, see Chapter 17, “Configuring STP and MST.”

QoS

The quality of service (QoS) feature prevents congestion by selecting network traffic and prioritizing it according to its relative importance. Implementing QoS in your network makes network performance more predictable and bandwidth use more effective.

The Catalyst 4500 series switch supports the following QoS features:

- Classification and marking
- Ingress and egress policing, including per-Port per-VLAN policing
- Sharing and shaping

Catalyst 4500 series switch supports trusted boundary, which uses the Cisco Discovery Protocol (CDP) to detect the presence of a Cisco IP phone (such as the Cisco IP Phone 7910, 7935, 7940, and 7960) on a switch port. If the telephone is not detected, the trusted boundary feature disables the trusted setting on the switch port and prevents misuse of a high-priority queue.

The Catalyst 4500 series switch also supports QoS Automation (AutoQoS), which simplifies the deployment of existing QoS features through automatic configuration.

Cisco Modular QoS Command-Line-Interface (Supervisor Engine 6-E)

Cisco Modular QoS CLI (MQC) is the framework used to implement Cisco IOS Software QoS. MQC allows the user to define a traffic class, create a traffic policy (containing the QoS feature to be applied to the traffic class), and attach the traffic policy to an interface. MQC is a cross-Cisco baseline that provides a consistent syntax and behavior of QoS features across multiple product families. Cisco IOS Software Release 12.2(40)SG complies to MQC for configuration of QoS features on the Supervisor Engine 6-E. MQC enables rapid deployment of new features and technology innovations and facilitates the management of network performance with respect to bandwidth, delay, jitter, and packet loss, enhancing the performance of mission-critical business applications. The rich and advanced QoS features that are supported as part of the Supervisor Engine 6-E are enabled using Cisco MQC.

Two-Rate Three-Color Policing (Supervisor Engine 6-E)

The Two-Rate Three-Color Policing feature (also termed *Hierarchical QoS*) limits the input or output transmission rate of a class of traffic based on user-defined criteria and marks or colors packets by setting the applicable differentiated services code point (DSCP) values. This feature is often configured on the interfaces at the edge of a network to limit the rate of traffic entering or leaving the network. Using this feature, traffic that conforms to user-defined criteria can be sent through the interfaces, while traffic that exceeds or violates these criteria is sent out with a decreased priority setting or even dropped.

For information on QoS and Auto QoS, see Chapter 32, "Configuring Quality of Service."

Spanning Tree Protocol

The Spanning Tree Protocol (STP) allows you to create fault-tolerant internetworks that ensure an active, loop-free data path between all nodes in the network. STP uses an algorithm to calculate the best loop-free path throughout a switched network.

For information on configuring STP, see Chapter 17, "Configuring STP and MST."

The Catalyst 4500 series switch supports the following STP enhancements:

- Spanning tree PortFast—PortFast allows a port with a directly attached host to transition to the forwarding state directly, bypassing the listening and learning states.
- Spanning tree UplinkFast—UplinkFast provides fast convergence after a spanning-tree topology change and achieves load balancing between redundant links using uplink groups. Uplink groups provide an alternate path in case the currently forwarding link fails. UplinkFast is designed to decrease spanning-tree convergence time for switches that experience a direct link failure.
- Spanning tree BackboneFast—BackboneFast reduces the time needed for the spanning tree to converge after a topology change caused by an indirect link failure. BackboneFast decreases spanning-tree convergence time for any switch that experiences an indirect link failure.
- Spanning tree root guard—Root guard forces a port to become a designated port so that no switch on the other end of the link can become a root switch.

For information on the STP enhancements, see Chapter 18, "Configuring Optional STP Features."

SSO

Stateful switchover (SSO) enables you to propagate configuration and state information from the active to the redundant supervisor engine so that sub-second interruptions in Layer 2 traffic occur when the active supervisor engine switches over to the redundant supervisor engine.

- Stateful IGMP Snooping

  This feature propagates the IGMP data learned by the active supervisor engine to the redundant supervisor engine so that when a switchover occurs, the newly active supervisor engine is aware of the multicast group membership, which alleviates a disruption to multicast traffic during a switchover.

- Stateful DHCP Snooping

  This feature propagates the DHCP-snooped data from the active supervisor engine to the redundant supervisor engine so that when a switchover occurs, the newly active supervisor engine is aware of the DHCP data that was already snooped, and the security benefits continue uninterrupted.

SVI Autostate

When a SVI has multiple ports on a VLAN, normally the SVI will go down when all the ports in the VLAN go down. You might design your network so that some ports should not be counted in the calculation of SVI “going up or down.” SVI Autostate provides a knob to mark a port so that it is not counted in the SVI “going up and down” calculation and applies to all VLANs that are enabled on that port.

UBRL

User Based Rate Limiting (UBRL) enables you to adopt microflow policing to dynamically learn traffic flows and rate limit each unique flow to an individual rate. UBRL is available only on the Supervisor Engine V-10GE with the built-in NetFlow support.

UDLD

The UniDirectional Link Detection (UDLD) protocol allows devices connected through fiber-optic or copper Ethernet cables to monitor the physical configuration of the cables and detect a unidirectional link.

For information about UDLD, see Chapter 24, “Configuring UDLD.”

Unidirectional Ethernet

Unidirectional Ethernet uses only one strand of fiber for either transmitting or receiving one-way traffic for the Gigaport, instead of two strands of fiber for a full-duplex Gigaport Ethernet.

For information about Unidirectional Ethernet, see Chapter 25, “Configuring Unidirectional Ethernet.”

VLANs

A VLAN configures switches and routers according to logical, rather than physical, topologies. Using VLANs, a network administrator can combine any collection of LAN segments within an internetwork into an autonomous user group, such that the segments appear as a single LAN in the network. VLANs logically segment the network into different broadcast domains so that packets are switched only between ports within the VLAN. Typically, a VLAN corresponds to a particular subnet, although not necessarily.

For more information about VLANs, VTP, and Dynamic VLAN Membership, see Chapter 13, “Configuring VLANs, VTP, and VMPS.”

The following VLAN-related features are also supported.

- VLAN Trunking Protocol (VTP)—VTP maintains VLAN naming consistency and connectivity between all devices in the VTP management domain. You can have redundancy in a domain by using multiple VTP servers, through which you can maintain and modify the global VLAN information. Only a few VTP servers are required in a large network.
- Private VLANs—Private VLANs are sets of ports that have the features of normal VLANs and also provide some Layer 2 isolation from other ports on the switch.

  For information about private VLANs, see Chapter 40, “Configuring Private VLANs.”

- Private VLAN Trunk Ports—Private VLAN trunk ports allow a secondary port on a private VLAN to carry multiple secondary VLANs.
- Private VLAN Promiscuous Trunk Ports—Private VLAN promiscuous trunk extends the promiscuous port to a 802.1Q trunk port, carrying multiple primary VLANs (hence multiple subnets). Private VLAN promiscuous trunk is typically used to offer different services or content on different primary VLANs to isolated subscribers. Secondary VLANs can not be carried over the private VLAN promiscuous trunk.
- Dynamic VLAN Membership—Dynamic VLAN Membership allows you to assign switch ports to VLANs dynamically, based on the source Media Access Control (MAC) address of the device connected to the port. When you move a host from a port on one switch in the network to a port on another switch in the network, that switch dynamically assigns the new port to the proper VLAN for that host. With the VMPS Client feature, you can convert a dynamic access port to a VMPS client. VMPS clients can use VQP queries to communicate with the VMPS server to obtain a VLAN assignment for the port based on the MAC address of the host attached to that port.

Layer 3 Software Features

A Layer 3 switch is a high-performance switch that has been optimized for a campus LAN or an intranet, and it provides both wirespeed Ethernet routing and switching services. Layer 3 switching improves network performance with two software functions—route processing and intelligent network services.

Compared to conventional software-based switches, Layer 3 switches process more packets faster; they do so by using application-specific integrated circuit (ASIC) hardware instead of microprocessor-based engines.

The following subsections describe the key Layer 3 switching software features on the Catalyst 4500 series switch:

- CEF, page 1-6
- HSRP, page 1-7
- IP Routing Protocols, page 1-7
- Multicast Services, page 1-10
- NSF with SSO, page 1-11
- Policy-Based Routing, page 1-11
- Unidirectional Link Routing, page 1-11
- VRF-lite, page 1-12

CEF

Cisco Express Forwarding (CEF) is an advanced Layer 3 IP-switching technology. CEF optimizes network performance and scalability in networks with large and dynamic traffic patterns, such as the Internet, and on networks that use intensive web-based applications or interactive sessions. Although you can use CEF in any part of a network, it is designed for high-performance, highly resilient Layer 3 IP-backbone switching.

For information on configuring CEF, see Chapter 27, “Configuring Cisco Express Forwarding.”

HSRP

The Hot Standby Router Protocol (HSRP) provides high network availability by routing IP traffic from hosts on Ethernet networks without relying on the availability of any single Layer 3 switch. This feature is particularly useful for hosts that do not support a router discovery protocol and do not have the functionality to switch to a new router when their selected router reloads or loses power.

For information on configuring HSRP, refer to the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008042fbb3.html

SSO Aware HSRP

SSO Aware HSRP offers continuous data packet forwarding during a supervisor engine switchover without a path change to the standby HSRP router. During supervisor engine switchover, NSF with SSO continues forwarding data packets along known routes using the HSRP virtual IP address. When both supervisor engines fail on the active HSRP router, the standby HSRP router takes over as the active HSRP router. It further extends reliability and availability offered by the Catalyst 4500's NSF with SSO to the Layer 3 aggregation with redundant chassis. SSO aware HSRP is available for Supervisor Engine IV, V and V-10GE on Catalyst 4507R and 4510R chassis with supervisor redundancy.

IP Routing Protocols

The following routing protocols are supported on the Catalyst 4500 series switch:

- BGP, page 1-7
- EIGRP, page 1-8
- GLBP, page 1-8
- IGRP, page 1-8
- IS-IS, page 1-9
- OSPF, page 1-9
- RIP, page 1-9
- VRRP, page 1-10

BGP

The Border Gateway Protocol (BGP) is an exterior gateway protocol that allows you to set up an interdomain routing system to automatically guarantee the loop-free exchange of routing information between autonomous systems. In BGP, each route consists of a network number, a list of autonomous systems that information has passed through (called the autonomous system path), and a list of other path attributes.

The Catalyst 4500 series switch supports BGP version 4, including classless interdomain routing (CIDR). CIDR lets you reduce the size of your routing tables by creating aggregate routes, resulting in supernets. CIDR eliminates the concept of network classes within BGP and supports the advertising of IP prefixes. CIDR routes can be carried by OSPF, EIGRP, and RIP.

BGP Route-Map Continue

The BGP Route-Map Continue feature introduces the continue clause to the BGP route-map configuration. The continue clause provides more programmable policy configuration and route filtering. It introduces the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow configuring and organizing more modular policy definitions to reduce the number of policy configurations that are repeated within the same route map.

EIGRP

The Enhanced Interior Gateway Routing Protocol (EIGRP) is a version of IGRP that combines the advantages of link-state protocols with distance-vector protocols. EIGRP incorporates the Diffusing Update Algorithm (DUAL). EIGRP includes fast convergence, variable-length subnet masks, partially bounded updates, and multiple network-layer support. When a network topology change occurs, EIGRP checks its topology table for a suitable new route to the destination. If such a route exists in the table, EIGRP updates the routing table instantly. You can use the fast convergence and partial updates that EIGRP provides to route Internetwork Packet Exchange (IPX) packets.

EIGRP saves bandwidth by sending routing updates only when routing information changes. The updates contain information only about the link that changed, not the entire routing table. EIGRP also takes into consideration the available bandwidth when determining the rate at which it transmits updates.

Note Layer 3 switching does not support the Next Hop Resolution Protocol (NHRP).

Note Customers can configure Enhanced Interior Gateway Routing Protocol (EIGRP) to route IPv6 prefixes. EIGRP configuration and protocol behavior for both IPv4 and IPv6 prefixes are similar, providing operational familiarity and continuity. EIGRP support for IPv6 will enable customers to use their existing EIGRP knowledge and processes, allowing them to deploy an IPv6 network at a low cost.

GLBP

The Gateway Load Balancing Protocol (GLBP) feature provides automatic router backup for IP hosts configured with a single default gateway on a LAN. Multiple first hop routers on the LAN combine to offer a single virtual first hop IP router while sharing the IP packet forwarding load. GLBP devices share packet-forwarding responsibilities, optimizing resource usage, thereby reducing costs. Other routers on the LAN may act as redundant GLBP routers that will become active if any of the existing forwarding routers fail. This improves the resiliency of the network and reduces administrative burden. GLBP is a feature that is applicable for the Supervisor Engine 6-E and the classic supervisor engines.

For details on GLBP, refer to this URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008042fb97.html

IGRP

The Interior Gateway Routing Protocol (IGRP) is a robust distance-vector Interior Gateway Protocol (IGP) developed by Cisco to provide for routing within an autonomous system (AS). Distance vector routing protocols request that a switch send all or a portion of its routing table data in a routing update message at regular intervals to each of its neighboring routers. As routing information proliferates through the network, routers can calculate distances to all nodes within the internetwork. IGRP uses a combination of metrics: internetwork delay, bandwidth, reliability, and load are all factored into the routing decision.

IS-IS

The Intermediate System-to-Intermediate System Protocol (IS-IS Protocol) uses a link-state routing algorithm. It closely follows the Open Shortest Path First (OSPF) routing protocol used within the TCP/IP environment. The operation of ISO IS-IS Protocol requires each router to maintain a full topology map of the network (that is, which intermediate systems and end systems are connected to which other intermediate systems and end systems). Periodically, the router runs an algorithm over its map to calculate the shortest path to all possible destinations.

The IS-IS Protocol uses a two-level hierarchy. Intermediate Systems (or routers) are classified as Level 1 and Level 2. Level 1 intermediate systems deal with a single routing area. Traffic is relayed only within that area. Any other internetwork traffic is sent to the nearest Level 2 intermediate systems, which also acts as a Level 1 intermediate systems. Level 2 intermediate systems move traffic between different routing areas within the same domain.

An IS-IS with multi-area support allows multiple Level 1 areas within in a single intermediate system, thus allowing an intermediate system to be in multiple areas. A single Level 2 area is used as backbone for inter-area traffic.

Only Ethernet frames are supported. The IS-IS Protocol does not support IPX.

OSPF

The Open Shortest Path First (OSPF) protocol is a standards-based IP routing protocol designed to overcome the limitations of RIP. Because OSPF is a link-state routing protocol, it sends link-state advertisements (LSAs) to all other routers within the same hierarchical area. Information on the attached interfaces and their metrics is used in OSPF LSAs. As routers accumulate link-state information, they use the shortest path first (SPF) algorithm to calculate the shortest path to each node. Additional OSPF features include equal-cost multipath routing and routing based on the upper-layer type of service (ToS) requests.

OSPF employs the concept of an area, which is a group of contiguous OSPF networks and hosts. OSPF areas are logical subdivisions of OSPF autonomous systems in which the internal topology is hidden from routers outside the area. Areas allow an additional level of hierarchy different from that provided by IP network classes, and they can be used to aggregate routing information and mask the details of a network. These features make OSPF particularly scalable for large networks.

RIP

The Routing Information Protocol (RIP) is a distance-vector, intradomain routing protocol. RIP works well in small, homogeneous networks. In large, complex internetworks, it has many limitations, such as a maximum hop count of 15, lack of support for variable-length subnet masks (VLSMs), inefficient use of bandwidth, and slow convergence. RIP II does support VLSMs.

VRRP

Virtual Router Redundancy Protocol (VRRP) is a standard based first-hop redundancy protocol. With VRRP, a group of routers function as one virtual router by sharing one virtual IP address and one virtual MAC address. The master router performs packet forwarding, while the backup routers stay idle. VRRP is typically used in the multi vendor first-hop gateway redundancy deployment.

Multicast Services

Multicast services save bandwidth by forcing the network to replicate packets only when necessary and by allowing hosts to join and leave groups dynamically. The following multicast services are supported:

- Cisco Group Management Protocol (CGMP) server—CGMP server manages multicast traffic. Multicast traffic is forwarded only to ports with attached hosts that request the multicast traffic.
- Internet Group Management Protocol (IGMP) snooping—IGMP snooping manages multicast traffic. The switch software examines IP multicast packets and forwards packets based on their content. Multicast traffic is forwarded only to ports with attached hosts that request multicast traffic.

  Support for IGMPv3 provides constrained flooding of multicast traffic in the presence of IGMPv3 hosts or routers. IGMPv3 snooping listens to IGMPv3 query and membership report messages to maintain host-to-multicast group associations. It enables a switch to propagate multicast data only to ports that need it. IGMPv3 snooping is fully interoperable with IGMPv1 and IGMPv2.

  Explicit Host Tracking (EHT) is an extension to IGMPv3 snooping. EHT enables immediate leave operations on a per-port basis. EHT can be used to track per host membership information or to gather statistics about all IGMPv3 group members.

  For information on configuring IGMP snooping, see Chapter 20, “Configuring IGMP Snooping and Filtering.”
- IPv6 Multicast Listen Discovery (MLD) and Multicast Listen Discovery snooping—(MLD is a protocol used by IPv6 multicast devices to discover the presence of multicast listeners (nodes wishing to receive IPv6 multicast packets) on its directly attached links and to discover which multicast packets are of interest to neighboring nodes. MLD snooping is supported in two different versions: MLD v1 and MLD v2. Network switches use MLD snooping to limit the flood of multicast traffic, causing IPv6 multicast data to be selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This lessens the load on devices in the network, minimizing unnecessary bandwidth on links, enabling efficient distribution of IPv6 multicast data.

  For information on configuring multicast services, see Chapter 21, “Configuring IPv6 MLD Snooping”.
- Protocol Independent Multicast (PIM)—PIM is protocol-independent because it can leverage whichever unicast routing protocol is used to populate the unicast routing table, including EIGRP, OSPF, BGP, or static route. PIM also uses a unicast routing table to perform the Reverse Path Forwarding (RPF) check function instead of building a completely independent multicast routing table.

  For information on configuring multicast services, see Chapter 29, “Configuring IP Multicast.”

NSF with SSO

Non-Stop Forwarding with Stateful Switchover (NSF/SSO) offers continuous data packet forwarding in a Layer 3 routing environment during supervisor engine switchover. It further extends reliability and availability offered by the Catalyst 4500's SSO and NSF-aware to the Layer 3 networks. During supervisor engine switchover, NSF/SSO continues forwarding data packets along known routes while the routing protocol information is recovered and validated, avoiding unnecessary route flaps and network instability. With NSF/SSO, IP phone calls do not drop. NSF/SSO is supported for OSPF, BGP, EIGRP, IS-IS, and Cisco Express Forwarding (CEF). NSF/SSO is typically deployed in the most critical parts of an enterprise or service provider network, such as Layer 3 aggregation/core or a resilient Layer 3 wiring closet design. It is an essential component of single chassis deployment for critical applications. NSF/SSO is available for all shipping supervisor engines on Catalyst 4507R and 4510R chassis with supervisor redundancy.

For information on NSF with SSO, see Chapter 9, "Configuring Cisco NSF with SSO Supervisor Engine Redundancy."

ISSU

SSO requires the same version of IOS on both the active and standby supervisor engines. Because of version mismatch during an upgrade or downgrade of the Cisco IOS software, a Catalyst 4500 series switch is forced into operating in RPR mode. In this mode, after the switchover, you can observe link-flaps and a disruption in service. This issue is solved by the In Service Software Upgrade (ISSU) feature that enables you to operate in SSO/NSF mode while performing software upgrade or downgrade.

ISSU allows an upgrade or downgrade of the Catalyst IOS images at different release levels on the both the active and standby supervisor engines by utilizing the Version Transformation Framework between the stateful components running on each supervisor engine.

Policy-Based Routing

Traditional IP forwarding decisions are based purely on the destination IP address of the packet being forwarded. Policy Based Routing (PBR) enables forwarding based upon other information associated with a packet, such as the source interface, IP source address, Layer 4 ports, and so on. This feature allows network managers more flexibility in how they configure and design their networks.

For more information on policy-based routing, see Chapter 30, "Configuring Policy-Based Routing."

Unidirectional Link Routing

Unidirectional link routing (UDLR) provides a way to forward multicast packets over a physical unidirectional interface (such as a satellite link of high bandwidth) to stub networks that have a back channel.

For information on configuring unidirectional link routing, refer to the chapter "Configuring Unidirectional Link Routing" in the *Cisco IP and IP Routing Configuration Guide*.

VRF-lite

VPN routing and forwarding (VRF-lite) is an extension of IP routing that provides multiple routing instances. Along with BGP, it enables the creation of a Layer 3 VPN service by keeping separate IP routing and forwarding tables for each VPN customer. VRF-lite uses input interfaces to distinguish routes for different VPNs. It forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF, allowing the creation of multiple Layer 3 VPNs on a single switch. Interfaces in a VRF could be either physical, such as an Ethernet port, or logical, such as a VLAN switch virtual interface (SVI). However, interfaces cannot belong to more than one VRF at any time.

For information on VRF-lite, see Chapter 31, “Configuring VRF-lite.”

Management Features

The Catalyst 4500 series switch offers network management and control through the CLI or through alternative access methods, such as SNMP. The switch software supports these network management features:

- Cisco Network Assistant and Embedded CiscoView, page 1-12
- Dynamic Host Control Protocol, page 1-13
- FAT File Management System (Supervisor Engine 6-E only), page 1-13
- Forced 10/100 Autonegotiation, page 1-13
- Intelligent Power Management, page 1-13
- MAC Address Notification, page 1-13
- MAC Notify MIB, page 1-14
- NetFlow Statistics, page 1-14
- Secure Shell, page 1-14
- Simple Network Management Protocol, page 1-14
- SPAN and RSPAN, page 1-14
- Virtual Router Redundancy Protocol, page 1-15
- Web Content Coordination Protocol, page 1-15

Cisco Network Assistant and Embedded CiscoView

Web-based tools to configure the Catalyst 4500 series switch. Cisco Network Assistant manages standalone devices, clusters of devices, or federations of devices from anywhere in your intranet. Using its graphical user interface, you can perform multiple configuration tasks without having to remember command-line interface commands. Embedded CiscoView is a device management application that can be embedded on the switch flash and provides dynamic status, monitoring, and configuration information for your switch.

Visual port status information—The switch LEDs provide visual management of port- and switch-level status.

For more information on Cisco Network Assistant and Embedded CiscoView, see Chapter 12, “Configuring the Catalyst 4500 Series Switch with Cisco Network Assistant.”

Dynamic Host Control Protocol

The Catalyst 4500 series switch uses DHCP in the following ways:

- Dynamic Host Control Protocol server—The Cisco IOS DHCP server feature is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. If the Cisco IOS DHCP server cannot satisfy a DHCP request from its own database, it can forward the request to one or more secondary DHCP servers defined by the network administrator.
- Dynamic Host Control Protocol autoconfiguration—With this feature your switch (the DHCP client) is automatically configured at startup with IP address information and a configuration file.

For more information on configuring the DHCP server, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t1/easyip2.htm

FAT File Management System (Supervisor Engine 6-E only)

The FAT file system is widely used to manage files on devices disks and flash. The support of the FAT file system allows one to easily remove, add, and/or transfer images to and from the flash.

Forced 10/100 Autonegotiation

This feature allows you to configure a port to limit the speed at which it will autonegotiate to a speed lower than the physically maximum speed. This method of reducing the throughput incurs much less overhead than using an ACL.

Intelligent Power Management

Working with powered devices (PDs) from Cisco, this feature uses power negotiation to refine the power consumption of an 802.3af-compliant PD beyond the granularity of power consumption provided by the 802.3af class. Power negotiation also enables the backward compatibility of newer PDs with older modules that do not support either 802.3af or high-power levels as required by IEEE standard.

For more information on Intelligent Power Management, see the “Intelligent Power Management” section in Chapter 11, “Configuring Power over Ethernet.”

MAC Address Notification

MAC address notification monitors the MAC addresses that are learned by, aged out or removed from the Catalyst 4500 series switch. Notifications are sent out or retrieved via the CISCO-MAC-NOTIFICATION MIB. It is typically used by a central network management application to collect such MAC address notification events for host moves. User configurable MAC table utilization thresholds can be defined to notify any potential DoS or man-in-the-middle attack.

For information on MAC Address Notification, see Chapter 4, “Administering the Switch.”

MAC Notify MIB

The MAC Notify MIB feature monitors network performance, utilization, and security conditions enabling a network administrator to track the MAC addresses that are learned or removed on the switch forwarding the Ethernet frames.

NetFlow Statistics

NetFlow Statistics is a global traffic monitoring feature that allows flow-level monitoring of all IPv4-routed traffic through the switch. Both routed and switched IP flows are supported.

For more information on NetFlow statistics, see Chapter 46, “Configuring NetFlow.”

Secure Shell

Secure Shell (SSH) is a program that enables you to log into another computer over a network, to execute commands remotely, and to move files from one machine to another. The switch may not initiate SSH connections: SSH will be limited to providing a remote login session to the switch and will only function as a server.

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) facilitates the exchange of management information between network devices. The Catalyst 4500 series switch supports these SNMP types and enhancements:

- SNMP—A full Internet standard
- SNMP v2—Community-based administrative framework for version 2 of SNMP
- SNMP v3—Security framework with three levels: noAuthNoPriv, authNoPriv, and authPriv (available only on a crypto image, like cat4000-i5k91s-mz)
- SNMP trap message enhancements—Additional information with certain SNMP trap messages, including spanning-tree topology change notifications and configuration change notifications

For more information on SNMP, see Chapter 45, “Configuring SNMP”.

SPAN and RSPAN

Switched Port Analyzer (SPAN) allows you to monitor traffic on any port for analysis by a network analyzer or Remote Monitoring (RMON) probe. You also can do the following:

- Configure ACLs on SPAN sessions.
- Allow incoming traffic on SPAN destination ports to be switched normally.
- Explicitly configure the encapsulation type of packets that are spanned out of a destination port.
- Restrict ingress sniffing depending on whether the packet is unicast, multicast, or broadcast, and depending on whether the packet is valid.
- Mirror packets sent to or from the CPU out of a SPAN destination port for troubleshooting purposes.

For information on SPAN, see Chapter 43, “Configuring SPAN and RSPAN.”

Remote SPAN (RSPAN) is an extension of SPAN, where source ports and destination ports are distributed across multiple switches, allowing remote monitoring of multiple switches across the network. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session on all participating switches.

For information on RSPAN, see Chapter 43, "Configuring SPAN and RSPAN."

Virtual Router Redundancy Protocol

The Virtual Router Redundancy Protocol (VRRP) operates between routers attached to a common LAN and enables them to provide first-hop resiliency to LAN clients.

For information on VRRP, see the URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cg/hiap_c/ch20/haipvrrp.htm

Web Content Coordination Protocol

Web Content Communication Protocol (WCCP) Version 2 Layer 2 (L2) redirection enables Catalyst 4500 series switches to transparently redirect content requests to the directly connected content engines via a Layer 2/MAC address rewrite. The WCCPv2 L2 redirection is accelerated in the switching hardware, and is therefore more efficient than Layer 3 (L3) redirection using Generic Routing Encapsulation (GRE). The content engines in a cache cluster transparently store frequently accessed content and then fulfill successive requests for the same content, eliminating repetitive transmissions of identical content from the original content servers. It supports the transparent redirection of HTTP and non-HTTP traffic with ports or dynamic services, such as Web caching, HTTPS caching, File Transfer Protocol (FTP) caching, proxy caching, media caching, and streaming services. WCCPv2 L2 redirection is typically deployed for transparent caching at network edge, such as regional or branch sites. WCCPv2 L2 redirection can not be enabled on the same input interface with PBR or VRF-lite. ACL based classification for L2 redirection is not supported.

For information on WCCP, see Chapter 49, "Configuring WCCP Version 2 Services."

Security Features

The Catalyst 4500 series switch offers network management and control through the CLI or through alternative access methods, such as SNMP. The switch software supports these security features:

- 802.1X Identity-Based Network Security, page 1-16
- Dynamic ARP Inspection, page 1-17
- Dynamic Host Configuration Protocol Snooping, page 1-17
- Flood Blocking, page 1-17
- Hardware-Based Control Plane Policing, page 1-17
- IP Source Guard for Static Hosts, page 1-18
- IP Source Guard, page 1-18
- Local Authentication, RADIUS, and TACACS+ Authentication, page 1-18
- Network Admission Control (NAC), page 1-18

- Network Security with ACLs, page 1-19
- Port Security, page 1-19
- Storm Control, page 1-19
- uRPF Strict Mode (Supervisor Engine 6-E only), page 1-20
- Utilities, page 1-20

802.1X Identity-Based Network Security

This security feature consists of the following:

- 802.1X protocol—This feature provides a means for a host that is connected to a switch port to be authenticated before it is given access to the switch services.
- 802.1X with VLAN assignment—This feature enables you to enable non-802.1X-capable hosts to access networks that use 802.1X authentication.
- 802.1X RADIUS accounting—This feature enables you to track the usage of network devices.
- 802.1X authentication for Guest VLANs—This feature enables you to use VLAN assignment to limit network access for certain users.
- 802.1X with MAC Authentication Bypass—This feature provides network access to agentless devices without 802.1X supplicant capabilities, such as printers. Upon detecting a new MAC address on a switch port, the Catalyst 4500 series switch will proxy an 802.1X authentication request based on the device's MAC address.
- 802.1X with Inaccessible Authentication Bypass—This feature applies when the AAA servers are unreachable or non-responsive. In this situation, 802.1X user authentication typically fails with the port closed, and the user is denied access. Inaccessible Authentication Bypass provides a configurable alternative on the Catalyst 4500 series switch to grant a critical port network access in a locally-specified VLAN.
- 802.1X with Unidirectional Controlled Port—This feature allows the Wake-on-LAN (WoL) magic packets to reach a workstation attached to an unauthorized 802.1X switch port. Unidirectional Controlled Port is typically used to push out Operating Systems or software updates from a central server to workstations at night.
- 802.1X Authentication Failed Open Assignment—This feature enables you to configure a switch to handle the case when a device fails to authenticate itself correctly through 802.1X (for example, not providing the correct password).
- 802.1X with Voice VLAN—This feature enables you to use 802.1X security on a port while enabling it to be used by both Cisco IP phones and devices with 802.1X supplicant support.
- 802.1X Convergence—This feature provides consistency between the switching business units in 802.1X configuration and implementation.
- Multi-Domain Authentication—This feature requires phones without an 802.1x supplicant to be authenticated via MAC Auth Bypass.

For more information on 802.1X identity-based network security, see Chapter 34, "Configuring 802.1X Port-Based Authentication."

Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) intercepts all ARP requests, replies on untrusted ports, and verifies each intercepted packet for valid IP to MAC bindings. Dynamic ARP Inspection helps to prevent attacks on a network by not relaying invalid ARP replies out to other ports in the same VLAN. Denied ARP packets are logged by the switch for auditing.

For more information on dynamic ARP inspection, see Chapter 38, "Configuring Dynamic ARP Inspection."

Dynamic Host Configuration Protocol Snooping

Dynamic Host Configuration Protocol (DHCP) Snooping is a security feature that is a component of a DHCP server. DHCP snooping provides security by intercepting untrusted DHCP messages and by building and maintaining a DHCP snooping binding table. An untrusted message is a message that is received from outside the network or firewall that can cause traffic attacks within your network.

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. It also provides a way to differentiate between untrusted interfaces connected to the end-user and trusted interfaces connected to the DHCP server or another switch.

For DHCP server configuration information, refer to the chapter, "Configuring DHCP," in the *Cisco IOS IP and IP Routing Configuration Guide* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ip_c/ipcprt1/1cddhcp.htm

For information on configuring DHCP snooping, see Chapter 37, "Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts."

Flood Blocking

Flood blocking enables users to disable the flooding of unicast and multicast packets on a per-port basis. Occasionally, unknown unicast or multicast traffic from an unprotected port is flooded to a protected port because a MAC address has timed out or has not been learned by the switch.

For information on flood blocking, see Chapter 41, "Port Unicast and Multicast Flood Blocking."

Hardware-Based Control Plane Policing

Control Plane Policing provides a unified solution to limit the rate of CPU bound control plane traffic in hardware. It enables users to install system wide control plane ACLs to protect the CPU by limiting rates or filtering out malicious DoS attacks. Control plane policing ensures the network stability, availability and packet forwarding, and prevents network outages such as loss of protocol updates despite an attack or heavy load on the switch. Hardware based control plane policing is available for all shipping Catalyst 4500 supervisor engines. It supports various Layer 2 and Layer 3 control protocols, such as CDP, EAPOL, STP, DTP, VTP, ICMP, CGMP, IGMP, DHCP, RIPv2, OSPF, PIM, TELNET, SNMP, HTTP and packets destined to 224.0.0.* multicast link local addresses. Pre-defined system policies or user-configurable policies can be applied to those control protocols.

For information on control plane policing, see Chapter 36, "Configuring Control Plane Policing."

IP Source Guard for Static Hosts

This feature allows you to secure the IP address learned from static hosts via ARP packets and then bind that IP address to a given MAC address using the device tracking database, allowing entries to survive through link down events.

IP Source Guard (IPSG) for static hosts allows multiple bindings per port per mac address for both dhcp and static hosts i.e. in both device tracking database as well as dhcp snooping binding data base. Moreover, it enable you to take action when a limit is exceeded.

For information on configuring IPSG for static hosts, see Chapter 37, “Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts.”

IP Source Guard

Similar to DHCP snooping, this feature is enabled on an untrusted 12 port that is configured for DHCP snooping. Initially all IP traffic on the port is blocked except for the DHCP packets, which are captured by the DHCP snooping process. When a client receives a valid IP address from the DHCP server, a PVACL is installed on the port, which restricts the client IP traffic only to clients with assigned IP addresses, so any IP traffic with source IP addresses other than those assigned by the DHCP server will be filtered out. This filtering prevents a malicious host from attacking a network by hijacking neighbor host's IP address.

For information on configuring IP Source Guard, see Chapter 37, “Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts.”

Local Authentication, RADIUS, and TACACS+ Authentication

RADIUS and TACACS+ control access to the switch. For additional information, refer to the chapter “Authentication, Authorization, and Accounting (AAA),” in *Cisco IOS Security Configuration Guide*, Release 12.1, at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/secur_c/scprt1/index.htm

Network Admission Control (NAC)

NAC supports consists of two features:

- NAC Layer 2 IP Validation

  NAC L2 IP is an integral part of Cisco Network Admission Control. It offers the first line of defense for infected hosts (PCs and other devices attached to a LAN port) attempting to connect to the corporate network. NAC L2 IP on the Cisco Catalyst 4500 series switch performs posture validation at the Layer 2 edge of the network for non-802.1x-enabled host devices. Host device posture validation includes anti-virus state and OS patch levels. Depending on the corporate access policy and host device posture, a host may be unconditionally admitted, admitted with restricted access, or quarantined to prevent the spread of viruses across the network.

  For more information on Layer 2 IP validation, see the URL:

  http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_configuration_guide09186a00805764fd.html

- NAC Layer 2 802.1X Authentication

  The Cisco Catalyst 4500 series switch extends NAC support to 802.1x-enabled devices. Like NAC L2 IP, the NAC L2 802.1x feature determines the level of network access based on endpoint information.

  For more information on 802.1X identity-based network security, see Chapter 34, "Configuring 802.1X Port-Based Authentication."

Network Security with ACLs

An access control list (ACL) filters network traffic by controlling whether routed packets are forwarded or blocked at the router interfaces. The Catalyst 4500 series switch examines each packet to determine whether to forward or drop the packet based on the criteria you specified within the access lists.

MAC access control lists (MACLs) and VLAN access control lists (VACLs) are supported. VACLs are also known as VLAN maps in Cisco IOS.

The following security features are supported:

- MAC address filtering, which enables you to block unicast traffic for a MAC address on a VLAN interface.
- Port ACLs, which enable you to apply ACLs to Layer 2 interfaces on a switch for inbound traffic.

For information on ACLs, MACLs, VLAN maps, MAC address filtering, and Port ACLs, see Chapter 39, "Configuring Network Security with ACLs."

Port Security

Port Security restricts traffic on a port based upon the MAC address of the workstation that accesses the port. Trunk port security extends this feature to trunks, including private VLAN isolated trunks, on a per-VLAN basis.

Sticky port security extends port security by saving the dynamically learned MAC addresses in the running configuration to survive port link down and switch reset. It enables a network administrator to restrict the MAC addresses allowed or the maximum number of MAC addresses on each port.

Voice VLAN sticky port security further extends the sticky port security to the Voice-over-IP deployment. Voice VLAN sticky port security locks a port and blocks access from a station with a MAC address different from the IP phone and the workstation behind the IP phone.

For information on port security, see Chapter 35, "Configuring Port Security."

Storm Control

Broadcast suppression is used to prevent LANs from being disrupted by a broadcast storm on one or more switch ports. A LAN broadcast storm occurs when broadcast packets flood the LAN, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation or in the network configuration can cause a broadcast storm. Multicast and broadcast suppression measures how much broadcast traffic is passing through a port and compares the broadcast traffic with some configurable threshold value within a specific time interval. If the amount of broadcast traffic reaches the threshold during this interval, broadcast frames are dropped, and optionally the port is shut down.

Cisco IOS Software Release 12.2(40)SG allows suppression of broadcast and multicast traffic on a per-port basis. (Supervisor Engine 6-E only)

For information on configuring broadcast suppression, see Chapter 42, “Configuring Storm Control.”

uRPF Strict Mode (Supervisor Engine 6-E only)

The uRPF feature mitigates problems caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. uRPF deflects denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This helps to protect the network of the customer, the ISP, and the rest of the Internet. When using uRPF in strict mode, the packet must be received on the interface that the router would use to forward the return packet. uRPF strict mode is supported for both IPv4 and IPv6 prefixes. IPv6 Forwarding in Hardware IPv6 is the next-generation IP protocol, designed to solve numerous problems that the original

For information on configuring broadcast suppression, see Chapter 28, “Configuring Unicast Reverse Path Forwarding”

Utilities

Layer 2 Traceroute

Layer 2 Traceroute allows the switch to identify the physical path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses.

For information about Layer 2 Traceroute, see Chapter 7, “Checking Port Status and Connectivity.”

Time Domain Reflectometry

Time Domain Reflectometry (TDR) is a technology used for diagnosing the state and reliability of cables. TDR can detect open, shorted, or terminated cable states. The calculation of the distance to the failure point is also supported.

For information about TDR, see Chapter 7, “Checking Port Status and Connectivity.”

Debugging Features

The Catalyst 4500 series switch has several commands to help you debug your initial setup. These commands are included in the following groups:

- **platform**
- **debug platform**

For more information, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

CHAPTER 2

Command-Line Interfaces

This chapter describes the CLIs you use to configure the Catalyst 4500 series switch. This chapter includes the following major sections:

- Accessing the Switch CLI, page 2-1
- Performing Command-Line Processing, page 2-3
- Performing History Substitution, page 2-3
- Understanding Cisco IOS Command Modes, page 2-4
- Getting a List of Commands and Syntax, page 2-5
- ROMMON Command-Line Interface, page 2-7

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

The following CLI cmds are changed with Supervisor Engine 6-E relative to other supervisor engines:

- the **verify** and **squeeze** commands are not supported in FAT file system.
- the **rename** command is supported in FAT file system.

  For Supervisor Engine 6-E, the **rename** command has been added for bootflash and slot0. For all other supervisor engines, the **rename** command is is supported for nvram devices only.
- the **fsck** command is supported for slot0 device. It is not supported in the file systems on supervisor engines other than 6-E

Accessing the Switch CLI

The following sections describe how to access the switch CLI:

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

Accessing the CLI Using 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 the initial switch configuration over a connection to the EIA/TIA-232 console interface. Refer to the *Catalyst 4500 Series Switch Module Installation Guide* for console interface cable connection procedures.

To access the switch through the console interface, perform this task:

	Command	Purpose
Step 1	Switch> **enable**	From the user EXEC prompt (>), enter **enable** to change to enable mode (also known as privileged mode or privileged EXEC mode).
Step 2	Password: *password* Switch#	At the password prompt, enter the system password. The prompt (#) appears, indicating that you have accessed the CLI in enabled mode.
Step 3	Switch# **quit**	When you are finished executing the task command, exit the session.

After accessing the switch through the EIA/TIA-232 interface, you see this display:

```
Press Return for Console prompt

Switch> enable
Password:< >
Switch#
```

Accessing the CLI Through Telnet

Note Before you make a Telnet connection to the switch, you must set the IP address for the switch. See the “Configuring Physical Layer 3 Interfaces” section on page 26-11.

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

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

	Command	Purpose
Step 1	**telnet** {*hostname* | *ip_addr*}	From the remote host, enter the **telnet** command and the name or IP address of the switch you want to access.
Step 2	Password: *password* Switch#	At the prompt, enter the password for the CLI. If no password has been configured, press **Return**.

	Command	Purpose
Step 3		Enter the necessary commands to complete your desired tasks.
Step 4	`Switch# quit`	When finished, exit the Telnet session.

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

```
unix_host% telnet Switch_1
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.
User Access Verification
Password:< >
Switch_1> enable
Password:
Switch_1#
```

Performing Command-Line Processing

Switch 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 a command at the prompt. Table 2-1 lists the keyboard shortcuts for entering and editing switch commands.

Table 2-1 Keyboard Shortcuts

Keystrokes	Result
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 command lines you entered. History substitution enables you to access these command lines without retyping them. 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 older commands successively.
Ctrl-N or the **Down Arrow** key[1]	Returns to more recent commands in the history buffer after commands have been recalled with **Ctrl-P** or the **Up Arrow** key. Repeat the key sequence to recall more recent commands.
`Switch# show history`	Lists the last several commands you have entered in EXEC mode.

[1] The Arrow keys function only on ANSI-compatible terminals such as VT100s.

Understanding Cisco IOS Command Modes

Note For complete information about Cisco IOS command modes, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and the *Cisco IOS Configuration Fundamentals Command Reference* at: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/index.htm

The Cisco IOS user interface has many different modes: user EXEC, privileged EXEC (enable), global configuration, interface, subinterface, and protocol-specific. The commands available to you depend on which mode you are in. To get a list of the commands in a given mode, enter a question mark (?) at the system prompt. See the "Getting a List of Commands and Syntax" section on page 2-5 for more information.

When you start a session on the switch, you begin in user mode, also called user EXEC mode. Only a small subset of commands are available in EXEC mode. To have access to all commands, you must enter privileged EXEC mode, also called enable mode. To access the privileged EXEC mode, you must enter a password. When you are in the privileged EXEC mode, you can enter any EXEC command or access global configuration mode. Most EXEC commands are one-time commands, such as **show** commands, which display the current configuration status, and **clear** commands, which reset counters or interfaces. The EXEC commands are not saved when the switch is rebooted.

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

You would use a separate mode called ROMMON when the switch cannot boot up properly. For example, the switch might enter ROMMON mode if it does not find a valid system image when it is booting, or if its configuration file is corrupted. For more information, see the "ROMMON Command-Line Interface" section on page 2-7.

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

Table 2-3 Frequently Used Cisco IOS Command Modes

Mode	What You Use It For	How to Access	Prompt
User EXEC	To connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and display system information.	Log in.	`Switch>`
Privileged EXEC (enable)	To set operating parameters. The privileged command set includes the commands in user EXEC mode, as well as the **configure** command. Use the **configure** command to access the other command modes.	From user EXEC mode, enter the **enable** command and the enable password (if a password has been configured).	`Switch#`
Global configuration	To configure features that affect the system as a whole, such as the system time or switch name.	From privileged EXEC mode, enter the **configure terminal** command.	`Switch(config)#`
Interface configuration	To enable or modify the operation of a 10-Gigabit Ethernet, Gigabit Ethernet, or Fast Ethernet interface with **interface commands.**	From global configuration mode, enter the **interface** *type location* command.	`Switch(config-if)#`
Console configuration	To configure the console interface; from the directly connected console or the virtual terminal; used with Telnet.	From global configuration mode, enter the **line console 0** command.	`Switch(config-line)#`

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

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

Getting a List of Commands and Syntax

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

```
Switch> ?
```

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

To list 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.

```
Switch# 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.

Type **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.

Virtual Console for Standby Supervisor Engine

Catalyst 4500 series switches can be configured with 2 supervisor engines to provide redundancy. When the switch is powered, one of the supervisor engines becomes active and remains active until a switchover occurs. The other supervisor engine remains in standby mode.

Each supervisor engine has its own console port. Access to the standby supervisor engine is possible only through the console port of the standby supervisor engine. Therefore, you must connect to the standby console to access, monitor or debug the standby supervisor.

Virtual Console for Standby Supervisor Engine enables you to access the standby console from the active supervisor engine without requiring a physical connection to the standby console. It uses IPC over EOBC to communicate with the standby supervisor engine and thus emulate the standby console on the active supervisor engine. Only one active standby console session is active at any time.

The Virtual Console for Standby Supervisor Engine enables users who are logged onto the active supervisor engine to remotely execute show commands on the standby supervisor engine and view the results on the active supervisor engine. Virtual Console is available only from the active supervisor engine.

You can access the standby virtual console from the active supervisor engine with the **attach module**, **session module**, or **remote login** commands on the active supervisor engine. You must be in privilege EXEC mode (level 15) to run these commands to access the standby console.

Once you enter the standby virtual console, the terminal prompt automatically changes to "<hostname>-standby-console#" where hostname is the configured name of the switch. The prompt is restored back to the original prompt when you exit the virtual console.

You exit the virtual console with the **exit** or **quit** commands. When the inactivity period of the terminal on the active supervisor engine where you logged in exceeds the configured idle time, you are automatically logged out of the terminal on the active supervisor engine. In such a case, the virtual console session is also terminated. Virtual console session is also automatically terminated when the standby is rebooted. After the standby boots up, you need to create another virtual console session.

To login to the standby supervisor engine using a virtual console, do the following:

```
Switch# session module 2
Connecting to standby virtual console
Type "exit" or "quit" to end this session

Switch-standby-console# exit
Switch#
```

If the standby console is not enabled, the following message appears.

```
Switch-standby-console#
Standby console disabled.
Valid commands are: exit, logout
```

Note The standby virtual console provides the standard features that are available from the supervisor console such as command history, command completion, command help and partial command keywords.

The following limitations apply to the standby virtual console:

- All commands on the virtual console run to completion. It does not provide the auto-more feature; it behaves as if the **terminal length 0** command has been executed. It is also non-interactive. Therefore, a running command cannot be interrupted or aborted by any key sequence on the active supervisor engine. Therefore if a command produces considerable output, the virtual console displays it on the supervisor screen.
- The virtual console is non-interactive. Because the virtual console does not detect the interactive nature of a command, any command that requires user interaction causes the virtual console to wait until the RPC timer aborts the command.

  The virtual console timer is set to 60 seconds. The virtual console returns to its prompt after 60 seconds. During this time, you cannot abort the command from the key board. You must wait for the timer to expire before you continue.
- You cannot use virtual console to view debug and syslog messages that are being displayed on the standby supervisor engine. The virtual console only displays the output of commands that are executed from the virtual console. Other information that is displayed on the real standby console does not appear on the virtual console.

ROMMON Command-Line Interface

ROMMON is a ROM-based program that is involved at power-up or reset, or when a fatal exception error occurs. The switch enters ROMMON 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 ROMMON mode. From the ROMMON mode, you can load a software image manually from Flash memory, from a network server file, or from bootflash.

You can also enter ROMMON mode by restarting the switch and pressing **Ctrl-C** during the first five seconds of startup.

Note **Ctrl-C** is always enabled for 60 seconds after you reboot the switch, even if **Ctrl-C** is configured to be off in the configuration register settings.

When you enter ROMMON mode, the prompt changes to **rommon 1>**. Use the **?** command to see the available ROMMON commands.

For more information about the ROMMON commands, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

http://www.cisco.com/en/US/products/hw/routers/ps380/products_configuration_guide_chapter09186a0080118d19.html

CHAPTER 3

Configuring the Switch for the First Time

This chapter describes how to initially configure a Catalyst 4500 series switch. The information presented here supplements the administration information and procedures in this publication:

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

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

This chapter includes the following major sections:

- Default Switch Configuration, page 3-1
- Configuring DHCP-Based Autoconfiguration, page 3-2
- Configuring the Switch, page 3-8
- Controlling Access to Privileged EXEC Commands, page 3-13
- Recovering a Lost Enable Password, page 3-25
- Modifying the Supervisor Engine Startup Configuration, page 3-25
- Resetting a Switch to Factory Default Settings, page 3-31

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Default Switch Configuration

This section describes the default configurations for the Catalyst 4500 series switch. Table 3-1 shows the default configuration settings for each feature.

Table 3-1 Default Switch Configuration

Feature	Default Settings
Administrative connection	Normal mode
Global switch information	No default value for system name, system contact, and location
System clock	No value for system clock time
Passwords	No passwords are configured for normal mode or enable mode (press the **Return** key)

Table 3-1 Default Switch Configuration (continued)

Feature	Default Settings
Switch prompt	`Switch>`
Interfaces	Enabled, with speed and flow control autonegotiated, and without IP addresses

Configuring DHCP-Based Autoconfiguration

These sections describe how to configure DHCP-based autoconfiguration.

- Understanding DHCP-Based Autoconfiguration, page 3-2
- DHCP Client Request Process, page 3-3
- Configuring the DHCP Server, page 3-3
- Configuring the TFTP Server, page 3-4
- Configuring the DNS Server, page 3-5
- Configuring the Relay Device, page 3-5
- Obtaining Configuration Files, page 3-6
- Example Configuration, page 3-7

If your DHCP server is a Cisco device, or if you are configuring the switch as a DHCP server, refer to the "*IP Addressing and Services*" section in the *Cisco IOS IP and IP Routing Configuration Guide for Cisco IOS Release 12.1* for additional information about configuring DHCP.

Understanding DHCP-Based Autoconfiguration

Note Starting with Release 12.2(20)EW, you can enable DHCP AutoConfiguration by issuing the **write erase** command. This command clears the startup-config in NVRAM. In images prior to Release 12.2(20)EW, this command will not enable autoconfiguration.

DHCP provides configuration information to Internet hosts and internetworking devices. This protocol consists of two components: one component for delivering configuration parameters from a DHCP server to a device and another component that is a mechanism for allocating network addresses to devices. DHCP is built on a client-server model, in which designated DHCP servers allocate network addresses and deliver configuration parameters to dynamically configured devices. The switch can act as both a DHCP client and a DHCP server.

With DHCP-based autoconfiguration, no DHCP client-side configuration is needed on your switch because your switch (the DHCP client) is automatically configured at startup with IP address information and a configuration file. However, you need to configure the DHCP server or the DHCP server feature on your switch for various lease options associated with IP addresses. If you are using DHCP to relay the configuration file location on the network, you might also need to configure a Trivial File Transfer Protocol (TFTP) server and a Domain Name System (DNS) server.

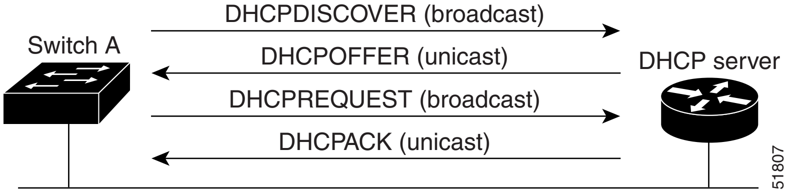
DHCP-based autoconfiguration replaces the BOOTP client functionality on your switch.

DHCP Client Request Process

At startup the switch automatically requests configuration information from a DHCP server if a configuration file is not present on the switch.

Figure 3-1 shows the sequence of messages that are exchanged between the DHCP client and the DHCP server.

Figure 3-1 DHCP Client and Server Message Exchange



The client, Switch A, broadcasts a DHCPDISCOVER message to locate a DHCP server. The DHCP server offers configuration parameters (such as an IP address, subnet mask, gateway IP address, DNS IP address, lease for the IP address, and so forth) to the client in a DHCPOFFER unicast message.

In a DHCPREQUEST broadcast message, the client returns a formal request for the offered configuration information to the DHCP server. The formal request is broadcast so that all other DHCP servers that received the DHCPDISCOVER broadcast message from the client can reclaim the IP addresses that they offered to the client.

The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client. With this message, the client and server are bound, and the client uses the configuration information that it received from the server. The amount of information the switch receives depends on how you configure the DHCP server. For more information, see the "Configuring the DHCP Server" section on page 3-3.

If the configuration parameters sent to the client in the DHCPOFFER unicast message are invalid (if configuration error exists), the client returns a DHCPDECLINE broadcast message to the DHCP server.

The DHCP server sends the client a DHCPNAK denial broadcast message, which means that the offered configuration parameters have not been assigned, that an error has occurred during the negotiation of the parameters, or that the client has been slow in responding to the DHCPOFFER message. (The DHCP server might have assigned the parameters to another client.)

A DHCP client might receive offers from multiple DHCP servers and can accept any of them; however, the client usually accepts the first offer it receives. The offer from the DHCP server is not a guarantee that the IP address will be allocated to the client; however, the server usually reserves the address until the client has had a chance to formally request the address.

Configuring the DHCP Server

A switch can act as both the DHCP client and the DHCP server. By default, the Cisco IOS DHCP server and relay agent features are enabled on your switch.

You should configure the DHCP server, or the DHCP server feature running on your switch, with reserved leases that are bound to each switch by the switch hardware address.

If you want the switch to receive IP address information, you must configure the DHCP server with these lease options:

- IP address of the client (required)
- Subnet mask of the client (required)
- DNS server IP address (optional)
- Router IP address (required)

Note The router IP address is the default gateway address for the switch.

If you want the switch to receive the configuration file from a TFTP server, you must configure the DHCP server with these lease options:

- TFTP server name or IP address (required)
- Boot filename (the name of the configuration file that the client needs) (recommended)
- Host name (optional)

Depending on the settings of the DHCP server or the DHCP server feature running on your switch, the switch can receive IP address information, the configuration file, or both.

If you do not configure the DHCP server, or the DHCP server feature running on your switch, with the lease options described earlier, the switch replies to client requests with only those parameters that are configured. If the IP address and subnet mask are not in the reply, the switch is not configured. If the router IP address or TFTP server name (or IP address) are not found, the switch might send broadcast, instead of unicast, TFTP requests. Unavailability of other lease options does not impact autoconfiguration.

The DHCP server, or the DHCP server feature running on your switch, can be on the same LAN or on a different LAN than the switch. If the DHCP server is running on a different LAN, you should configure a DHCP relay, which forwards broadcast traffic between two directly connected LANs. A router does not forward broadcast packets, but it forwards packets based on the destination IP address in the received packet. For more information on relay devices, see the “Configuring the Relay Device” section on page 3-5.

Configuring the TFTP Server

Based on the DHCP server configuration, the switch attempts to download one or more configuration files from the TFTP server. If you configured the DHCP server to respond to the switch with all the options required for IP connectivity to the TFTP server, and if you configured the DHCP server with a TFTP server name, address, and configuration filename, the switch attempts to download the specified configuration file from the specified TFTP server.

If you did not specify the configuration filename or the TFTP server name, or if the configuration file could not be downloaded, the switch attempts to download a configuration file using various combinations of filenames and TFTP server addresses. The files include the specified configuration filename (if any) and the following files: network-confg, cisconet.cfg, *hostname*.confg, or *hostname*.cfg, where *hostname* is the current hostname of the switch and router-confg and ciscortr.cfg. The TFTP server addresses used include the specified TFTP server address (if any) and the broadcast address (255.255.255.255).

For the switch to successfully download a configuration file, the TFTP server must contain one or more configuration files in its base directory. The files can include the following:

- The configuration file named in the DHCP reply (the actual switch configuration file).
- The network-confg or the cisconet.cfg file (known as the default configuration files).

- The router-confg or the ciscortr.cfg file. (These files contain commands common to all switches. Normally, if the DHCP and TFTP servers are properly configured, these files are not accessed.)

If you specify the TFTP server name in the DHCP server-lease database, you must also configure the TFTP server name-to-IP-address mapping in the DNS-server database.

If the TFTP server you plan to use is on a different LAN from the switch, or if you plan to access it with the switch through the broadcast address (which occurs if the DHCP server response does not contain all the required information described earlier), you must configure a relay to forward the TFTP packets to the TFTP server. For more information, see the “Configuring the Relay Device” section on page 3-5. The preferred solution is to configure either the DHCP server or the DHCP server feature running on your switch with all the required information.

Configuring the DNS Server

The DHCP server, or the DHCP server feature running on your switch, uses the DNS server to resolve the TFTP server name to an IP address. You must configure the TFTP server name-to-IP address map on the DNS server. The TFTP server contains the configuration files for the switch.

You can configure the IP addresses of the DNS servers in the lease database of the DHCP server where the DHCP replies will retrieve them. You can enter up to two DNS server IP addresses in the lease database.

The DNS server can be on the same or on a different LAN as the switch. If it is on a different LAN, the switch must be able to access it through a router.

Configuring the Relay Device

You must configure a relay device to forward received broadcast packets to the destination host whenever a switch sends broadcast packets to which a host on a different LAN must respond. Examples of such broadcast packets are DHCP, DNS, and in some cases, TFTP packets.

If the relay device is a Cisco router, enable IP routing (**ip routing** global configuration command), and configure helper addresses (**ip helper-address** interface configuration command). For example, in Figure 3-2, configure the router interfaces as follows:

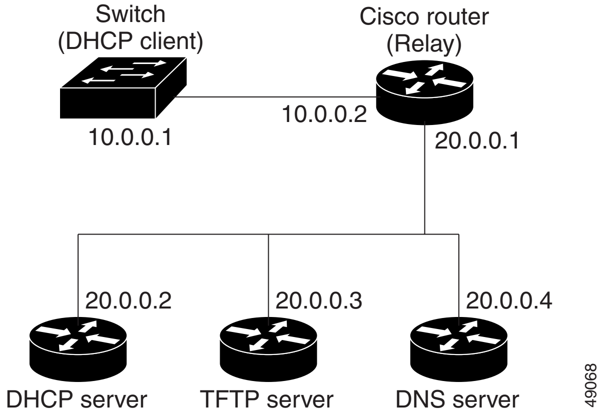
On interface 10.0.0.2:

```
router(config-if)# ip helper-address 20.0.0.2
router(config-if)# ip helper-address 20.0.0.3
router(config-if)# ip helper-address 20.0.0.4
```

On interface 20.0.0.1

```
router(config-if)# ip helper-address 10.0.0.1
```

Figure 3-2 Relay Device Used in Autoconfiguration



Obtaining Configuration Files

Depending on the availability of the IP address and the configuration filename in the DHCP reserved lease, the switch obtains its configuration information in these ways:

- The IP address and the configuration filename are reserved for the switch and provided in the DHCP reply (one-file read method).

  The switch receives its IP address, subnet mask, TFTP server address, and the configuration filename from either the DHCP server or the DHCP server feature running on your switch. The switch sends a unicast message to the TFTP server to retrieve the named configuration file from the base directory of the server, and upon receipt, completes its boot-up process.

- The IP address and the configuration filename is reserved for the switch, but the TFTP server address is not provided in the DHCP reply (one-file read method).

  The switch receives its IP address, subnet mask, and the configuration filename from either the DHCP server or the DHCP server feature running on your switch. The switch sends a broadcast message to a TFTP server to retrieve the named configuration file from the base directory of the server, and upon receipt, completes its boot-up process.

- Only the IP address is reserved for the switch and provided in the DHCP reply. The configuration filename is not provided (two-file read method).

  The switch receives its IP address, subnet mask, and the TFTP server address from either the DHCP server or the DHCP server feature running on your switch. The switch sends a unicast message to the TFTP server to retrieve the network-confg or cisconet.cfg default configuration file. (If the network-confg file cannot be read, the switch reads the cisconet.cfg file.)

  The default configuration file contains the host names-to-IP-address mapping for the switch. The switch fills its host table with the information in the file and obtains its host name. If the host name is not found in the file, the switch uses the host name in the DHCP reply. If the host name is not specified in the DHCP reply, the switch uses the default *Switch* as its host name.

  After obtaining its host name from the default configuration file or the DHCP reply, the switch reads the configuration file that has the same name as its host name (*hostname*-confg or *hostname*.cfg, depending on whether or not the network-confg file or the cisconet.cfg file was read earlier) from the TFTP server. If the cisconet.cfg file is read, the filename of the host is truncated to eight characters.

If the switch cannot read the network-confg, cisconet.cfg, or the hostname file, it reads the router-confg file. If the switch cannot read the router-confg file, it reads the ciscortr.cfg file.

Note The switch broadcasts TFTP server requests provided that one of these conditions is met: 1) the TFTP server is not obtained from the DHCP replies; 2) all attempts to read the configuration file through unicast transmissions fail, or 3) the TFTP server name cannot be resolved to an IP address.

Example Configuration

Figure 3-3 shows a network example for retrieving IP information using DHCP-based autoconfiguration.

Figure 3-3 DHCP-Based Autoconfiguration Network Example

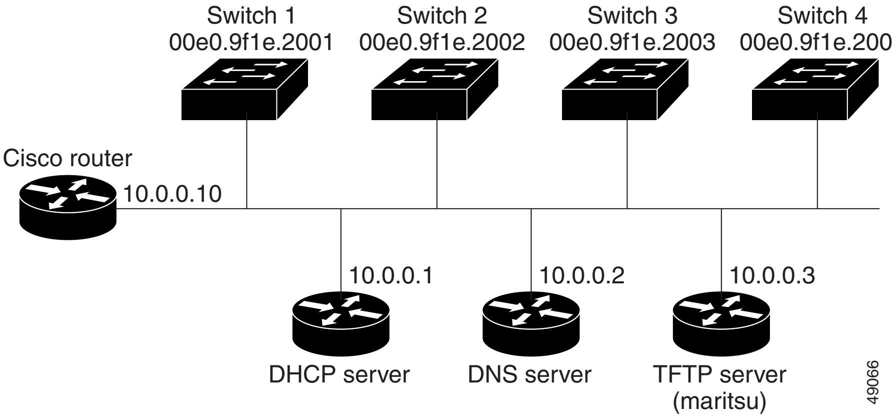


Table 3-2 shows the configuration of the reserved leases on either the DHCP server or the DHCP server feature running on your switch.

Table 3-2 DHCP Server Configuration

	Switch 1	Switch 2	Switch 3	Switch 4
Binding key (hardware address)	00e0.9f1e.2001	00e0.9f1e.2002	00e0.9f1e.2003	00e0.9f1e.2004
IP address	10.0.0.21	10.0.0.22	10.0.0.23	10.0.0.24
Subnet mask	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Router address	10.0.0.10	10.0.0.10	10.0.0.10	10.0.0.10
DNS server address	10.0.0.2	10.0.0.2	10.0.0.2	10.0.0.2
TFTP server name	*maritsu* or *10.0.0.3*	*maritsu* or *10.0.0.3*	*maritsu* or *10.0.0.3*	*maritsu* or *10.0.0.3*
Boot filename (configuration file) (optional)	switch1-confg	switch2-confg	switch3-confg	switch4-confg
Host name (optional)	switch1	switch2	switch3	switch4

DNS Server Configuration

The DNS server maps the TFTP server name *maritsu* to IP address 10.0.0.3.

TFTP Server Configuration (on UNIX)

The TFTP server base directory is set to /tftpserver/work/. This directory contains the network-confg file used in the two-file read method. This file contains the host name that you plan to assign to the switch based on its IP address. The base directory also contains a configuration file for each switch (*switch1-confg*, *switch2-confg*, and so forth) as shown in the following display:

```
prompt> cd /tftpserver/work/
prompt> ls
network-confg
switch1-confg
switch2-confg
switch3-confg
switch4-confg
prompt> cat network-confg
ip host switch1 10.0.0.21
ip host switch2 10.0.0.22
ip host switch3 10.0.0.23
ip host switch4 10.0.0.24
```

DHCP Client Configuration

No configuration file is present on Switch 1 through Switch 4.

Configuration Explanation

In Figure 3-3, Switch 1 reads its configuration file as follows:

- Switch 1 obtains its IP address 10.0.0.21 from the DHCP server.
- If no configuration filename is given in the DHCP server reply, Switch 1 reads the network-confg file from the base directory of the TFTP server.
- Switch 1 adds the contents of the network-confg file to its host table.
- Switch 1 reads its host table by indexing its IP address 10.0.0.21 to its host name (switch1).
- Switch 1 reads the configuration file that corresponds to its host name; for example, it reads *switch1-confg* from the TFTP server.

Switches 2 through 4 retrieve their configuration files and IP addresses in the same way.

Configuring the Switch

The following sections describe how to configure your switch:

- Using Configuration Mode to Configure Your Switch, page 3-8
- Verifying the Running Configuration Settings, page 3-9
- Saving the Running Configuration Settings to Your Start-Up File, page 3-10
- Reviewing the Configuration in NVRAM, page 3-10
- Configuring a Default Gateway, page 3-11
- Configuring a Static Route, page 3-11

Using Configuration Mode to Configure Your Switch

To configure your switch from configuration mode, perform this procedure:

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

Step 2 After a few seconds, you will see the user EXEC prompt (`Switch>`). Now, you may want to enter privileged EXEC mode, also known as enable mode. Type **enable** to enter enable mode:

```
Switch> enable
```

Note You must be in enable mode to make configuration changes.

The prompt will change to the enable prompt (#):

```
Switch#
```

Step 3 At the enable prompt (#), enter the **configure terminal** command to enter global configuration mode:

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

Step 4 At the global configuration mode prompt, enter the **interface** *type slot/interface* command to enter interface configuration mode:

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

Step 5 In either of these configuration modes, enter changes to the switch configuration.

Step 6 Enter the **end** command to exit configuration mode.

Step 7 Save your settings. (See the “Saving the Running Configuration Settings to Your Start-Up File” section on page 3-10.)

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.

Verifying the Running Configuration Settings

To verify the configuration settings you entered or the changes you made, enter the **show running-config** command at the enable prompt (#), as shown in this example:

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

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch

<...output truncated...>

!
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
Switch#
```

Saving the Running Configuration Settings to Your Start-Up File

Caution 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.

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

```
Switch# copy running-config startup-config
```

Reviewing the Configuration in NVRAM

To display information stored in NVRAM, enter the **show startup-config** EXEC command.

The following example shows a typical system configuration:

```
Switch# show startup-config
Using 1579 out of 491500 bytes, uncompressed size = 7372 bytes
Uncompressed configuration from 1579 bytes to 7372 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service compress-config
!
hostname Switch
!
!
ip subnet-zero
!
!
!
!
interface GigabitEthernet1/1
 no snmp trap link-status
!
interface GigabitEthernet1/2
 no snmp trap link-status
!--More--

<...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 dsipcon mop telnet rlogin udptn nasi
!
end

Switch#
```

Configuring a Default Gateway

Note The switch uses the default gateway only when it is not configured with a routing protocol.

Configure a default gateway to send data to subnets other than its own when the switch is not configured with a routing protocol. The default gateway must be the IP address of an interface on a router that is directly connected to the switch.

To configure a default gateway, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`ip default-gateway`** *`IP-address`*	Configures a default gateway.
Step 2	`Switch#` **`show ip route`**	Verifies that the default gateway is correctly displayed in the IP routing table.

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

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

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

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	`Switch(config)#` **`ip route`** *`dest_IP_address mask`* {*`forwarding_IP`* | **`vlan`** *`vlan_ID`*}	Configures a static route to the remote network.
Step 2	`Switch#` **`show running-config`**	Verifies that the static route is displayed correctly.

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:

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

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

```
Switch# 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

Switch#
```

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

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

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

```
Switch# show running-config
Building configuration...
.
<...output truncated...>
.
ip default-gateway 172.20.52.35
ip classless
ip route 171.20.5.3 255.255.255.255 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

Switch#
```

Controlling Access to Privileged EXEC Commands

The procedures in these sections let you control access to the system configuration file and privileged EXEC commands:

- Setting or Changing a Static enable Password, page 3-13
- Using the enable password and enable secret Commands, page 3-13
- Setting or Changing a Privileged Password, page 3-14
- Encrypting Passwords, page 3-22
- Encrypting Passwords, page 3-22
- Configuring Multiple Privilege Levels, page 3-23

Setting or Changing a Static enable Password

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

Table 3-1

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

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

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

For instructions on how to display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-24.

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** command. Both commands configure an encrypted password that you must enter to access the enable mode (the default) or any other privilege level that you specify.

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 one of these tasks:

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

When you enter either of these password commands with the **level** option, you 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 the password with the **more system:running-config** command, the password displays the password in encrypted form.

If you specify an encryption type, you must provide an encrypted password—an encrypted password you copy from another Catalyst 4500 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 3-25 for more information.

For information on how to display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-24.

Setting or Changing a Privileged Password

To set or change a privileged password, perform this task:

Table 3-2

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

For information on how to display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-24.

Controlling Switch Access with TACACS+

This section describes how to enable and configure TACACS+, which provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through authentication, authorization, accounting (AAA) and can be enabled only through AAA commands.

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Security Command Reference, Release 12.2*.

This section contains this configuration information:

- Understanding TACACS+, page 3-15
- TACACS+ Operation, page 3-17
- Configuring TACACS+, page 3-17
- Displaying the TACACS+ Configuration, page 3-22

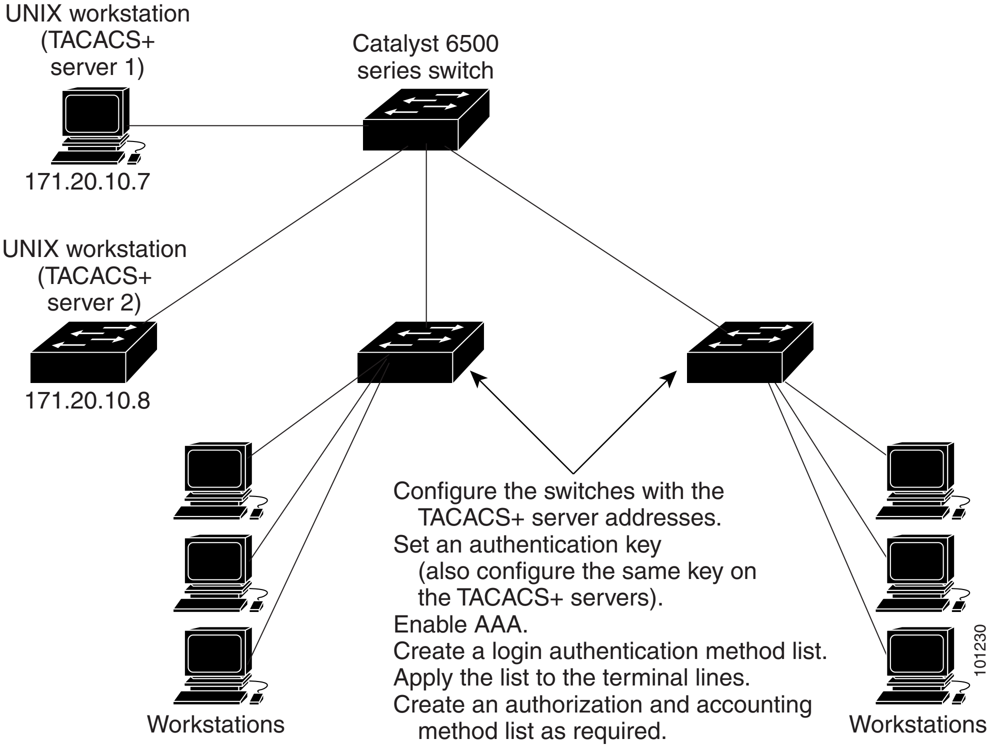
Understanding TACACS+

TACACS+ is a security application that provides centralized validation of users attempting to gain access to your switch. TACACS+ services are maintained in a database on a TACACS+ daemon typically running on a UNIX or Windows NT workstation. You should have access to and should configure a TACACS+ server before configuring TACACS+ features on your switch.

TACACS+ provides for separate and modular AAA facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be locked into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The goal of TACACS+ is to provide a method for managing multiple network access points from a single management service. Your switch can be a network access server along with other Cisco routers and access servers. A network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks as shown in Figure 3-4.

Figure 3-4 Typical TACACS+ Network Configuration


UNIX workstation (TACACS+ server 1)
171.20.10.7
Catalyst 6500 series switch
UNIX workstation (TACACS+ server 2)
171.20.10.8
Workstations
Workstations
101230

Configure the switches with the TACACS+ server addresses.
Set an authentication key (also configure the same key on the TACACS+ servers).
Enable AAA.
Create a login authentication method list.
Apply the list to the terminal lines.
Create an authorization and accounting method list as required.

TACACS+ administered through the AAA security services can provide these services:

- Authentication—Provides complete control of authentication through login and password dialog, challenge and response, and messaging support.

  The authentication facility can conduct a dialog with the user (such as, after a username and password are provided, to challenge a user with several questions such as home address, mother's maiden name, service type, and social security number). The TACACS+ authentication service can also send messages to user screens. For example, a message could notify users that their passwords must be changed because of the company's password aging policy.

- Authorization—Provides strict control over user capabilities for the duration of the user's session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on the commands a user can execute with the TACACS+ authorization feature.

- Accounting—Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. Accounting records include user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the switch and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between the switch and the TACACS+ daemon are encrypted.

You need a system running the TACACS+ daemon software to use TACACS+ on your switch.

TACACS+ Operation

When a user attempts a simple ASCII login by authenticating to a switch using TACACS+, this process occurs:

1. When the connection is established, the switch contacts the TACACS+ daemon to obtain a username prompt, which is then displayed to the user. The user enters a username, and the switch then contacts the TACACS+ daemon to obtain a password prompt. The switch displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.

   TACACS+ allows a conversation between the daemon and the user until the daemon receives enough information to authenticate the user. The daemon prompts for a username and password combination, but can include other items such as the user's mother's maiden name.

2. The switch eventually receives one of these responses from the TACACS+ daemon:
   - ACCEPT—The user is authenticated and service can begin. If the switch is configured to require authorization, authorization begins at this time.
   - REJECT—The user is not authenticated. The user can be denied access or is prompted to retry the login sequence, depending on the TACACS+ daemon.
   - ERROR—An error occurred at some time during authentication with the daemon or in the network connection between the daemon and the switch. If an ERROR response is received, the switch typically tries to use an alternative method for authenticating the user.
   - CONTINUE—The user is prompted for additional authentication information.

   After authentication, the user undergoes an additional authorization phase if authorization has been enabled on the switch. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3. If TACACS+ authorization is required, the TACACS+ daemon is again contacted, and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the response contains data in the form of attributes that direct the EXEC or NETWORK session for that user and the services that the user can access:
   - Telnet, Secure Shell (SSH), rlogin, or privileged EXEC services
   - Connection parameters, including the host or client IP address, access list, and user timeouts

Configuring TACACS+

This section describes how to configure your switch to support TACACS+. At a minimum, you must identify the host or hosts maintaining the TACACS+ daemon and define the method lists for TACACS+ authentication. You can optionally define method lists for TACACS+ authorization and accounting. A method list defines the sequence and methods used to authenticate, to authorize, or to keep accounts on a user. You can use method lists to designate one or more security protocols, ensuring a backup system if the initial method fails. The software uses the first method listed to authenticate, to authorize, or to keep accounts on users; if that method does not respond, the software selects the next method in the list. This process continues until there is successful communication with a listed method or the method list is exhausted.

This section contains this configuration information:

- Default TACACS+ Configuration, page 3-18
- Identifying the TACACS+ Server Host and Setting the Authentication Key, page 3-18
- Configuring TACACS+ Login Authentication, page 3-19

- Configuring TACACS+ Authorization for Privileged EXEC Access and Network Services, page 3-21
- Starting TACACS+ Accounting, page 3-21

Default TACACS+ Configuration

TACACS+ and AAA are disabled by default.

To prevent a lapse in security, you cannot configure TACACS+ through a network management application. When enabled, TACACS+ can authenticate users accessing the switch through the CLI.

Note Although TACACS+ configuration is performed through the CLI, the TACACS+ server authenticates HTTP connections that have been configured with a privilege level of 15.

Identifying the TACACS+ Server Host and Setting the Authentication Key

You can configure the switch to use a single server or AAA server groups in order to group existing server hosts for authentication. You can group servers to select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list and contains the list of IP addresses of the selected server hosts.

Beginning in privileged EXEC mode, follow these steps to identify the IP host or host maintaining TACACS+ server and optionally set the encryption key:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**tacacs-server host** *hostname* [**port** *integer*] [**timeout** *integer*] [**key** *string*]	Identify the IP host or hosts maintaining a TACACS+ server. Enter this command multiple times to create a list of preferred hosts. The software searches for hosts in the order in which you specify them. • For *hostname*, specify the name or IP address of the host. • (Optional) For **port** *integer*, specify a server port number. The default is port 49. The range is 1 to 65535. • (Optional) For **timeout** *integer*, specify a time in seconds the switch waits for a response from the daemon before it times out and declares an error. The default is 5 seconds. The range is 1 to 1000 seconds. • (Optional) For **key** *string*, specify the encryption key for encrypting and decrypting all traffic between the switch and the TACACS+ daemon. You must configure the same key on the TACACS+ daemon for encryption to succeed.
Step 3	**aaa new-model**	Enable AAA.
Step 4	**aaa group server tacacs+** *group-name*	(Optional) Define the AAA server-group with a group name. This command puts the switch in a server group subconfiguration mode.
Step 5	**server** *ip-address*	(Optional) Associate a particular TACACS+ server with the defined server group. Repeat this step for each TACACS+ server in the AAA server group. Each server in the group must be previously defined in Step 2.
Step 6	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 7	**show tacacs**	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the specified TACACS+ server name or address, use the **no tacacs-server host** *hostname* global configuration command. To remove a server group from the configuration list, use the **no aaa group server tacacs+** *group-name* global configuration command. To remove the IP address of a TACACS+ server, use the **no server ip-address** server group subconfiguration command.

Configuring TACACS+ Login Authentication

To configure AAA authentication, define a named list of authentication methods and then apply that list to various ports. The method list defines the types of authentication you intend to perform and the sequence in which you intend to perform them; you must apply the list to a specific port before you can perform any of the defined authentication methods. The only exception is the default method list (which, by coincidence, is named *default*). The default method list is automatically applied to all ports except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list describes the sequence and authentication methods that must be queried to authenticate a user. You can designate one or more security protocols for authentication, ensuring a backup system for authentication in case the initial method fails. 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—meaning that the security server or local username database responds by denying the user access—the authentication process stops, and no other authentication methods are attempted.

Beginning in privileged EXEC mode, follow these steps to configure login authentication:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa new-model**	Enable AAA.

	Command	Purpose
Step 3	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]	Create a login authentication method list. • To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that you plan to use in default situations. The default method list is automatically applied to all ports. • For *list-name*, specify a character string to name the list you are creating. • For *method1...*, specify the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. Select one of these methods: • **enable**—Use the enable password for authentication. Before you can use this authentication method, you must define an enable password by using the **enable** *password* global configuration command. • **group tacacs+**—Uses TACACS+ authentication. Before you can use this authentication method, you must configure the TACACS+ server. For more information, see the "Identifying the TACACS+ Server Host and Setting the Authentication Key" section on page 3-18. • **line**—Use the line password for authentication. Before you can use this authentication method, you must define a line password. Use the **password** *password* line configuration command. • **local**—Use the local username database for authentication. You must enter username information in the database. Use the **username** *password* global configuration command. • **local-case**—Use a case-sensitive local username database for authentication. You must enter username information in the database by using the **username** *name* **password** global configuration command. • **none**—Do not use any authentication for login.
Step 4	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]	Enter line configuration mode, and configure the lines to which you want to apply the authentication list.
Step 5	**login authentication** {**default** | *list-name*}	Apply the authentication list to a line or set of lines. • If you specify **default**, use the default list created with the **aaa authentication login** command. • For *list-name*, specify the list created with the **aaa authentication login** command.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show running-config**	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable AAA authentication, use the **no aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] global configuration command. To either disable TACACS+ authentication for logins or to return to the default value, use the **no login authentication** {**default** | *list-name*} line configuration command.

Configuring TACACS+ Authorization for Privileged EXEC Access and Network Services

AAA authorization limits the services available to a user. When AAA authorization is enabled, the switch uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

You can use the **aaa authorization** global configuration command with the **tacacs+** keyword to set parameters that restrict a user's network access to privileged EXEC mode.

The **aaa authorization exec tacacs+ local** command sets these authorization parameters:

- Use TACACS+ for privileged EXEC access authorization if authentication was performed by using TACACS+.
- Use the local database if authentication was not performed by using TACACS+.

Note Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

Beginning in privileged EXEC mode, follow these steps to specify TACACS+ authorization for privileged EXEC access and network services:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa authorization network tacacs+**	Configure the switch for user TACACS+ authorization for all network-related service requests.
Step 3	**aaa authorization exec tacacs+**	Configure the switch for user TACACS+ authorization if the user has privileged EXEC access. The **exec** keyword might return user profile information (such as **autocommand** information).
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable authorization, use the **no aaa authorization** {**network** | **exec**} *method1* global configuration command.

Starting TACACS+ Accounting

The AAA accounting feature tracks the services that users are accessing and the amount of network resources that they are consuming. When AAA accounting is enabled, the switch reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, or auditing.

Beginning in privileged EXEC mode, follow these steps to enable TACACS+ accounting for each Cisco IOS privilege level and for network services:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa accounting network start-stop tacacs+**	Enable TACACS+ accounting for all network-related service requests.
Step 3	**aaa accounting exec start-stop tacacs+**	Enable TACACS+ accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable accounting, use the **no aaa accounting** {**network** | **exec**} {**start-stop**} *method1...* global configuration command.

Displaying the TACACS+ Configuration

To display TACACS+ server statistics, use the **show tacacs** privileged EXEC command.

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:

Table 3-3

Command	Purpose
`Switch(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 having lost or forgotten the encrypted password. See the "Recovering a Lost Enable Password" section on page 3-25 for more information.

For information on how to display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-24.

Configuring Multiple Privilege Levels

By default, 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 fairly widely. If you want more restricted access to the **configure** command, you can assign it level 3 security and distribute that password to fewer users.

The procedures in the following sections describe how to configure additional levels of security:

- Setting the Privilege Level for a Command, page 3-23
- Changing the Default Privilege Level for Lines, page 3-23
- Logging In to a Privilege Level, page 3-24
- Exiting a Privilege Level, page 3-24
- Displaying the Password, Access Level, and Privilege Level Configuration, page 3-24

Setting the Privilege Level for a Command

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

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

For information on how to display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-24.

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:

Table 3-4

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

For information on how to display the password or access level configuration, see the "Displaying the Password, Access Level, and Privilege Level Configuration" section on page 3-24.

Logging In to a Privilege Level

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

Table 3-5

Command	Purpose
`Switch#` **`enable`** *`level`*	Logs in to a specified privilege level.

Exiting a Privilege Level

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

Table 3-6

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

Displaying the Password, Access Level, and Privilege Level Configuration

To display detailed password information, perform this task:

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

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

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

Current configuration:
!
version 12.0
service timestamps debug datetime localtime
service timestamps log datetime localtime
no service password-encryption
!
hostname Switch
!
boot system flash sup-bootflash
enable password lab
!
<...output truncated...>
```

This example shows how to display the privilege level configuration:

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

Recovering a Lost Enable Password

Note For more information on the configuration register which is preconfigured in NVRAM, see "Configuring the Software Configuration Register" section on page 3-26.

Perform these steps to recover a lost enable password:

Step 1 Connect to the console interface.

Step 2 Stop the boot sequence and enter ROM monitor by pressing **Ctrl-C** during the first 5 seconds of bootup.

Step 3 Configure the switch to boot-up without reading the configuration memory (NVRAM).

Step 4 Reboot the system.

Step 5 Access enable mode (this can be done without a password if a password has not been configured).

Step 6 View or change the password, or erase the configuration.

Step 7 Reconfigure the switch to boot-up and read the NVRAM as it normally does.

Step 8 Reboot the system.

Modifying the Supervisor Engine Startup Configuration

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

- Understanding the Supervisor Engine Boot Configuration, page 3-25
- Configuring the Software Configuration Register, page 3-26
- Specifying the Startup System Image, page 3-30
- Controlling Environment Variables, page 3-31

Understanding the Supervisor Engine Boot Configuration

The supervisor engine boot process involves two software images: ROM monitor and supervisor engine software. When the switch is booted or reset, the ROMMON code is executed. Depending on the NVRAM configuration, the supervisor engine either stays in ROMMON 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 3-27. The BOOT environment variable is described in the "Specifying the Startup System Image" section on page 3-30.

Understanding the ROM Monitor

The ROM monitor (ROMMON) is invoked at switch bootup, reset, or when a fatal exception occurs. The switch enters ROMMON 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 ROMMON mode. From ROMMON mode, you can manually load a software image from bootflash or a Flash disk, or you can boot up from the management interface. ROMMON mode loads a primary image from which you can configure a secondary image to boot up from a specified source either locally or through the network using the BOOTLDR environment variable. This variable is described in the "Switch#" section on page 3-31.

You can also enter ROMMON mode by restarting the switch and then pressing **Ctrl-C** during the first five 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 ROMMON mode.

Note **Ctrl-C** is always enabled for five seconds after you reboot the switch, regardless of whether the configuration-register setting has **Ctrl-C** disabled.

The ROM monitor has these features:

- Power-on confidence test
- Hardware initialization
- Boot capability (manual bootup and autoboot)
- File system (read-only while in ROMMON)

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 preconfigured in NVRAM.

Here are some reasons why you might want to change the software configuration register settings:

- To select a boot source and default boot filename
- To control broadcast addresses
- To set the console terminal baud rate
- To load operating software from Flash memory
- To recover a lost password
- To manually boot the system using the **boot** command at the bootstrap program prompt
- To force an automatic bootup 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

Caution To avoid possibly halting the Catalyst 4500 series switch switch, remember that valid configuration register settings might be combinations of settings and not just the individual settings listed in Table 3-3. For example, the factory default value of 0x2101 is a combination of settings.

Table 3-3 lists the meaning of each of the software configuration memory bits. Table 3-4 defines the *boot field*.

Table 3-3 Software Configuration Register Bits

Bit Number[1]	Hexadecimal	Meaning
00 to 03	0x0000 to 0x000F	Boot field (see Table 3-4)
04	0x0010	Unused
05	0x0020	Bit two of console line speed
06	0x0040	Causes system software to ignore NVRAM contents
07	0x0080	OEM[2] bit enabled
08	0x0100	Unused
09	0x0200	Unused
10	0x0400	IP broadcast with all zeros
11 to 12	0x0800 to 0x1000	Bits one and zero of Console line speed (default is 9600 baud)
13	0x2000	Loads ROM monitor after netboot fails
14	0x4000	IP broadcasts do not have network numbers

[1] The factory default value for the configuration register is 0x2101. This value is a combination of the following: binary bit 13, bit 8 = 0x0100 and binary bits 00 through 03 = 0x0001. (See Table 3-4.)

[2] OEM = original equipment manufacturer.

Table 3-4 Explanation of Boot Field (Configuration Register Bits 00 to 03)

Boot Field	Meaning
00	Stays at the system bootstrap prompt (does not autoboot).
01	Boots the first system image in onboard Flash memory.
02 to 0F	Autoboots using image(s) specified by the BOOT environment variable. If more than one image is specified, the switch attempts to boot the first image specified in the BOOT variable. As long as the switch can successfully boot from this image, the same image will be used on a reboot. If the switch fails to boot from the image specified in the BOOT variable, the switch will try to boot from the next image listed in the BOOT variable. If the end of the BOOT variable is reached without the switch booting successfully, the switch attempts the boot from the beginning of the BOOT variable. The autoboot continues until the switch successfully boots from one of the images specified in the BOOT variable.

Modifying the Boot Field and Using the boot Command

The configuration register boot field determines whether the switch loads an operating system image and, if so, where it obtains this system image. The following sections describe how to use and set the configuration register boot field and the procedures you must perform to modify the configuration register boot field. In ROMMON, you can use the **confreg** command to modify the configuration register and change boot settings.

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

Note The factory default configuration register setting for systems and spares is 0x2101. However, the recommended value is 0x0102.

When the boot field is set to either 00 or 01 (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 00, you must boot up the operating system manually by issuing the **boot** command at the system bootstrap or ROMMON prompt.
- When the boot field is set to 01, the system boots the first image in the 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.

Caution If you set bootfield to a value between 0-0-1-0 and 1-1-1-1, you must specify a value in the **boot system** command, else the switch cannot boot up and will remain in ROMMON.

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 up from a specific Flash image (using the **boot system flash** *filename* command).

You can also use the **boot** command to boot up images stored in the compact Flash cards located in slot 0 on the supervisor engine.

Modifying the Boot Field

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

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

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

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

```
Switch> enable
Password:
Switch#
```

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

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Switch(config)#
```

Step 3 Configure the configuration register to 0x102 as follows:

```
Switch(config)# config-register 0x102
```

Set the contents of the configuration register by specifying the *value* command variable, where *value* is a hexadecimal number preceded by 0x (see Table 3-3 on page 3-27).

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 is rebooted.

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

```
Configuration register is 0x141 (will be 0x102 at next reload)
```

Step 6 Save your settings. (See the "Saving the Running Configuration Settings to Your Start-Up File" section on page 3-10. 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 up.

Verifying the Configuration Register Setting

Enter the **show version** EXEC command to verify the current configuration register setting. In ROMMON mode, enter the **show version** command to verify the configuration register setting.

To verify the configuration register setting for the switch, perform this task:

Table 3-7

Command	Purpose
`Switch# show version`	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 ROMMON mode and waits for you to enter ROM monitor commands.

```
Switch#show version
Cisco Internetwork Operating System Software
IOS (tm) Catalyst 4000 L3 Switch Software (cat4000-IS-M), Experimental
Version 12.1(20010828:211314) [cisco 105]
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Thu 06-Sep-01 15:40 by
Image text-base:0x00000000, data-base:0x00ADF444

ROM:1.15
Switch uptime is 10 minutes
System returned to ROM by reload
Running default software

cisco Catalyst 4000 (MPC8240) processor (revision 3) with 262144K bytes
of memory.
Processor board ID Ask SN 12345
Last reset from Reload
Bridging software.
```

```
49 FastEthernet/IEEE 802.3 interface(s)
20 Gigabit Ethernet/IEEE 802.3 interface(s)
271K bytes of non-volatile configuration memory.

Configuration register is 0xEC60

Switch#
```

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.

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*.

Use the following sections to configure your switch to boot from Flash memory. Flash memory can be either single in-line memory modules (SIMMs) or Flash disks. Check the appropriate hardware installation and maintenance guide for information about types of Flash memory.

Using Flash Memory

Flash memory allows you to do the following:

- Copy the system image to Flash memory using TFTP
- Boot the system from Flash memory either automatically or manually
- Copy the Flash memory image to a network server using TFTP or RCP

Flash Memory Features

Flash memory allows you to do the following:

- Remotely load multiple system software images through TFTP or RCP transfers (one transfer for each file loaded)
- Boot a switch manually or automatically from a system software image stored in Flash memory (you can also boot directly from ROM)

Security Precautions

Note the following security precaution when loading from Flash memory:

Caution You can only change the system image stored in Flash memory from privileged EXEC level on the console terminal.

Configuring Flash Memory

To configure your switch to boot from Flash memory, perform the following procedure. (Refer to the appropriate hardware installation and maintenance publication for complete instructions on installing the hardware.)

Step 1 Copy a system image to Flash memory using TFTP or other protocols. Refer to the “Cisco IOS File Management” and “Loading and Maintaining System Images” chapters in the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.2, at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fun_c/fcprt2/fcd203.htm

Step 2 Configure the system to boot automatically from the desired 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 3-27, for more information on modifying the configuration register.

Step 3 Save your configurations.

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

Controlling Environment Variables

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

Note When you use the **boot system and boot bootldr** global configuration commands, you affect only the running configuration. To save the configuration for future use, you must save the environment variable settings to your startup configuration, which places the information under ROM monitor control. Enter the **copy system:running-config nvram:startup-config** command to save the environment variables from your running configuration to your startup configuration.

You can view the contents of the BOOT and BOOTLDR variables using the **show bootvar** command. This command displays the settings for these variables as they exist in the startup configuration and in the running configuration if a running configuration setting differs from a startup configuration setting. This example shows how to check the BOOT and BOOTLDR variables on the switch:

```
Switch# show bootvar
BOOTLDR variable = bootflash:cat4000-is-mz,1;
Configuration register is 0x0
Switch#
```

Resetting a Switch to Factory Default Settings

Manufacturing and repair centers can use the **erase /all non-default** command to do the following:

- Clear the non-volatile configurations and states of the local supervisor engine (NVRAM and flashes).
- Set the factory default parameters on the Catalyst 4500 series switch before it is ready to ship to a customer.

For example, entering this command can generate the following output:

```
Switch# erase /all non-default
```

```
Erase and format operation will destroy all data in non-volatile storage.  Continue?
[confirm]
Formatting bootflash: ...

Format of bootflash complete
Erasing nvram:
Erasing cat4000_flash:
Clearing crashinfo:data
Clearing the last power failure timestamp
Clearing all ROMMON variables
Setting default ROMMON variables:
     ConfigReg=0x2101
     PS1=rommon ! >
     EnableAutoConfig=1
Setting vtp mode to transparent
%WARNING! Please reboot the system for the changes to take effect
Switch#
00:01:48: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
Switch#
```

If the Catalyst 4500 series switch is accessible to an tftp server, you can copy an image to the bootflash memory with the tftp command:

```
Switch# copy tftp://192.20.3.123/tftpboot/abc/cat4500-entservices-mz.bin bootflash:
```

When the copying completes, you can reboot the just-copied Catalyst 4500 series switch image to the image stored in the bootflash memory with the **reload** command:

```
Switch# reload

 System configuration has been modified. Save? [yes/no]: no
 Proceed with reload? [confirm]

 00:06:17: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload Command.
```

To see details about the default parameters set by the **erase /all non-default** command, see the usage guidelines for the **erase** command page in the *Catalyst 4500 Series Switch Command Reference*.

CHAPTER 4

Administering the Switch

This chapter describes how to perform one-time operations to administer the Catalyst 4500Series switch.

This chapter also describes how to install and configure the Embedded CiscoView network management system to provide a graphical representation of a Catalyst 4500 series switch and to provide a GUI-based management and configuration interface.

This chapter includes the following major sections:

Managing the System Time and Date

You can configure the system time and date on your switch manually or automatically by using Network Time Protocol (NTP).

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

These sections contain this configuration information:

The System Clock

The core of the time service is the system clock, which monitors the date and time. This clock starts when the system starts.

The system clock can provide time to these services:

- User **show** commands
- Logging and debugging messages

The system clock keeps track of time internally based on Universal Time Coordinated (UTC), also known as Greenwich Mean Time (GMT). You can configure information about the local time zone and summer time (daylight saving time) so that the time is correct for the local time zone.

The system clock keeps track of whether the time is *authoritative* or not (whether it has been set by a time source considered to be authoritative). If it is not authoritative, the time is available only for display purposes and is not redistributed. For configuration information, see the "Configuring Time and Date Manually" section on page 4-11.

Understanding Network Time Protocol

The NTP is designed to synchronize a network of devices. NTP runs over User Datagram Protocol (UDP), which runs over IP. NTP is documented in RFC 1305.

An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two devices to within a millisecond of one another.

NTP uses the concept of a *stratum* to describe how many NTP hops away a device is from an authoritative time source. A stratum 1 time server has a radio or atomic clock directly attached, a stratum 2 time server receives its time through NTP from a stratum 1 time server, and so on. A device running NTP automatically chooses as its time source the device with the lowest stratum number with which it communicates through NTP. This strategy effectively builds a self-organizing tree of NTP speakers.

NTP avoids synchronizing to a device whose time might not have been synchronized. NTP also compares the time reported by several devices and does not synchronize to a device whose time is significantly different than the others, even if its stratum is lower.

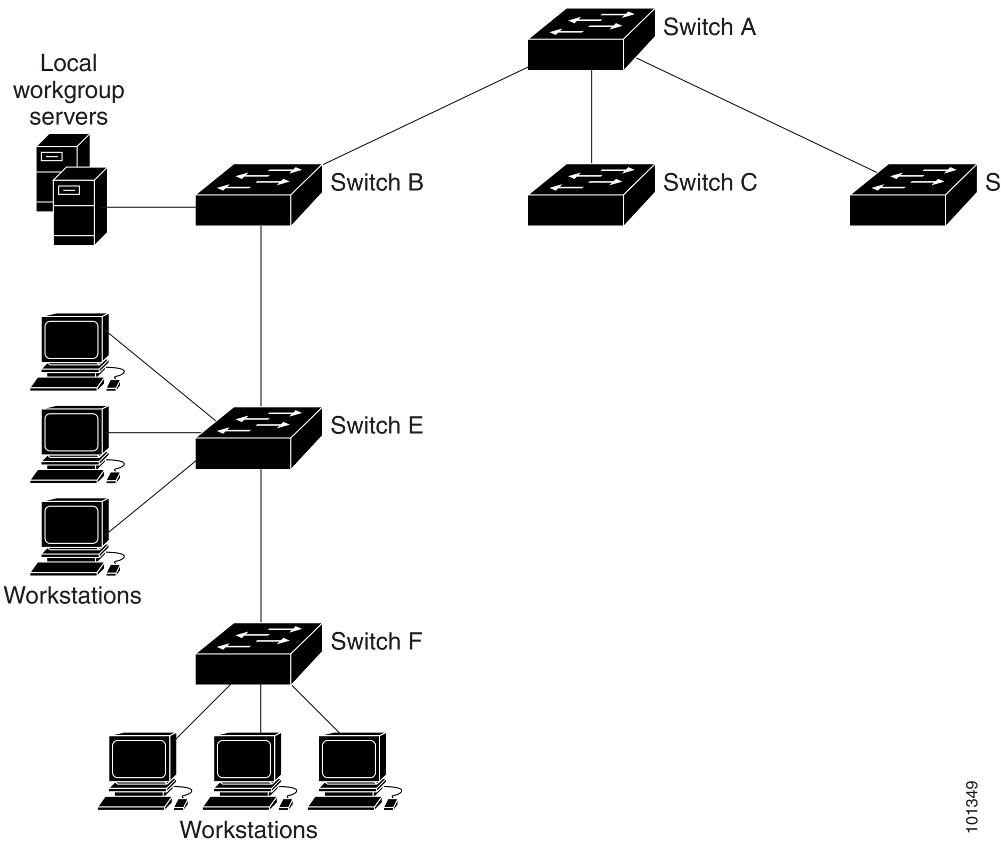
The communications between devices running NTP (known as *associations*) are usually statically configured; each device is given the IP address of all devices with which it should associate. Accurate timekeeping is possible by exchanging NTP messages between each pair of devices with an association. However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each device can simply be configured to send or receive broadcast messages; however, information flow is one-way only.

The time kept on a device is a critical resource; you should use the security features of NTP to avoid the accidental or malicious setting of an incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

Cisco's implementation of NTP does not support stratum 1 service; it is not possible to connect to a radio or atomic clock. We recommend that the time service for your network be derived from the public NTP servers available on the IP Internet.

Figure 4-1 shows a typical network example using NTP. Switch A is the NTP master, with Switches B, C, and D configured in NTP server mode, in server association with Switch A. Switch E is configured as an NTP peer to the upstream and downstream switches, Switch B and Switch F, respectively.

Figure 4-1 Typical NTP Network Configuration



If the network is isolated from the Internet, Cisco's implementation of NTP allows a device to act as if it is synchronized through NTP, when it is not. Other devices then synchronize to that device through NTP.

NTP time overrides the time set by any other method.

Several manufacturers include NTP software for their host systems, and a public version for systems running UNIX and its various derivatives is also available. This software allows host systems to be synchronized as well.

Configuring NTP

These sections contain this configuration information:

- Default NTP Configuration, page 4-4
- Configuring NTP Authentication, page 4-4
- Configuring NTP Associations, page 4-6
- Configuring NTP Broadcast Service, page 4-7

- Configuring NTP Access Restrictions, page 4-8
- Configuring the Source IP Address for NTP Packets, page 4-10
- Displaying the NTP Configuration, page 4-11

Default NTP Configuration

Table 4-1 shows the default NTP configuration.

Table 4-1 Default NTP Configuration

Feature	Default Setting
NTP authentication	Disabled. No authentication key is specified.
NTP peer or server associations	None configured.
NTP broadcast service	Disabled; no interface sends or receives NTP broadcast packets.
NTP access restrictions	No access control is specified.
NTP packet source IP address	The source address is set by the outgoing interface.

NTP is enabled on all interfaces by default. All interfaces receive NTP packets.

Configuring NTP Authentication

This procedure must be coordinated with the administrator of the NTP server; the information you configure in this procedure must be matched by the servers used by the switch to synchronize its time to the NTP server.

To authenticate the associations (communications between devices running NTP that provide for accurate timekeeping) with other devices for security purposes, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ntp authenticate**	Enables the NTP authentication feature, which is disabled by default.
Step 3	**ntp authentication-key** *number* **md5** *value*	Defines the authentication keys. By default, none are defined. • For *number*, specify a key number. The range is 1 to 4294967295. • **md5** specifies that message authentication support is provided by using the message digest algorithm 5 (MD5). • For *value*, enter an arbitrary string of up to eight characters for the key. The switch does not synchronize to a device unless both have one of these authentication keys, and the key number is specified by the **ntp trusted-key** *key-number* command.

	Command	Purpose
Step 4	**ntp trusted-key** *key-number*	Specifies one or more key numbers (defined in Step 3) that a peer NTP device must provide in its NTP packets for this switch to synchronize to it. By default, no trusted keys are defined. For *key-number*, specify the key defined in Step 3. This command provides protection against accidentally synchronizing the switch to a device that is not trusted.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable NTP authentication, use the **no ntp authenticate** global configuration command. To remove an authentication key, use the **no ntp authentication-key** *number* global configuration command. To disable authentication of the identity of a device, use the **no ntp trusted-key** *key-number* global configuration command.

This example shows how to configure the switch to synchronize only to devices providing authentication key 42 in the device's NTP packets:

```
Switch# configure terminal
Switch(config)# ntp authenticate
Switch(config)# ntp authentication-key 42 md5 aNiceKey
Switch(config)# ntp trusted-key 42
Switch(config)# end
Switch#
```

Configuring NTP Associations

An NTP association can be a peer association (this switch can either synchronize to the other device or allow the other device to synchronize to it), or it can be a server association (meaning that only this switch synchronizes to the other device, and not the other way around).

To form an NTP association with another device, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ntp peer** *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**] or **ntp server** *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]	Configures the switch system clock to synchronize a peer or to be synchronized by a peer (peer association). or Configures the switch system clock to be synchronized by a time server (server association). No peer or server associations are defined by default. • For *ip-address* in a peer association, specify either the IP address of the peer providing, or being provided, the clock synchronization. For a server association, specify the IP address of the time server providing the clock synchronization. • (Optional) For *number*, specify the NTP version number. The range is 1 to 3. By default, Version 3 is selected. • (Optional) For *keyid*, enter the authentication key defined with the **ntp authentication-key** global configuration command. • (Optional) For *interface*, specify the interface from which to pick the IP source address. By default, the source IP address is taken from the outgoing interface. • (Optional) Enter the **prefer** keyword to make this peer or server the preferred one that provides synchronization. This keyword reduces switching back and forth between peers and servers.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

You need to configure only one end of an association; the other device can automatically establish the association. If you are using the default NTP version (Version 3) and NTP synchronization does not occur, try using NTP Version 2. Many NTP servers on the Internet run Version 2.

To remove a peer or server association, use the **no ntp peer** *ip-address* or the **no ntp server** *ip-address* global configuration command.

This example shows how to configure the switch to synchronize its system clock with the clock of the peer at IP address 172.16.22.44 using NTP Version 2:

```
Switch# configure terminal
Switch(config)# ntp server 172.16.22.44 version 2
Switch(config)# end
Switch#
```

Configuring NTP Broadcast Service

The communications between devices running NTP (known as *associations*) are usually statically configured; each device is given the IP addresses of all devices with which it should form associations. Accurate timekeeping is possible by exchanging NTP messages between each pair of devices with an association. However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each device can simply be configured to send or receive broadcast messages. However, the information flow is one-way only.

The switch can send or receive NTP broadcast packets on an interface-by-interface basis if there is an NTP broadcast server, such as a router, broadcasting time information on the network. The switch can send NTP broadcast packets to a peer so that the peer can synchronize to it. The switch can also receive NTP broadcast packets to synchronize its own clock. This section provides procedures for both sending and receiving NTP broadcast packets.

To configure the switch to send NTP broadcast packets to peers so that they can synchronize their clock to the switch, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Specifies the interface to send NTP broadcast packets, and enter interface configuration mode.
Step 3	**ntp broadcast** [**version** *number*] [**key** *keyid*] [*destination-address*]	Enables the interface to send NTP broadcast packets to a peer. By default, this feature is disabled on all interfaces. • (Optional) For *number*, specify the NTP version number. The range is 1 to 3. If you do not specify a version, Version 3 is used. • (Optional) For *keyid*, specify the authentication key to use when sending packets to the peer. • (Optional) For *destination-address*, specify the IP address of the peer that is synchronizing its clock to this switch.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.
Step 7		Configures the connected peers to receive NTP broadcast packets as described in the next procedure.

To disable the interface from sending NTP broadcast packets, use the **no ntp broadcast** interface configuration command.

This example shows how to configure a port to send NTP Version 2 packets:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ntp broadcast version 2
Switch(config-if)# end
Switch#
```

To configure the switch to receive NTP broadcast packets from connected peers, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Specifies the interface to receive NTP broadcast packets, and enter interface configuration mode.
Step 3	**ntp broadcast client**	Enables the interface to receive NTP broadcast packets. By default, no interfaces receive NTP broadcast packets.
Step 4	**exit**	Returns to global configuration mode.
Step 5	**ntp broadcastdelay** *microseconds*	(Optional) Changes the estimated round-trip delay between the switch and the NTP broadcast server. The default is 3000 microseconds; the range is 1 to 999999.
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show running-config**	Verifies your entries.
Step 8	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable an interface from receiving NTP broadcast packets, use the **no ntp broadcast client** interface configuration command. To change the estimated round-trip delay to the default, use the **no ntp broadcastdelay** global configuration command.

This example shows how to configure a port to receive NTP broadcast packets:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ntp broadcast client
Switch(config-if)# end
Switch#
```

Configuring NTP Access Restrictions

You can control NTP access on two levels as described in these sections:

- Creating an Access Group and Assigning a Basic IP Access List, page 4-9
- Disabling NTP Services on a Specific Interface, page 4-10

Creating an Access Group and Assigning a Basic IP Access List

To control access to NTP services by using access lists, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ntp access-group** {**query-only** | **serve-onl**y | **serve** | **peer**} *access-list-number*	Creates an access group, and apply a basic IP access list. The keywords have these meanings: • **query-only**—Allows only NTP control queries. • **serve-only**—Allows only time requests. • **serve**—Allows time requests and NTP control queries, but does not allow the switch to synchronize to the remote device. • **peer**—Allows time requests and NTP control queries and allows the switch to synchronize to the remote device. For *access-list-number*, enter a standard IP access list number from 1 to 99.
Step 3	**access-list** *access-list-number* **permit** *source* [*source-wildcard*]	Creates the access list. • For *access-list-number*, enter the number specified in Step 2. • Enter the **permit** keyword to permit access if the conditions are matched. • For *source*, enter the IP address of the device that is permitted access to the switch. • (Optional) For *source-wildcard*, enter the wildcard bits to be applied to the source. **Note** When creating an access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The access group keywords are scanned in this order, from least restrictive to most restrictive:

1. **peer**—Allows time requests and NTP control queries and allows the switch to synchronize itself to a device whose address passes the access list criteria.
2. **serve**—Allows time requests and NTP control queries, but does not allow the switch to synchronize itself to a device whose address passes the access list criteria.
3. **serve-only**—Allows only time requests from a device whose address passes the access list criteria.
4. **query-only**—Allows only NTP control queries from a device whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first type is granted. If no access groups are specified, all access types are granted to all devices. If any access groups are specified, only the specified access types are granted.

To remove access control to the switch NTP services, use the **no ntp access-group** {**query-only** | **serve-only** | **serve** | **peer**} global configuration command.

This example shows how to configure the switch to allow itself to synchronize to a peer from access list 99. However, the switch restricts access to allow only time requests from access list 42:

```
Switch# configure terminal
Switch(config)# ntp access-group peer 99
Switch(config)# ntp access-group serve-only 42
Switch(config)# access-list 99 permit 172.20.130.5
Switch(config)# access list 42 permit 172.20.130.6
Switch(config)# end
Switch#
```

Disabling NTP Services on a Specific Interface

NTP services are enabled on all interfaces by default.

To disable NTP packets from being received on an interface, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specify the interface to disable.
Step 3	**ntp disable**	Disables NTP packets from being received on the interface. By default, all interfaces receive NTP packets. To re-enable receipt of NTP packets on an interface, use the **no ntp disable** interface configuration command
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring the Source IP Address for NTP Packets

When the switch sends an NTP packet, the source IP address is normally set to the address of the interface through which the NTP packet is sent. Use the **ntp source** global configuration command when you want to use a particular source IP address for all NTP packets. The address is taken from the specified interface. This command is useful if the address on an interface cannot be used as the destination for reply packets.

To configure a specific interface from which the IP source address is to be taken, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ntp source** *type number*	Specifies the interface type and number from which the IP source address is taken. By default, the source address is set by the outgoing interface.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The specified interface is used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** keyword in the **ntp peer** or **ntp server** global configuration command as described in the "Configuring NTP Associations" section on page 4-6.

Displaying the NTP Configuration

You can use two privileged EXEC commands to display NTP information:

- **show ntp associations** [**detail**]
- **show ntp status**

For detailed information about the fields in these displays, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.3*.

Configuring Time and Date Manually

If no other source of time is available, you can manually configure the time and date after the system is restarted. The time remains accurate until the next system restart. We recommend that you use manual configuration only as a last resort. If you have an outside source to which the switch can synchronize, you do not need to manually set the system clock.

These sections contain this configuration information:

- Setting the System Clock, page 4-11
- Displaying the Time and Date Configuration, page 4-12
- Configuring the Time Zone, page 4-12
- Configuring Summer Time (Daylight Saving Time), page 4-13

Setting the System Clock

If you have an outside source on the network that provides time services, such as an NTP server, you do not need to manually set the system clock.

To set the system clock, perform this task:

	Command	Purpose
Step 1	**clock set** *hh***:***mm***:***ss day month year* or **clock set** *hh***:***mm***:***ss month day year*	Manually sets the system clock using one of these formats. • For *hh***:***mm***:***ss*, specify the time in hours (24-hour format), minutes, and seconds. The time specified is relative to the configured time zone. • For *day*, specify the day by date in the month. • For *month*, specify the month by name. • For *year*, specify the year (no abbreviation).

This example shows how to manually set the system clock to 1:32 p.m. on July 23, 2001:

```
Switch# clock set 13:32:00 23 July 2001
```

Displaying the Time and Date Configuration

To display the time and date configuration, use the **show clock** [**detail**] privileged EXEC command.

The system clock keeps an *authoritative* flag that shows whether the time is authoritative (believed to be accurate). If the system clock has been set by a timing source such as NTP, the flag is set. If the time is not authoritative, it is used only for display purposes. Until the clock is authoritative and the *authoritative* flag is set, the flag prevents peers from synchronizing to the clock when the peers' time is invalid.

The symbol that precedes the **show clock** display has this meaning:

- *—Time is not authoritative.
- (blank)—Time is authoritative.
- .—Time is authoritative, but NTP is not synchronized.

Configuring the Time Zone

To manually configure the time zone, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**clock timezone** *zone hours-offset* [*minutes-offset*]	Sets the time zone. To set the time to UTC, use the **no clock timezone** global configuration command. The switch keeps internal time in universal time coordinated (UTC), so this command is used only for display purposes and when the time is manually set. • For *zone*, enter the name of the time zone to be displayed when standard time is in effect. The default is UTC. • For *hours-offset*, enter the hours offset from UTC. • (Optional) For *minutes-offset*, enter the minutes offset from UTC.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The *minutes-offset* variable in the **clock timezone** global configuration command is available for those cases where a local time zone is a percentage of an hour different from UTC. For example, the time zone for some sections of Atlantic Canada (AST) is UTC-3.5, where the 3 means 3 hours and .5 means 50 percent. In this case, the necessary command is **clock timezone AST -3 30**.

Configuring Summer Time (Daylight Saving Time)

To configure summer time (daylight saving time) in areas where it starts and ends on a particular day of the week each year, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**clock summer-time** *zone* **recurring** [*week day month hh*:*mm week day month hh*:*mm* [*offset*]]	Configures summer time to start and end on the specified days every year. Summer time is disabled by default. If you specify **clock summer-time** *zone* **recurring** without parameters, the summer time rules default to the United States rules. • For *zone*, specify the name of the time zone (for example, PDT) to be displayed when summer time is in effect. • (Optional) For *week*, specify the week of the month (1 to 5 or **last**). • (Optional) For *day*, specify the day of the week (Sunday, Monday...). • (Optional) For *month*, specify the month (January, February...). • (Optional) For *hh*:*mm*, specify the time (24-hour format) in hours and minutes. • (Optional) For *offset*, specify the number of minutes to add during summer time. The default is 60.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The first part of the **clock summer-time** global configuration command specifies when summer time begins, and the second part specifies when it ends. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to summer time. If the starting month is after the ending month, the system assumes that you are in the southern hemisphere.

This example shows how to specify that summer time starts on the first Sunday in April at 02:00 and ends on the last Sunday in October at 02:00:

```
Switch# configure terminal
Switch(config)# clock summer-time PDT recurring 1 Sunday April 2:00 last Sunday October
2:00
Switch(config)# end
Switch#
```

If summer time in your area does not follow a recurring pattern (configure the exact date and time of the next summer time events), perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**clock summer-time** *zone* **date** [*month date year hh***:***mm month date year hh***:***mm* [*offset*]] or **clock summer-time** *zone* **date** [*date month year hh***:***mm date month year hh***:***mm* [*offset*]]	Configures summer time to start on the first date and end on the second date. To disable summer time, use the **no clock summer-time** global configuration command. Summer time is disabled by default. • For *zone*, specify the name of the time zone (for example, PDT) to be displayed when summer time is in effect. • (Optional) For *week*, specify the week of the month (1 to 5 or **last**). • (Optional) For *day*, specify the day of the week (Sunday, Monday...). • (Optional) For *month*, specify the month (January, February...). • (Optional) For *hh***:***mm*, specify the time (24-hour format) in hours and minutes. • (Optional) For *offset*, specify the number of minutes to add during summer time. The default is 60.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

The first part of the **clock summer-time** global configuration command specifies when summer time begins, and the second part specifies when it ends. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to summer time. If the starting month is after the ending month, the system assumes that you are in the southern hemisphere.

To disable summer time, use the **no clock summer-time** global configuration command.

This example shows how to set summer time to start on October 12, 2000, at 02:00, and end on April 26, 2001, at 02:00:

```
Switch# configure terminal
Switch(config)# clock summer-time pdt date 12 October 2000 2:00 26 April 2001 2:00
Switch#
```

Configuring a System Name and Prompt

You configure the system name on the switch to identify it. By default, the system name and prompt are *Switch*.

If you have not configured a system prompt, the first 20 characters of the system name are used as the system prompt. A greater-than symbol [>] is appended. The prompt is updated whenever the system name changes.

For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.3* and the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.3*.

These sections contain this configuration information:

- Default System Name and Prompt Configuration, page 4-15
- Configuring a System Name, page 4-15
- Understanding DNS, page 4-15

Default System Name and Prompt Configuration

The default switch system name and prompt is *Switch*.

Configuring a System Name

To manually configure a system name, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**hostname** *name*	Manually configures a system name. The default setting is *switch*. The name must follow the rules for ARPANET hostnames. They must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, and hyphens. Names can be up to 63 characters. To return to the default hostname, use the **no hostname** global configuration command.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-confi**g	(Optional) Saves your entries in the configuration file.

When you set the system name, it is also used as the system prompt.

Understanding DNS

The DNS protocol controls the Domain Name System (DNS), a distributed database with which you can map hostnames to IP addresses. When you configure DNS on your switch, you can substitute the hostname for the IP address with all IP commands, such as **ping**, **telnet**, **connect**, and related Telnet support operations.

IP defines a hierarchical naming scheme that allows a device to be identified by its location or domain. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco Systems is a commercial organization that IP identifies by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, for example, the File Transfer Protocol (FTP) system is identified as *ftp.cisco.com*.

To keep track of domain names, IP has defined the concept of a domain name server, which holds a cache (or database) of names mapped to IP addresses. To map domain names to IP addresses, you must first identify the hostnames, specify the name server that is present on your network, and enable the DNS.

These sections contain this configuration information:

- Default DNS Configuration, page 4-16
- Setting Up DNS, page 4-16
- Displaying the DNS Configuration, page 4-17

Default DNS Configuration

Table 4-2 shows the default DNS configuration.

Table 4-2 Default DNS Configuration

Feature	Default Setting
DNS enable state	Enabled.
DNS default domain name	None configured.
DNS servers	No name server addresses are configured.

Setting Up DNS

To set up your switch to use the DNS, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip domain-name** *name*	Defines a default domain name that the software uses to complete unqualified hostnames (names without a dotted-decimal domain name). To remove a domain name, use the **no ip domain-name** *name* global configuration command. Do not include the initial period that separates an unqualified name from the domain name. At boot time, no domain name is configured; however, if the switch configuration comes from a BOOTP or Dynamic Host Configuration Protocol (DHCP) server, then the default domain name might be set by the BOOTP or DHCP server (if the servers were configured with this information).
Step 3	**ip name-server** *server-address1* [*server-address2 ... server-address6*]	Specifies the address of one or more name servers to use for name and address resolution. To remove a name server address, use the **no ip name-server** *server-address* global configuration command. You can specify up to six name servers. Separate each server address with a space. The first server specified is the primary server. The switch sends DNS queries to the primary server first. If that query fails, the backup servers are queried.

	Command	Purpose
Step 4	**ip domain-lookup**	(Optional) Enables DNS-based hostname-to-address translation on your switch. This feature is enabled by default. To disable DNS on the switch, use the **no ip domain-lookup** global configuration command. If your network devices require connectivity with devices in networks for which you do not control name assignment, you can dynamically assign device names that uniquely identify your devices by using the global Internet naming scheme (DNS).
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

If you use the switch IP address as its hostname, the IP address is used and no DNS query occurs. If you configure a hostname that contains no periods (.), a period followed by the default domain name is appended to the hostname before the DNS query is made to map the name to an IP address. The default domain name is the value set by the **ip domain-name** global configuration command. If there is a period (.) in the hostname, the Cisco IOS software looks up the IP address without appending any default domain name to the hostname.

Displaying the DNS Configuration

To display the DNS configuration information, use the **show running-config** privileged EXEC command.

Creating a Banner

You can configure a message-of-the-day (MOTD) and a login banner. The MOTD banner displays on all connected terminals at login and is useful for sending messages that affect all network users (such as impending system shutdowns).

The login banner also displays on all connected terminals. It appears after the MOTD banner and before the login prompts.

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.3*.

These sections contain this configuration information:

- Default Banner Configuration, page 4-18
- Configuring a Message-of-the-Day Login Banner, page 4-18
- Configuring a Login Banner, page 4-19

Default Banner Configuration

The MOTD and login banners are not configured.

Configuring a Message-of-the-Day Login Banner

You can create a single or multiline message banner that appears on the screen when someone logs in to the switch.

To configure a MOTD login banner, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**banner motd** *c message c*	Specifies the message of the day. To delete the MOTD banner, use the **no banner motd** global configuration command. For *c*, enter the delimiting character of your choice, for example, a pound sign (#), and press the **Return** key. The delimiting character signifies the beginning and end of the banner text. Characters after the ending delimiter are discarded. For *message*, enter a banner message up to 255 characters. You cannot use the delimiting character in the message.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to configure a MOTD banner for the switch by using the pound sign (#) symbol as the beginning and ending delimiter:

```
Switch(config)# banner motd #
This is a secure site. Only authorized users are allowed.
For access, contact technical support.
#
Switch(config)#
```

This example shows the banner that appears from the previous configuration:

```
Unix> telnet 172.2.5.4
Trying 172.2.5.4...
Connected to 172.2.5.4.
Escape character is '^]'.

This is a secure site. Only authorized users are allowed.
For access, contact technical support.

User Access Verification

Password:
```

Configuring a Login Banner

You can configure a login banner to be displayed on all connected terminals. This banner appears after the MOTD banner and before the login prompt.

To configure a login banner, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**banner login** *c message c*	Specifies the login message. To delete the login banner, use the **no banner login** global configuration command. For *c*, enter the delimiting character of your choice, for example, a pound sign (#), and press the **Return** key. The delimiting character signifies the beginning and end of the banner text. Characters after the ending delimiter are discarded. For *message*, enter a login message up to 255 characters. You cannot use the delimiting character in the message.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to configure a login banner for the switch by using the dollar sign ($) symbol as the beginning and ending delimiter:

```
Switch# configuration terminal
Switch(config)# banner login $
Access for authorized users only. Please enter your username and password.
$
Switch(config)# end
Switch#
```

Managing the MAC Address Table

The MAC address table contains address information that the switch uses to forward traffic between ports. All MAC addresses in the address table are associated with one or more ports. The address table includes these types of addresses:

- Dynamic address: a source MAC address that the switch learns and then ages when it is not in use.
- Static address: a manually entered unicast address that does not age and that is not lost when the switch resets.

The address table lists the destination MAC address, the associated VLAN ID, and port number associated with the address and the type (static or dynamic).

Note For complete syntax and usage information for the commands used in this section, see the command reference for this release.

These sections contain this configuration information:

- Building the Address Table, page 4-20
- MAC Addresses and VLANs, page 4-20
- Default MAC Address Table Configuration, page 4-21
- Changing the Address Aging Time, page 4-21
- Removing Dynamic Address Entries, page 4-22
- Configuring MAC Change Notification Traps, page 4-22
- Configuring MAC Move Notification Traps, page 4-24
- Configuring MAC Threshold Notification Traps, page 4-26
- Adding and Removing Static Address Entries, page 4-27
- Configuring Unicast MAC Address Filtering, page 4-28
- Displaying Address Table Entries, page 4-30

Building the Address Table

With multiple MAC addresses supported on all ports, you can connect any port on the switch to individual workstations, repeaters, switches, routers, or other network devices. The switch provides dynamic addressing by learning the source address of packets it receives on each port and adding the address and its associated port number to the address table. As stations are added or removed from the network, the switch updates the address table, adding new dynamic addresses and aging out those that are not in use.

The aging interval is globally configured. However, the switch maintains an address table for each VLAN, and STP can accelerate the aging interval on a per-VLAN basis.

The switch sends packets between any combination of ports, based on the destination address of the received packet. Using the MAC address table, the switch forwards the packet only to the port associated with the destination address. If the destination address is on the port that sent the packet, the packet is filtered and not forwarded. The switch always uses the store-and-forward method: complete packets are stored and checked for errors before transmission.

MAC Addresses and VLANs

All addresses are associated with a VLAN. An address can exist in more than one VLAN and have different destinations in each. Unicast addresses, for example, could be forwarded to port 1 in VLAN 1 and ports 9, 10, and 1 in VLAN 5.

Each VLAN maintains its own logical address table. A known address in one VLAN is unknown in another until it is learned or statically associated with a port in the other VLAN.

When private VLANs are configured, address learning depends on the type of MAC address:

- Dynamic MAC addresses learned in one VLAN of a private VLAN are replicated in the associated VLANs. For example, a MAC address learned in a private-VLAN secondary VLAN is replicated in the primary VLAN.
- Static MAC addresses configured in a primary or secondary VLAN are not replicated in the associated VLANs. When you configure a static MAC address in a private VLAN primary or secondary VLAN, you should also configure the same static MAC address in all associated VLANs.

For more information about private VLANs, see Chapter 38, “Configuring Private VLANs.”

Default MAC Address Table Configuration

Table 4-3 shows the default MAC address table configuration.

Table 4-3 Default MAC Address Table Configuration

Feature	Default Setting
Aging time	300 seconds
Dynamic addresses	Automatically learned
Static addresses	None configured

Changing the Address Aging Time

Dynamic addresses are source MAC addresses that the switch learns and then ages when they are not in use. You can change the aging time setting for all VLANs or for a specified VLAN.

Setting too short an aging time can cause addresses to be prematurely removed from the table. Then when the switch receives a packet for an unknown destination, it floods the packet to all ports in the same VLAN as the receiving port. This unnecessary flooding can impact performance. Setting too long an aging time can cause the address table to be filled with unused addresses, which prevents new addresses from being learned. Flooding results, which can impact switch performance.

To configure the dynamic address table aging time, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**mac address-table aging-time** [**0** | *10-1000000*] [**vlan** *vlan-id*]	Sets the length of time that a dynamic entry remains in the MAC address table after the entry is used or updated. To return to the default value, use the **no mac address-table aging-time** global configuration command. The range is 10 to 1000000 seconds. The default is 300. You can also enter 0, which disables aging. Static address entries are never aged or removed from the table. For *vlan-id*, valid IDs are 1 to 4094.
Step 3	**end**	Returns to privileged EXEC mode.

	Command	Purpose
Step 4	**show mac address-table aging-time**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Removing Dynamic Address Entries

To remove all dynamic entries, use the **clear mac address-table dynamic** command in EXEC mode. You can also remove a specific MAC address (**clear mac address-table dynamic address** *mac-address*), remove all addresses on the specified physical port or port channel (**clear mac address-table dynamic interface** *interface-id*), or remove all addresses on a specified VLAN (**clear mac address-table dynamic vlan** *vlan-id*).

To verify that dynamic entries have been removed, use the **show mac address-table dynamic** privileged EXEC command.

Configuring MAC Change Notification Traps

MAC change notification enables you to track users on a network by storing the MAC change activity on the switch. Whenever the switch learns or removes a MAC address, an SNMP notification can be generated and sent to the network management system. If you have many users entering and exiting the network, you can set a trap interval time to bundle the notification traps and reduce network traffic. The MAC notification history table stores the MAC address activity for each hardware port for which the trap is enabled. MAC address notifications are generated for dynamic and static MAC addresses; events are not generated for self addresses or multicast addresses.

To send MAC change notification traps to an NMS host, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**snmp-server host** *host-addr* [**traps** | **informs**] {**version** {**1** | **2c** | **3**}} [auth | noauth | priv] *community-string* ***[udp-port** port] [notification-type]*	Specifies the recipient of the trap message. • For *host-addr*, specify the name or address of the NMS. • Specify **traps** (the default) to send SNMP traps to the host. Specify **informs** to send SNMP informs to the host. • Specify the SNMP version to support. Version 1, the default, is not available with informs. • For *community-string,* specify the string to send with the notification operation. Though you can set this string by using the **snmp-server host** command, we recommend that you define this string by using the **snmp-server community** command before using the **snmp-server host** command. • For *notification-type*, use the **mac-notification** keyword.

	Command	Purpose
Step 3	**snmp-server enable traps mac-notification change**	Enables the switch to send MAC change traps to the NMS. To disable the switch from sending MAC change notification traps, use the **no snmp-server enable traps mac-notification change** global configuration command.
Step 4	**mac address-table notification change**	Enables the MAC address change notification feature.
Step 5	**mac address-table notification change** [**interval** *value*] | [**history-size** *value*]	Enters the trap interval time and the history table size. • (Optional) For **interval** *value,* specify the notification trap interval in seconds between each set of traps that are generated to the NMS. The range is 0 to 2147483647 seconds; the default is 1 second. • (Optional) For **history-size** *value*, specify the maximum number of entries in the MAC notification history table. The range is 0 to 500; the default is 1. To disable the MAC change notification feature, use the **no mac address-table notification change** global configuration command.
Step 6	**interface** *interface-id*	Enters interface configuration mode, and specify the interface on which to enable the SNMP MAC change notification trap.
Step 7	**snmp trap mac-notification change** {**added** | **removed**}	Enables the MAC change notification trap. • Enable the MAC change notification trap whenever a MAC address is **added** on this interface. • Enable the MAC change notification trap whenever a MAC address is **removed** from this interface. To disable the MAC change notification traps on a specific interface, use the **no snmp trap mac-notification change** {**added** | **removed**} interface configuration command.
Step 8	**end**	Returns to privileged EXEC mode.
Step 9	**show mac address-table notification change interface** **show running-config**	Verifies your entries.
Step 10	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to specify 172.69.59.93 as the network management system, enable the switch to send MAC change notification traps to the network management system, enable the MAC change notification feature, set the interval time to 60 seconds, set the history-size to 100 entries, and enable traps whenever a MAC address is added on the specified port.

```
Switch# configure terminal
Switch(config)# snmp-server host 172.69.59.93 private mac-notification
Switch(config)# snmp-server enable traps mac-notification change
Switch(config)# mac address-table notification change
Switch(config)# mac address-table notification change interval 60
Switch(config)# mac address-table notification change history-size 100
Switch(config)# interface fastethernet0/2
Switch(config-if)# snmp trap mac-notification change added
Switch(config-if)# end
Switch# show mac address-table notification change interface
MAC Notification Feature is Enabled on the switch
MAC Notification Flags For All Ethernet Interfaces :
----------------------------------------------------
Interface            MAC Added Trap MAC Removed Trap
---------            -------------- ----------------
GigabitEthernet1/1   Enabled        Enabled
GigabitEthernet1/2   Enabled        Enabled
GigabitEthernet1/3   Enabled        Enabled
GigabitEthernet1/4   Enabled        Enabled
GigabitEthernet1/5   Enabled        Enabled
GigabitEthernet1/6   Enabled        Enabled
GigabitEthernet1/7   Enabled        Enabled
GigabitEthernet1/8   Enabled        Enabled
GigabitEthernet1/9   Enabled        Enabled
GigabitEthernet1/10  Enabled        Enabled
GigabitEthernet1/11  Enabled        Enabled
GigabitEthernet1/12  Enabled        Enabled


Switch#
```

Configuring MAC Move Notification Traps

When you configure MAC move notification, an SNMP notification is generated and sent to the network management system whenever a MAC address moves from one port to another within the same VLAN.

To configure MAC move notification, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**snmp-server host** *host-addr* [**traps** | **informs**] {**version** {**1** | **2c** | **3**}} [auth | noauth | priv] *community-string [**udp-port** port] [notification-type]*	Specifies the recipient of the trap message. • For *host-addr*, specify the name or address of the NMS. • Specify **traps** (the default) to send SNMP traps to the host. Specify **informs** to send SNMP informs to the host. • Specify the SNMP version to support. Version 1, the default, is not available with informs. • For *community-string,* specify the string to send with the notification operation. Though you can set this string by using the **snmp-server host** command, we recommend that you define this string by using the **snmp-server community** command before using the **snmp-server host** command. • For *notification-type*, use the **mac-notification** keyword.
Step 3	**snmp-server enable traps mac-notification move**	Enables the switch to send MAC move notification traps to the NMS. To disable the switch from sending MAC notification traps, use the **no snmp-server enable traps mac-notification move** global configuration command.
Step 4	**mac address-table notification mac-move**	Enables the MAC-move notification feature. To disable this feature, use the **no mac-address-table notification mac-move** global configuration command.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show mac address-table notification mac-move** **show running-config**	Displays the MAC-move notification status.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to specify 172.69.59.93 as the network management system, enable the switch to send MAC move notification traps to the NMS, enable the MAC move notification feature, and enable traps whenever a MAC address moves from one port to another.

```
Switch# configure terminal
Switch(config)# snmp-server host 171.69.59.93 private mac-notification
Switch(config)# snmp-server enable traps mac-notification move
Switch(config)# mac address-table notification mac-move
Switch(config)# end
Switch# show mac address-table notification mac-move
MAC Move Notification: Enabled
```

Configuring MAC Threshold Notification Traps

When you configure MAC threshold notification, an SNMP notification is generated and sent to the network management system when a MAC address table (MAT) threshold limit is reached or exceeded.

To configure MAC address threshold notification, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**snmp-server host** *host-addr* [**traps** | **informs**] {**version** {**1** | **2c** | **3**}} [auth | noauth | priv] *community-string [**udp-port** port] [**notification-type**]*	Specifies the recipient of the trap message. • For *host-addr*, specify the name or address of the NMS. • Specify **traps** (the default) to send SNMP traps to the host. Specify **informs** to send SNMP informs to the host. • Specify the SNMP version to support. Version 1, the default, is not available with informs. • For *community-string,* specify the string to send with the notification operation. Though you can set this string by using the **snmp-server host** command, we recommend that you define this string by using the **snmp-server community** command before using the **snmp-server host** command. • For *notification-type*, use the **mac-notification** keyword.
Step 3	**snmp-server enable traps mac-notification threshold**	Enables the switch to send MAC threshold notification traps to the NMS. To disable the switch from sending MAC threshold notification traps, use the **no snmp-server enable traps mac-notification threshold** global configuration command.
Step 4	**mac address-table notification threshold**	Enables the MAC address threshold notification feature. To disable this feature, use the **no address-table notification threshold** global configuration command.
Step 5	**mac address-table notification threshold** [**limit** *percentage*] | [**interval** *time*]	Enters the threshold value for the MAT usage monitoring. • (Optional) For **limit** *percentage,* specify the percentage of the MAT utilization; valid values are from 1 to 100 percent. Default is 50 per cent. • (Optional) For **interval** *time,* specify the time between notifications; valid values are greater than or equal to 120 seconds. Default is 120 seconds.

	Command	Purpose
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show mac address-table notification threshold** **show running-config**	Displays the MAT utilization threshold notification status.
Step 8	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to specify 172.69.59.93 as the network management system, enable the MAC threshold notification feature, enable the switch to send MAC threshold notification traps to the NMS, set the interval to 123 seconds, and set the limit to 78 per cent.

```
Switch# configure terminal
Switch(config)# snmp-server host 171.69.59.93 private mac-notification
Switch(config)# snmp-server enable traps mac-notification threshold
Switch(config)# mac address-table notification threshold
Switch(config)# mac address-table notification threshold interval 123
Switch(config)# mac address-table notification threshold limit 78
Switch(config)# end
Switch# show mac-address-table notification threshold
     Status       limit        Interval
-------------+-----------+-------------
   enabled        78           123
Switch#
```

Adding and Removing Static Address Entries

A static address has these characteristics:

- It is manually entered in the address table and must be manually removed.
- It can be a unicast or multicast address.
- It does not age and is retained when the switch restarts.

You can add and remove static addresses and define the forwarding behavior for them. The forwarding behavior defines how a port that receives a packet forwards it to another port for transmission. Because all ports are associated with at least one VLAN, the switch acquires the VLAN ID for the address from the ports that you specify. You can specify a different list of destination ports for each source port.

A packet with a static address that arrives on a VLAN where it has not been statically entered is flooded to all ports and not learned.

You add a static address to the address table by specifying the destination MAC unicast address and the VLAN from which it is received. Packets received with this destination address are forwarded to the interface specified with the *interface-id* option.

When you configure a static MAC address in a private-VLAN primary or secondary VLAN, you should also configure the same static MAC address in all associated VLANs. Static MAC addresses configured in a private-VLAN primary or secondary VLAN are not replicated in the associated VLAN. For more information about private VLANs, see Chapter 38, “Configuring Private VLANs.”

To add a static address, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**mac address-table static** *mac-addr* **vlan** *vlan-id* **interface** *interface-id*	Adds a static address to the MAC address table. • For *mac-addr*, specify the destination MAC unicast address to add to the address table. Packets with this destination address received in the specified VLAN are forwarded to the specified interface. • For *vlan-id*, specify the VLAN for which the packet with the specified MAC address is received. Valid VLAN IDs are 1 to 4094. • For *interface-id*, specify the interface to which the received packet is forwarded. Valid interfaces include physical ports or port channels. For static multicast addresses, you can enter multiple interface IDs. For static unicast addresses, you can enter only one interface at a time, but you can enter the command multiple times with the same MAC address and VLAN ID. To remove static entries from the address table, use the **no mac address-table static** *mac-addr* **vlan** *vlan-id* [**interface** *interface-id*] global configuration command.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show mac address-table static**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to add the static address c2f3.220a.12f4 to the MAC address table. When a packet is received in VLAN 4 with this MAC address as its destination address, the packet is forwarded to the specified port:

```
Switch# configure terminal
Switch(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface
gigabitethernet0/1
Switch(config)# end
Switch#
```

Configuring Unicast MAC Address Filtering

Note Unicast MAC Address Filtering is *not* supported on Supervisor Engine 6-E.

When unicast MAC address filtering is enabled, the switch drops packets with specific source or destination MAC addresses. This feature is disabled by default and only supports unicast static addresses.

Follow these guidelines when using this feature:

- Multicast MAC addresses, broadcast MAC addresses, and router MAC addresses are not supported. If you specify one of these addresses when entering the **mac address-table static** *mac-addr* **vlan** *vlan-id* **drop** global configuration command, one of these messages appears:

  ```
  % Only unicast addresses can be configured to be dropped
  ```

```
% CPU destined address cannot be configured as drop address
```

- Packets that are forwarded to the CPU are also not supported.
- If you add a unicast MAC address as a static address and configure unicast MAC address filtering, the switch either adds the MAC address as a static address or drops packets with that MAC address, depending on which command was entered last. The second command that you entered overrides the first command.

  For example, if you enter the **mac address-table static** *mac-addr* **vlan** *vlan-id* **interface** global configuration command followed by the **mac address-table static** *mac-addr* **vlan** *vlan-id* **drop** command, the switch drops packets with the specified MAC address as a source or destination.

  If you enter the **mac address-table static** *mac-addr* **vlan** *vlan-id* **drop** global configuration command followed by the **mac address-table static** *mac-addr* **vlan** *vlan-id* **interface** command, the switch adds the MAC address as a static address.

You enable unicast MAC address filtering and configure the switch to drop packets with a specific address by specifying the source or destination unicast MAC address and the VLAN from which it is received.

To configure the switch to drop a source or destination unicast static address, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**mac address-table static** *mac-addr* **vlan** *vlan-id* **drop**	Enables unicast MAC address filtering and configure the switch to drop a packet with the specified source or destination unicast static address. • For *mac-addr*, specify a source or destination unicast MAC address. Packets with this MAC address are dropped. • For *vlan-id*, specify the VLAN for which the packet with the specified MAC address is received. Valid VLAN IDs are 1 to 4094. To disable unicast MAC address filtering, use the **no mac address-table static** *mac-addr* **vlan** *vlan-id* global configuration command.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show mac address-table static**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to enable unicast MAC address filtering and to configure the switch to drop packets that have a source or destination address of c2f3.220a.12f4. When a packet is received in VLAN 4 with this MAC address as its source or destination, the packet is dropped:

```
Switch# configure terminal
Switch(config)# mac address-table static c2f3.220a.12f4 vlan 4 drop
Switch(config)# end
Switch#
```

Displaying Address Table Entries

You can display the MAC address table by using one or more of the privileged EXEC commands described in Table 4-4:

Table 4-4 Commands for Displaying the MAC Address Table

Command	Description
show ip igmp snooping groups	Displays the Layer 2 multicast entries for all VLANs or the specified VLAN.
show mac address-table address	Displays MAC address table information for the specified MAC address.
show mac address-table aging-time	Displays the aging time in all VLANs or the specified VLAN.
show mac address-table count	Displays the number of addresses present in all VLANs or the specified VLAN.
show mac address-table dynamic	Displays only dynamic MAC address table entries.
show mac address-table interface	Displays the MAC address table information for the specified interface.
show mac address-table notification	Displays the MAC notification parameters and history table.
show mac address-table static	Displays only static MAC address table entries.
show mac address-table vlan	Displays the MAC address table information for the specified VLAN.

Managing the ARP Table

To communicate with a device (over Ethernet, for example), the software first must learn the 48-bit MAC address or the local data link address of that device. The process of learning the local data link address from an IP address is called *address resolution*.

The Address Resolution Protocol (ARP) associates a host IP address with the corresponding media or MAC addresses and the VLAN ID. Using an IP address, ARP finds the associated MAC address. When a MAC address is found, the IP-MAC address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP). By default, standard Ethernet-style ARP encapsulation (represented by the **arpa** keyword) is enabled on the IP interface.

ARP entries added manually to the table do not age and must be manually removed.

For CLI procedures, see the Cisco IOS Release 12.3 documentation on Cisco.com.

Configuring Embedded CiscoView Support

The Catalyst 4500 series switch supports CiscoView web-based administration through the Catalyst Web Interface (CWI) tool. CiscoView is a device management application that can be embedded on the switch flash and provides dynamic status, monitoring, and configuration information for your switch. CiscoView displays a physical view of your switch chassis with color-coded modules and ports and monitoring capabilities that display the switch status, performance, and other statistics. Configuration capabilities allow comprehensive changes to devices, if the required security privileges have been granted. The configuration and monitoring capabilities for the Catalyst 4500 series of switches mirror those available in CiscoView in all server-based CiscoWorks solutions, including CiscoWorks LAN Management Solution (LMS) and CiscoWorks Routed WAN Management Solution (RWAN).

These sections describe the Embedded CiscoView support available with Cisco IOS Release 12.1(20)EW and later releases:

- Understanding Embedded CiscoView, page 4-31
- Installing and Configuring Embedded CiscoView, page 4-31
- Displaying Embedded CiscoView Information, page 4-33

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 at http://www.cisco.com/cgi-bin/tablebuild.pl/cview-cat4000

Installing and Configuring Embedded CiscoView

To install and configure Embedded CiscoView, perform this task:

	Command	Purpose
Step 1	`Router# dir device_name`	Displays the contents of the device. If you are installing Embedded CiscoView for the first time, or if the CiscoView directory is empty, skip to Step 5.
Step 2	`Switch# delete device_name:cv/*`	Removes existing files from the CiscoView directory.
Step 3	`Switch# squeeze device_name:`	Recovers the space in the file system.
Step 4	`Switch# copy tftp bootflash`	Copies the tar file to bootflash.
Step 5	`Switch# 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 6	`Switch# dir device_name:`	Displays the contents of the device. In a redundant configuration, repeat Step 1 through Step 6 for the file system on the redundant supervisor engine.
Step 7	`Switch# configure terminal`	Enters global configuration mode.
Step 8	`Switch(config)# ip http server`	Enables the HTTP web server.
Step 9	`Switch(config)# snmp-server community string ro`	Configures the SNMP password for read-only operation.
Step 10	`Switch(config)# snmp-server community string rw`	Configures the SNMP password for read/write operation.

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

The following example shows how to install and configure Embedded CiscoView on your switch:

```
Switch# dir
Directory of bootflash:/
Directory of bootflash:/
    1  -rw-     9572396  Dec 30 2002 01:05:01 +00:00  cat4000-i9k2s-mz.121-19.EW
    2  -rw-     9604192   Jan 3 2003 07:46:49 +00:00  cat4000-i5k2s-mz.121-19.EW
    3  -rw-     1985024  Jan 21 2003 03:31:20 +00:00  Cat4000IOS.v4-0.tar
    4  -rw-     1910127  Jan 23 2003 04:23:39 +00:00  cv/Cat4000IOS-4.0.sgz
    5  -rw-        7258  Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_ace.html
    6  -rw-         405  Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_error.html
    7  -rw-        2738  Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_install.html
    8  -rw-       20450  Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_jks.jar
    9  -rw-       20743  Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_nos.jar
   10  -rw-       12383  Jan 23 2003 04:23:46 +00:00  cv/applet.html
   11  -rw-         529  Jan 23 2003 04:23:46 +00:00  cv/cisco.x509
   12  -rw-        2523  Jan 23 2003 04:23:46 +00:00  cv/identitydb.obj
   13  -rw-        1173  Mar 19 2003 05:50:26 +00:00  post-2003.03.19.05.50.07-passed.txt

32578556 bytes total (38199688 bytes free)
Switch#
Switch# del cv/*
Delete filename [cv/*]?
Delete bootflash:cv/Cat4000IOS-4.0.sgz? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_ace.html? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_error.html? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_install.html? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_jks.jar? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_nos.jar? [confirm]y
Delete bootflash:cv/applet.html? [confirm]y
Delete bootflash:cv/cisco.x509? [confirm]y
Delete bootflash:cv/identitydb.obj? [confirm]y
Switch#

Switch# squeeze bootflash:
All deleted files will be removed. Continue? [confirm]y
Squeeze operation may take a while. Continue? [confirm]y
Squeeze of bootflash complete
Switch#
Switch# copy tftp bootflash
Address or name of remote host []? 10.5.5.5
Source filename []? Cat4000IOS.v5-1.tar
Destination filename [Cat4000IOS.v5-1.tar]?
Accessing tftp://10.5.5.5/Cat4000IOS.v5-1.tar...
Loading Cat4000IOS.v5-1.tar from 10.5.5.5 (via FastEthernet2/1):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 2031616 bytes]

2031616 bytes copied in 11.388 secs (178400 bytes/sec)
Switch#
Switch# dir
Directory of bootflash:/

Directory of bootflash:/
    1  -rw-     9572396  Dec 30 2002 01:05:01 +00:00  cat4000-i9k2s-mz.121-19.EW
    2  -rw-     9604192   Jan 3 2003 07:46:49 +00:00  cat4000-i5k2s-mz.121-19.EW
    3  -rw-     1985024  Jan 21 2003 03:31:20 +00:00  Cat4000IOS.v4-0.tar
    4  -rw-        1173  Mar 19 2003 05:50:26 +00:00  post-2003.03.19.05.50.07-passed.txt
    5  -rw-     2031616  Mar 26 2003 05:33:12 +00:00  Cat4000IOS.v5-1.tar

32578556 bytes total (38199688 bytes free)

Switch#
Switch# archive tar /xtract Cat4000IOS.v5-1.tar /cv
extracting Cat4000IOS-5.1.sgz (1956591 bytes)
extracting Cat4000IOS-5.1_ace.html (7263 bytes)
extracting Cat4000IOS-5.1_error.html (410 bytes)
extracting Cat4000IOS-5.1_install.html (2743 bytes)
extracting Cat4000IOS-5.1_jks.jar (20450 bytes)
extracting Cat4000IOS-5.1_nos.jar (20782 bytes)
extracting applet.html (12388 bytes)
extracting cisco.x509 (529 bytes)
extracting identitydb.obj (2523 bytes)
Switch#
Switch# dir

Directory of bootflash:/
    1  -rw-     9572396  Dec 30 2002 01:05:01 +00:00  cat4000-i9k2s-mz.121-19.EW
    2  -rw-     9604192   Jan 3 2003 07:46:49 +00:00  cat4000-i5k2s-mz.121-19.EW
    3  -rw-     1985024  Jan 21 2003 03:31:20 +00:00  Cat4000IOS.v4-0.tar
    4  -rw-        1173  Mar 19 2003 05:50:26 +00:00  post-2003.03.19.05.50.07-passed.txt
    5  -rw-     2031616  Mar 26 2003 05:33:12 +00:00  Cat4000IOS.v5-1.tar
    6  -rw-     1956591  Mar 26 2003 05:36:11 +00:00  cv/Cat4000IOS-5.1.sgz
    7  -rw-        7263  Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_ace.html
    8  -rw-         410  Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_error.html
    9  -rw-        2743  Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_install.html
   10  -rw-       20450  Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_jks.jar
   11  -rw-       20782  Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_nos.jar
   12  -rw-       12388  Mar 26 2003 05:36:19 +00:00  cv/applet.html
   13  -rw-         529  Mar 26 2003 05:36:19 +00:00  cv/cisco.x509
   14  -rw-        2523  Mar 26 2003 05:36:19 +00:00  cv/identitydb.obj

32578556 bytes total (7358284 bytes free)

Switch#
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip http server
Switch(config)# snmp-server community public ro
Switch(config)# snmp-server community public rw
Switch(config)# exit
Switch# wr
Building configuration...
Compressed configuration from 2735 bytes to 1169 bytes[OK]
Switch# show ciscoview ?
  package  ADP Package Details
  version  ADP version
  |        Output modifiers
  <
```

For more information about web access to the switch, refer to the "Using the Cisco Web Browser" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide* at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fun_c/fcprt1/fcd105.htm

Displaying Embedded CiscoView Information

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

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

The following example shows how to display the Embedded CiscoView file and version information:

```
Switch# show ciscoview package
```

```
File source:
CVFILE                          SIZE(in bytes)
------------------------------------------------
Cat4000IOS-5.1.sgz               1956591
Cat4000IOS-5.1_ace.html          7263
Cat4000IOS-5.1_error.html        410
Cat4000IOS-5.1_install.html      2743
Cat4000IOS-5.1_jks.jar           20450
Cat4000IOS-5.1_nos.jar           20782
applet.html                      12388
cisco.x509                       529
identitydb.obj                   2523

Switch# show ciscoview version
Engine Version: 5.3.4 ADP Device: Cat4000IOS ADP Version: 5.1 ADK: 49
Switch#
```

CHAPTER 5

Configuring the Cisco IOS In Service Software Upgrade Process

Note The In Service Software Upgrade Process is *not* supported on Supervisor Engine 6-E.

Operating on redundant systems, the In Service Software Upgrade (ISSU) process allows Cisco IOS software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco IOS software to be modified while packet forwarding continues. This increases network availability and reduces downtime caused by planned software upgrades. This document provides information about ISSU concepts and describes the steps taken to perform ISSU in a system.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Related Documents

Related Topic	Document Title
Performing ISSU	*Cisco IOS Software: Guide to Performing In Service Software Upgrades*
Information about Cisco Nonstop Forwarding	*Cisco Nonstop Forwarding* *http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s20/fsnsf20s.htm*
Information about stateful switchover	*Stateful Switchover* *http://www.cisco.com/univercd/cc/td/doc/product/software/ios122s/122snwft/release/122s20/fssso20s.htm*
ISSU and MPLS clients	ISSU MPLS Clients

Contents

Prerequisites for Performing ISSU

The following prerequisites apply:

- ISSU is applicable only on a redundant chassis.
- Ensure that both the active and the standby supervisor engines are available in the system and are of the same type (e.g. WS-X4516-10GE).
- The new and old Cisco IOS software images must be loaded into the file systems (bootflash or compact flash) of both the active and the standby supervisor engines before you begin the ISSU process. The old image should be available either in bootflash or compact flash and the system should have been booted from one of these locations because the boot variable should not be changed before the ISSU process unfolds.
- Stateful Switchover (SSO) must be configured and the standby supervisor engine should be in STANDBY HOT state.

  Several commands tell you whether SSO is enabled: **show module**, **show running-config**, **show redundancy state**.

  This example shows how to use the **show redundancy state** command to display information about the redundancy facility state:

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

```
Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State              = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
  Communications = Up

   client count = 39
 client_notification_TMR = 240000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 18
           RF debug mask = 0x0

Switch#
```

If you do not have SSO enabled, see the *Stateful Switchover* document for further information on how to enable and configure SSO.

- Nonstop Forwarding (NSF) must be configured and working properly. If you do not have NSF enabled, see the *Cisco Nonstop Forwarding* document for further information on how to enable and configure NSF.

Restrictions for Performing ISSU

The following restrictions apply:

- Before you perform ISSU, ensure the system is configured for redundancy mode SSO and that the file system for both the active and the standby supervisor engines contains the new ISSU-compatible image. The current IOS version running in the system must also support ISSU.

  You can issue various commands on the Catalyst 4500 series switch to determine supervisor engine versioning and IOS compatibility. Alternatively, you can use the ISSU application on Cisco Feature Navigator are to determine this.
- Do not make any hardware changes while performing an ISSU process.
- ISSU is available in Cisco IOS 12.2(31)SGA and later releases.

Note All linecards are supported.

Information About Performing ISSU

Before you perform ISSU, you should understand the following concepts:

- Stateful Switchover Overview, page 5-4
- NSF Overview, page 5-6
- ISSU Process Overview, page 5-7
- Versioning Capability in Cisco IOS Software to Support ISSU, page 5-12
- SNMP Support for ISSU, page 5-13
- Compatibility Verification Using Cisco Feature Navigator, page 5-14

Stateful Switchover Overview

Development of the SSO feature is an incremental step within an overall program to improve the availability of networks constructed with Cisco IOS switches.

In specific Cisco networking devices that support dual supervisor engines, SSO takes advantage of supervisor engine redundancy to increase network availability. SSO achieves this by establishing one of the supervisor engines as the active processor while the other supervisor engine is designated as the standby processor. Following an initial synchronization between the two supervisor engines, SSO dynamically synchronizes supervisor engine state information between them in real-time.

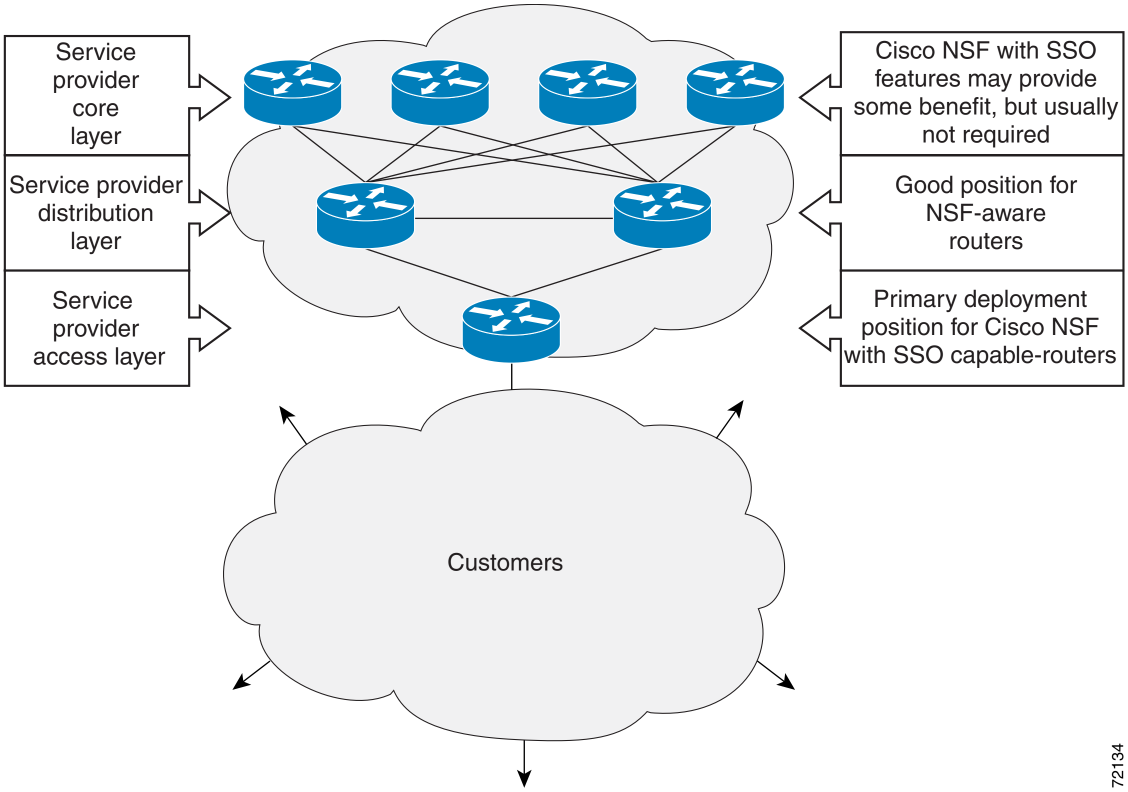
A switchover from the active to the standby processor occurs when the active supervisor engine fails or is removed from the networking device.

Cisco NSF is used with SSO. Cisco NSF allows the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With Cisco NSF, peer networking devices do not experience routing flaps, thereby reducing loss of service outages for customers.

Figure 5-1 illustrates how SSO is typically deployed in service provider networks. In this example, Cisco NSF with SSO is enabled at the access layer (edge) of the service provider network. A fault at this point could result in loss of service for enterprise customers requiring access to the service provider network.

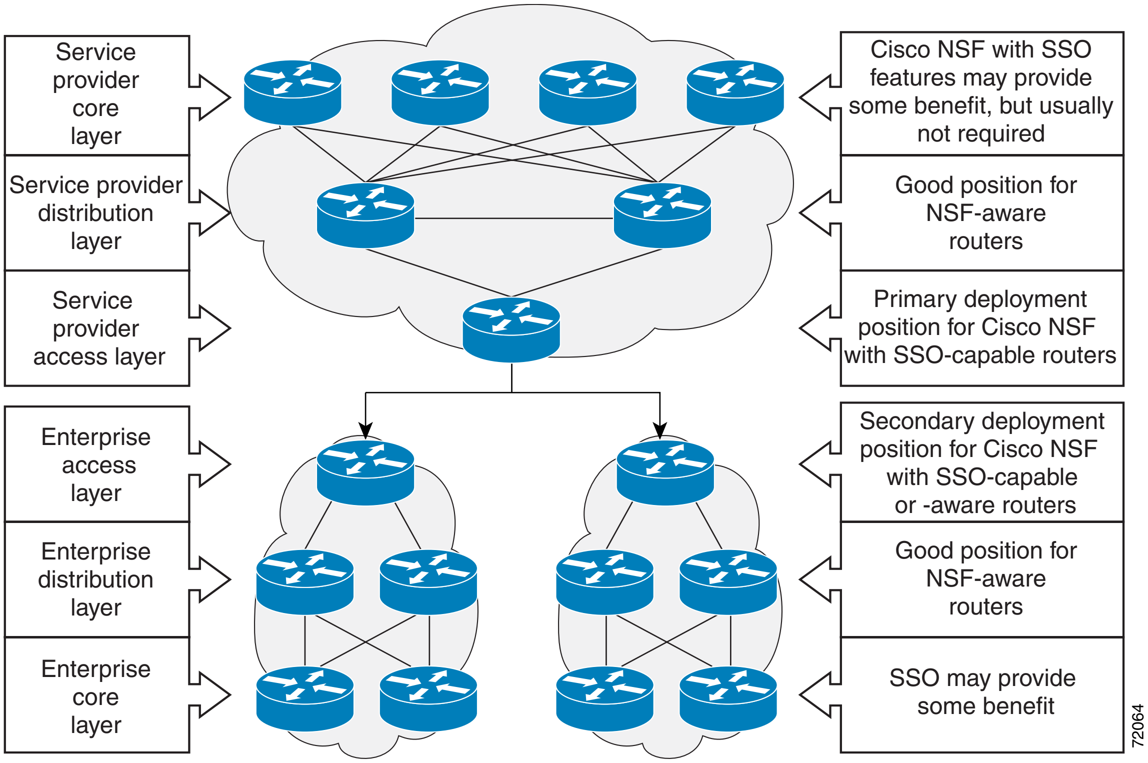
For Cisco NSF protocols that require neighboring devices to participate in Cisco NSF, Cisco NSF-aware software images must be installed on those neighboring distribution layer devices. Depending on your objectives, you may decide to deploy Cisco NSF and SSO features at the core layer of your network. Doing this can help reduce the time to restore network capacity and service for certain failures, which leads to additional availability.

Figure 5-1 Cisco NSF with SSO Network Deployment: Service Provider Networks


Service provider core layer
Service provider distribution layer
Service provider access layer
Cisco NSF with SSO features may provide some benefit, but usually not required
Good position for NSF-aware routers
Primary deployment position for Cisco NSF with SSO capable-routers
Customers
72134

Additional levels of availability may be gained by deploying Cisco NSF with SSO at other points in the network where a single point of failure exists. Figure 5-2 illustrates an optional deployment strategy that applies Cisco NSF with SSO at the enterprise network access layer. In this example, each access point in the enterprise network represents another single point of failure in the network design. In the event of a switchover or a planned software upgrade, enterprise customer sessions would continue uninterrupted through the network in this example.

Figure 5-2 Cisco NSF with SSO Network Deployment: Enterprise Networks


Service provider core layer
Service provider distribution layer
Service provider access layer
Enterprise access layer
Enterprise distribution layer
Enterprise core layer
Cisco NSF with SSO features may provide some benefit, but usually not required
Good position for NSF-aware routers
Primary deployment position for Cisco NSF with SSO-capable routers
Secondary deployment position for Cisco NSF with SSO-capable or -aware routers
Good position for NSF-aware routers
SSO may provide some benefit
72064

For further information on SSO, see the *Stateful Switchover* document.

NSF Overview

Cisco NSF works with the SSO feature in Cisco IOS software. SSO is a prerequisite of Cisco NSF. NSF works with SSO to minimize the amount of time a network is unavailable to its users following a switchover. The main objective of Cisco NSF is to continue forwarding IP packets following a supervisor engine switchover.

Usually, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in what is called a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Cisco NSF helps to suppress routing flaps in SSO-enabled devices, thus reducing network instability.

Cisco NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With Cisco NSF, peer networking devices do not experience routing flaps. Data traffic is forwarded while the standby supervisor engine assumes control from the failed active supervisor engine during a switchover. The ability of physical links to remain up through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active supervisor engine is key to Cisco NSF operation.

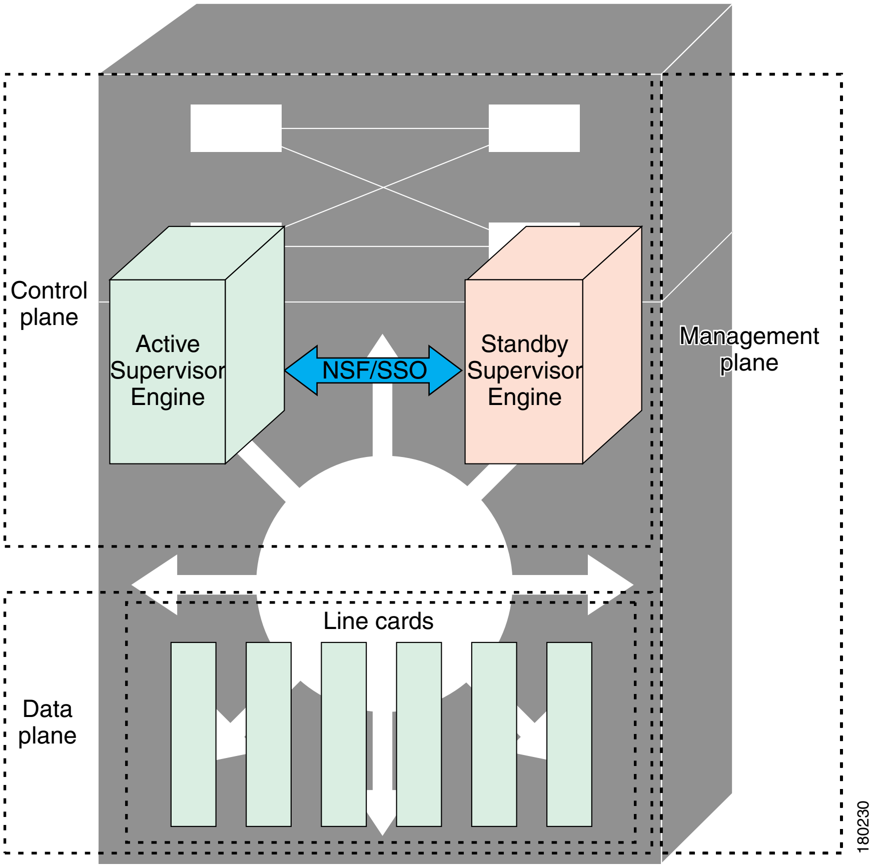
ISSU Process Overview

The ISSU process allows you to perform a Cisco IOS software upgrade or downgrade while the system continues to forward packets. Cisco IOS ISSU takes advantage of the Cisco IOS high availability infrastructure—Cisco NSF with SSO and hardware redundancy—and eliminates downtime associated with software upgrades or version changes by allowing changes while the system remains in service (see Figure 5-3).

SSO and NSF mode support configuration and runtime state synchronization from the active to the standby supervisor engine. For this, the images on both the active and the standby supervisor engines must be the same. When images on active and standby supervisor engines are different ISSU allows the two supervisor engines to be kept in synchronization even when these two versions of IOS support different sets of features and commands.

Figure 5-3 High Availability Features and Hardware Redundancy in the ISSU Process

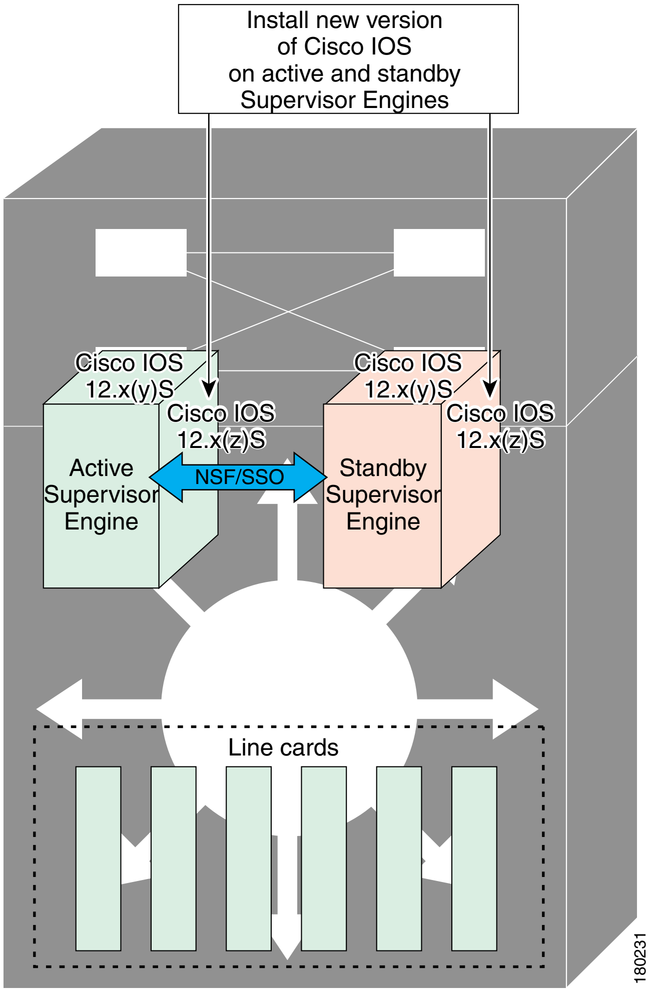
I



An ISSU-capable switch consists of two supervisor engines (active and standby) and one or more line cards. Before initiating the ISSU process, copy the Cisco IOS software into the file systems of both supervisor engines (see Figure 5-4).

Note In the following figure, Cisco IOS 12.x(y)S represents the *current* version of IOS.

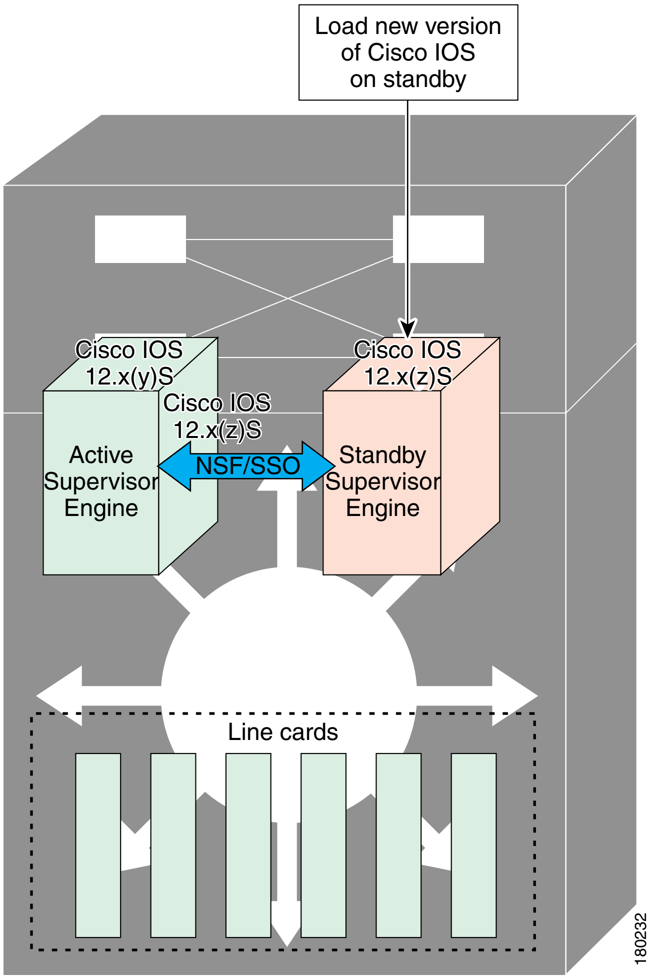
Figure 5-4 Install/Copy New Version of Cisco IOS Software on Both Supervisor Engines



After you have copied the Cisco IOS software to both file systems, load the new version of Cisco IOS software onto the standby supervisor engine (see Figure 5-5).

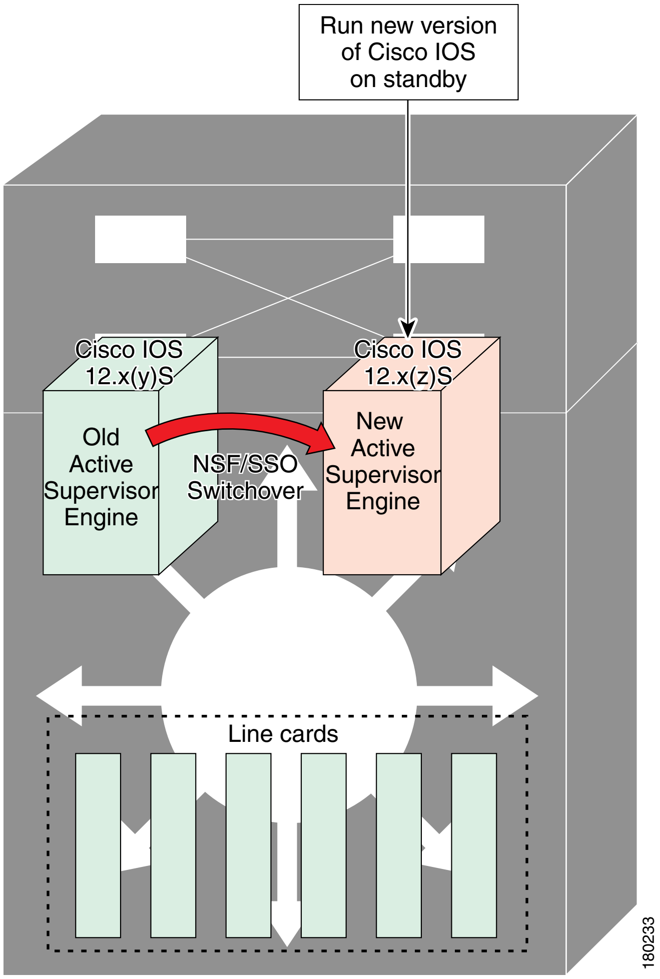
Note Without the ISSU feature, you cannot have SSO/NSF functioning between the active and standby supervisor engines when they are running two different versions of IOS image.

Figure 5-5 Load New Version of Cisco IOS Software on the Standby Supervisor Engine



After a switchover (NSF/SSO, not RPR), the standby supervisor engine takes over as the new active supervisor engine (see Figure 5-6).

Figure 5-6 Switch Over to Standby Supervisor Engine


Run new version of Cisco IOS on standby
Cisco IOS 12.x(y)S
Old Active Supervisor Engine
NSF/SSO Switchover
Cisco IOS 12.x(z)S
New Active Supervisor Engine
Line cards
180233

The former active supervisor engine is loaded with old IOS image so that if the new active supervisor engine experiences problems, you can abort and conduct a switchover to the former active, which is already running the old image. Next, the former active supervisor engine is loaded with the new version of Cisco IOS software and becomes the new standby supervisor engine (see Figure 5-7).

Figure 5-7 Load New Standby Supervisor Engine with New Cisco IOS Software

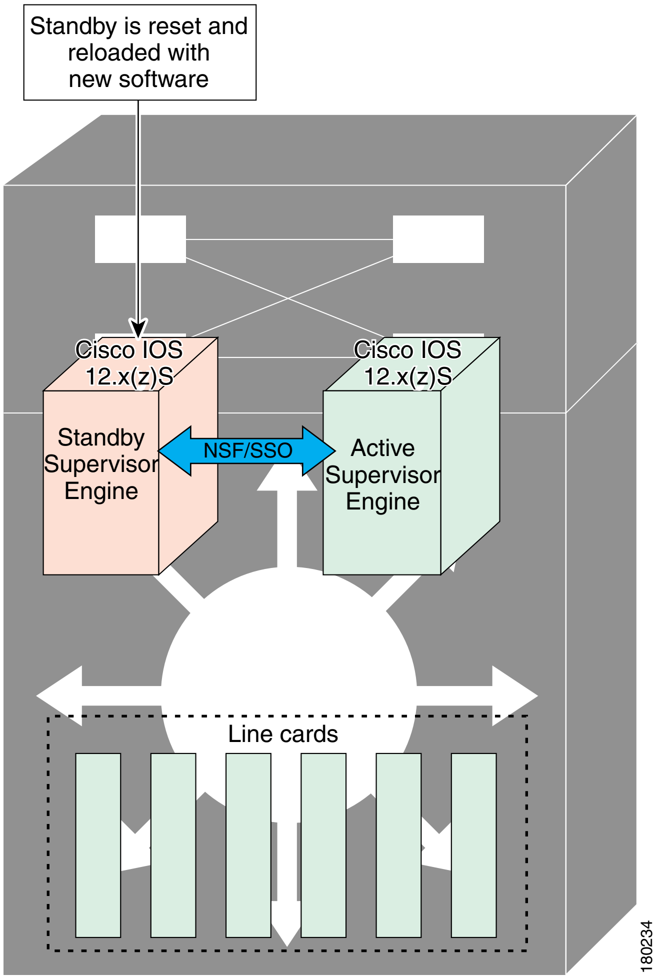
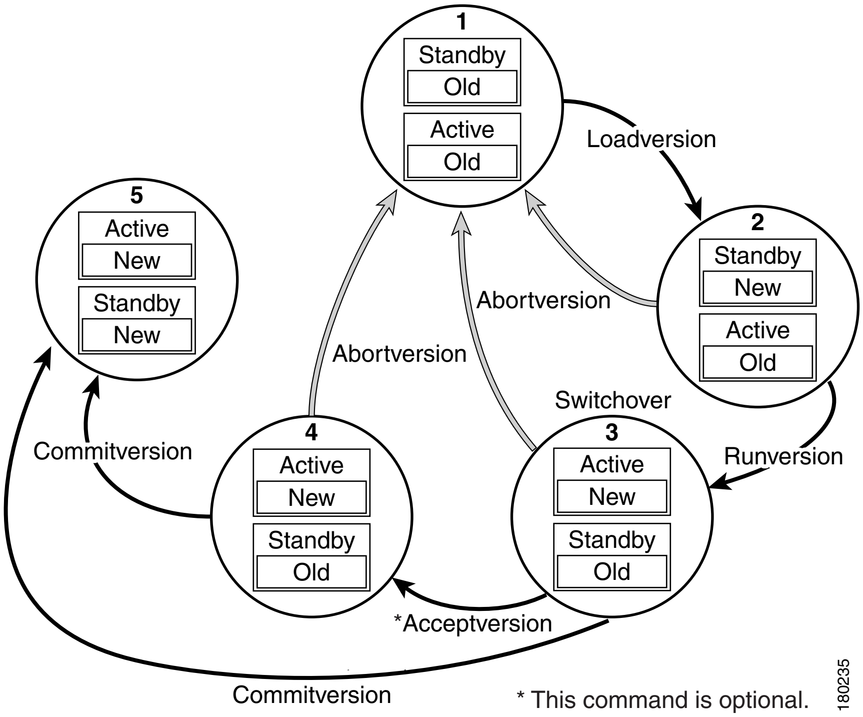


Figure 5-8 shows the steps during the ISSU process.

Figure 5-8 Steps During the ISSU Process



Versioning Capability in Cisco IOS Software to Support ISSU

Before the introduction of ISSU, the SSO mode of operation required each supervisor engine to be running the same versions of Cisco IOS software.

Note The operating mode of the system in a redundant HA configuration is determined by exchanging version strings when the standby supervisor engine registers with the active supervisor engine.

The system entered SSO mode only if the versions running on the both supervisor engines were the same. If not, the redundancy mode will be changed to RPR. With ISSU capability, the implementation allows two different but compatible release levels of Cisco IOS images to interoperate in SSO mode and enables software upgrades while packet forwarding continues. Version checking done before ISSU capability was introduced is no longer sufficient to allow the system to determine the operating mode.

ISSU requires additional information to determine compatibility between software versions. Therefore, a compatibility matrix is defined that contains information about other images with respect to the one in question. This compatibility matrix represents the compatibility of two software versions, one running on the active and the other on the standby supervisor engine, and to allow the system to determine the highest operating mode it can achieve. Incompatible versions will not be able to progress to SSO operational mode.

The Cisco IOS infrastructure has been internally modified and redesigned to accommodate subsystem versioning with ISSU. Cisco IOS subsystems correspond to feature sets and software component groupings. Features or subsystems that maintain state information across supervisor engines are HA-aware or SSO clients. A mechanism called ISSU Framework, or ISSU protocol, allows subsystems within Cisco IOS software to communicate between the active and the standby supervisor engines and to negotiate the message version for communication between supervisor engines. Internally, all NSF- and SSO-compliant applications or subsystems that are HA-aware must follow this protocol to establish communication with their peer across different versions of software. (For further information on operating modes, see the *Stateful Switchover* document.)

Compatibility Matrix

You can perform the ISSU process when the Cisco IOS software on both the active and the standby supervisor engine is capable of ISSU and the old and new images are compatible. The compatibility matrix information stores the compatibility among releases as follows:

- Compatible—The base-level system infrastructure and all optional HA-aware subsystems are compatible. An in-service upgrade or downgrade between these versions will succeed with minimal service impact. The matrix entry designates the images to be compatible (C).
- Base-level compatible—One or more of the optional HA-aware subsystems is not compatible. An in-service upgrade or downgrade between these versions will succeed; however, some subsystems will not be able to maintain state always during the transition from the old to the new version of IOS. The matrix entry designates the images to be base-level compatible (B).
- Incompatible—A core set of system infrastructure exists in IOS that must be able to interoperate in a stateful manner for SSO to function correctly. If any of these required features or subsystems is not interoperable, then the two versions of the Cisco IOS software images are declared to be incompatible. An in-service upgrade or downgrade between these versions is not possible. The matrix entry designates the images to be incompatible (I). The system operates in RPR mode during the period when the versions of IOS at the active and standby supervisor engines are incompatible.

  If you attempt to perform ISSU with a peer that does not support ISSU, the system automatically uses RPR instead.

The compatibility matrix represents the compatibility relationship a Cisco IOS software image has with all of the other Cisco IOS software versions within the designated support window (for example, all of those software versions the image “knows” about) and is populated and released with every image. The matrix stores compatibility information between its own release and prior releases. It is always the newest release that contains the latest information about compatibility with existing releases in the field. The compatibility matrix is available within the Cisco IOS software image and on Cisco.com so that users can determine in advance whether an upgrade can be done using the ISSU process.

To display the compatibility matrix data between two software versions on a given system, enter the **show issu comp-matrix stored** command.

Note This command is useful *only for verification purpose*s because it is available *only after* the ISSU process has started. You might want to check the compatibility matrix prior to starting ISSU. Use the Feature Navigator to obtain the needed information

http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp

SNMP Support for ISSU

SNMP for SSO provides a mechanism for synchronizing the SNMP configurations and the MIBs that support SSO from the active supervisor engine to the standby supervisor engine, assuming that both supervisor engines are running the same version of Cisco IOS software. This assumption is not valid for ISSU.

With ISSU, an SNMP client can handle transformations for the MIBs across two different versions of Cisco IOS, if needed. An SNMP client handles transformation for all MIBs and handles the transmit and receive functionality across the active and standby supervisor engines. During SNMP, a MIB is completely synchronized from the active supervisor engine to the standby supervisor engine only if the versions of the MIB on both Cisco IOS releases are the same.

Compatibility Verification Using Cisco Feature Navigator

The ISSU application on Cisco Feature Navigator allows you to:

- Select an ISSU-capable image
- Identify which images are compatible with that image
- Compare two images and understand the compatibility level of the images (that is, compatible, base-level compatible, and incompatible)
- Compare two images and see the client compatibility for each ISSU client
- Provide links to release notes for the image

How to Perform the ISSU Process

Unlike SSO, which is a mode of operation for the device and a prerequisite for performing ISSU, the ISSU process is a series of steps performed while the switch is in operation. The steps result in an upgrade to a new or modified Cisco IOS software, and have a minimal impact to traffic.

Be aware of the following restrictions while performing the ISSU process:

- Even with ISSU, it is recommended that upgrades be performed during a maintenance window.
- The new features should not be enabled (if they require change of configuration) during the ISSU process.
- In a downgrade scenario, if any feature is not available in the downgrade revision of Cisco IOS software image, that feature should be disabled prior to initiating the ISSU process.

This section includes the following topics:

- Verifying the ISSU Software Installation, page 5-14
- Loading New Cisco IOS Software on the Standby Supervisor Engine, page 5-17 (required)
- Switching to the Standby Supervisor Engine, page 5-20 (required)
- Stopping the ISSU Rollback Timer (Optional), page 5-23 (optional)
- Loading New Cisco IOS Software on the New Standby Supervisor Engine, page 5-24
- Aborting a Software Upgrade During ISSU, page 5-26
- Configuring the Rollback Timer to Safeguard Against Upgrade Issues, page 5-27
- Displaying ISSU Compatibility Matrix Information, page 5-29

Verifying the ISSU Software Installation

During the ISSU process, there are five valid states: init, load version, run version, and system reset. Use the **show issu state** command to obtain the current ISSU state:

- Disabled state—The state for the standby supervisor engine while this engine is resetting.
- Init state—The initial state is two supervisor engines, one active and one standby, before the ISSU process is started. It is also the final state after the ISSU process completes.
- Load version (LV) state—The standby supervisor engine is loaded with the new version of Cisco IOS software.
- Run version (RV) state—The **issu runversion** command forces the switchover of the supervisor engines. The newly active supervisor engine now runs the new Cisco IOS software image.
- System reset (SR) state—This state occurs either when you issue the **issu abortversion** command before the Init state is reached, or if the rollback timer expires before you execute the **issu acceptversion** command.

You can verify the ISSU software installation by entering **show** commands to provide information on the state of the during the ISSU process.

SUMMARY STEPS

1. **enable**
2. **show issu state** [**detail**]
3. **show redundancy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Switch# show issu state [detail]`	Displays the state of the during the ISSU process.
Step 3	`Switch# show redundancy`	Displays current or historical status, mode, and related redundancy information about the device.

This example shows how to display the state and the current status of the supervisor engine during the ISSU process:

```
Switch> enable
Switch# show issu state
Switch# show redundancy
```

Verifying Redundancy Mode Before Beginning the ISSU Process

Before you begin the ISSU process, verify the redundancy mode for the system and be sure to configure NSF and SSO.

The following example displays verification that the system is in SSO mode, that slot 1 is the active supervisor engine, and that slot 2 is the standby supervisor engine. Both supervisor engines are running the same Cisco IOS software image.

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

```
Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State              = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
  Communications = Up

   client count = 39
 client_notification_TMR = 240000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 18
           RF debug mask = 0x0


Switch# show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 1 minute
Switchovers system experienced = 0
              Standby failures = 0
        Last switchover reason = none

                 Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
     Operating Redundancy Mode = Stateful Switchover
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 1
        Current Software state = ACTIVE
       Uptime in current state = 0 minutes
                 Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
                          BOOT = bootflash:old_image,1;
        Configuration register = 0x822

Peer Processor Information :
----------------------------
              Standby Location = slot 2
        Current Software state = STANDBY HOT
       Uptime in current state = 1 minute
                 Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
                          BOOT = bootflash:old_image,1;
        Configuration register = 0x822
```

Verifying the ISSU State Before Beginning the ISSU Process

Ensure that the active and standby supervisor engines are up and in ISSU Init state and that the boot variables are set and pointing to valid files.

The following example displays the ISSU state before the process begins:

```
Switch# show issu state detail
                          Slot = 1
                      RP State = Active
                    ISSU State = Init
                 Boot Variable = bootflash:old_image,1;
                Operating Mode = Stateful Switchover
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = bootflash:old_image

                          Slot = 2
                      RP State = Standby
                    ISSU State = Init
                 Boot Variable = bootflash:old_image,1;
                Operating Mode = Stateful Switchover
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = bootflash:old_image
```

The new version of the Cisco IOS software must be present on both of the supervisor engines. The directory information displayed for each of the supervisor engines (or supervisor engines) shows that the new version is present.

```
Switch# dir bootflash:
Directory of bootflash:/

    5  -rwx    13636500   Sep 6 2006 09:32:33 +00:00  old_image
    6  -rwx    13636500   Sep 6 2006 09:34:07 +00:00  new_image

61341696 bytes total (1111388 bytes free)

Switch# dir slavebootflash:
Directory of slavebootflash:/

    4  -rwx    13636500   Sep 6 2006 09:40:10 +00:00  old_image
    5  -rwx    13636500   Sep 6 2006 09:42:13 +00:00  new_image

61341696 bytes total (1116224 bytes free)
```

Loading New Cisco IOS Software on the Standby Supervisor Engine

This task describes how to use ISSU to load a new version of Cisco IOS software to the standby supervisor engine.

Prerequisites

- Ensure that the new version of Cisco IOS software image is already present in the file system of both the active and standby supervisor engines. Also ensure that appropriate boot parameters (BOOT string and config-register) are set for the standby supervisor engine.
- Optionally, perform additional tests and commands to determine the current state of peers and interfaces for later comparison.
- Ensure the system (both active and standby supervisor engines) is in SSO redundancy mode. If the system is in RPR mode rather than SSO mode, you can still upgrade the system using the ISSU CLI commands, but the system will experience extended packet loss during the upgrade.'

Refer to the *Stateful Switchover* document for more details on how to configure SSO mode on supervisor engines.

- For ISSU to function, the image names on the active and standby supervisor engines must match.

Perform the following steps at the active supervisor engine.

SUMMARY STEPS

1. **enable**
2. **issu loadversion** *active-slot active-image-new standby-slot standby-image-new* [**forced**]
3. **show issu state** [**detail**]
4. **show redundancy**[**states**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Switch# issu loadversion active-slot active-image-new standby-slot standby-image-new [forced]`	Starts the ISSU process and (optionally) overrides the automatic rollback when the new Cisco IOS software version is detected to be incompatible. It may take several seconds after the **issu loadversion** command is entered for Cisco IOS software to load onto the standby supervisor engine and for the standby supervisor engine to transition to SSO mode. This causes the standby supervisor engine to reload with the new image. If you use the **forced** option, the standby supervisor engine is booted with the new image. After the image is loaded on the standby supervisor engine, if the image is incompatible, the system is forced to the RPR mode. Otherwise the system will continue in the SSO mode.
Step 3	`Switch# show issu state [detail]`	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode. It may take several seconds after entering the **issu loadversion** command for Cisco IOS software to load onto the standby supervisor engine and the standby supervisor engine to transition to SSO mode. If you enter the **show issu state** command too quickly, you may not see the information you need.
Step 4	`Switch# show redundancy [states]`	Displays redundancy facility state information.

This example shows how to start the ISSU process, boot the standby supervisor engine in the Standby Hot state, and load the standby supervisor engine slot (2) with the new image:

```
Switch> enable
Switch# issu loadversion 1 bootflash:new_image 2 slavebootflash:new_image
Switch# show issu state detail
                          Slot = 1
                      RP State = Active
```

```
                    ISSU State = Load Version
                 Boot Variable = bootflash:old_image,12
                Operating Mode = Stateful Switchover
               Primary Version = bootflash:old_image
             Secondary Version = bootflash:new_image
               Current Version = bootflash:old_image

                          Slot = 2
                      RP State = Standby
                    ISSU State = Load Version
                 Boot Variable = bootflash:new_image,12;bootflash:old_image,12
                Operating Mode = Stateful Switchover
               Primary Version = bootflash:old_image
             Secondary Version = bootflash:new_image
               Current Version = bootflash:new_image



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


Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State              = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
  Communications = Up

   client count = 39
 client_notification_TMR = 240000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

The following examples shows how the forced option places the system in RPR mode:

```
Switch> enable
Switch# issu loadversion 1 bootflash:new_image 2 slavebootflash:new_image forced
Switch# show issu state detail
                          Slot = 1
                      RP State = Active
                    ISSU State = Load Version
                 Boot Variable = bootflash:old_image,12
                Operating Mode = RPR
               Primary Version = bootflash:old_image
             Secondary Version = bootflash:new_image
               Current Version = bootflash:old_image

                          Slot = 2
                      RP State = Standby
                    ISSU State = Load Version
                 Boot Variable = bootflash:new_image,12;bootflash:old_image,12
                Operating Mode = RPR
               Primary Version = bootflash:old_image
             Secondary Version = bootflash:new_image
               Current Version = bootflash:new_image
```

The following example shows the redundancy mode as RPR:

```
Switch# show redundancy states
       my state = 13 -ACTIVE
     peer state = 4  -STANDBY COLD
           Mode = Duplex
           Unit = Primary
        Unit ID = 1


Redundancy Mode (Operational) = RPR
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State              = RPR
Maintenance Mode = Disabled
    Manual Swact = enabled
  Communications = Up

   client count = 39
 client_notification_TMR = 240000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

Switching to the Standby Supervisor Engine

This task describes how to switchover to the standby supervisor engine, which is running the new Cisco IOS software image.

Perform the following steps at the active supervisor engine.

SUMMARY STEPS

1. **enable**
2. **issu runversion** *standby-slot [standby-image-new]*
3. **show issu state** [**detail**]
4. **show redundancy**[**states**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch>` **`enable`**	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	`Switch#` **`issu runversion`** *`standby-slot [standby-image-new]`*	Forces a switchover from the active to the standby supervisor engine and reloads the former active (current standby) supervisor engines with the old image. When you enter the **issu runversion** command, an SSO switchover will be performed, and NSF procedures will be invoked if so configured.
Step 3	`Switch#` **`show issu state`** `[`**`detail`**`]`	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that a switchover occurs to slot 2.
Step 4	`Switch#` **`show redundancy`** `[`**`states`**`]`	Displays redundancy facility state information.

This example shows how to cause a switchover to the former standby supervisor engine (slot 2), reset the former active supervisor engine and reload it with the old image so it becomes the standby supervisor engine:

```
Switch> enable
Switch# issu runversion 2 slavebootflash:new_image
This command will reload the Active unit.  Proceed ? [confirm]

A switchover happens at this point. At the new active supervisor engine, do the following
after old active supervisor engine comes up as standby.

Switch# show issu state detail
                          Slot = 2
                      RP State = Active
                    ISSU State = Run Version
                 Boot Variable = bootflash:new_image,12;bootflash:old_image,12
                Operating Mode = Stateful Switchover
               Primary Version = bootflash:new_image
             Secondary Version = bootflash:old_image
               Current Version = bootflash:new_image

                          Slot = 1
                      RP State = Standby
                    ISSU State = Run Version
                 Boot Variable = bootflash:old_image,12
                Operating Mode = Stateful Switchover
               Primary Version = bootflash:new_image
             Secondary Version = bootflash:old_image
               Current Version = bootflash:old_image
```

Note The new active supervisor engine is now running the new version of software, and the standby supervisor engine is running the old version of software and is in the STANDBY-HOT state.

```
Switch# show redundancy states
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Secondary
        Unit ID = 2
```

```
Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State              = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
  Communications = Up

   client count = 39
 client_notification_TMR = 240000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
           RF debug mask = 0x0
```

Once the Runversion has completed, the new active supervisor engine will be running the new version of software and the previously active supervisor engine will now become the standby supervisor engine. The standby will be reset and reloaded, but it will remain on the previous version of software and come back online in STANDBY-HOT status. The following example shows how to verify these conditions.

```
Switch# show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 23 minutes
Switchovers system experienced = 1
              Standby failures = 0
        Last switchover reason = user forced

                 Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
     Operating Redundancy Mode = Stateful Switchover
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 2
        Current Software state = ACTIVE
       Uptime in current state = 11 minutes
                 Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
                          BOOT = bootflash:new_image,12;bootflash:old_image,12
        Configuration register = 0x822

Peer Processor Information :
----------------------------
              Standby Location = slot 1
        Current Software state = STANDBY HOT
       Uptime in current state = 4 minutes
                 Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
                          BOOT = bootflash:old_image,12
        Configuration register = 0x822
```

Stopping the ISSU Rollback Timer (Optional)

This optional task describes how to stop the rollback timer.

If you do not run the following procedure before the rollback timer "timeout," the system automatically aborts the ISSU process and reverts to the original Cisco IOS software version. By default the rollback timer is 45 minutes.

Use the following to decide what action you should take:

- You need to stop the rollback timer (then validate and run the **commitversion** command directly), if you want to retain your switch in this state for an extended period.
- You do not need to stop the roll-back timer, if you want to proceed to the following step (running "acceptversion") within the rollback timer window of 45 minutes.

Note The **issu acceptversion** command may be optionally executed after the **issu runversion** command.

SUMMARY STEPS

1. **enable**
2. **issu acceptversion** *active-slot-number [active-slot-number]*
3. **show issu state** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Switch# issu acceptversion active-slot [active-image-new]`	Halts the rollback timer and ensures the new Cisco IOS ISSU process is not automatically aborted during the ISSU process. Enter the **issu acceptversion** command within the time period specified by the rollback timer to acknowledge that the supervisor engine has achieved connectivity to the outside world; otherwise, the ISSU process is terminated, and the system reverts to the previous version of Cisco IOS software by switching to the standby supervisor engine.
Step 3	`Switch# show issu rollback-timer`	Displays the amount of time left before an automatic rollback will occur.

This example displays the Timer before you stop it. In the following example, the "Automatic Rollback Time" information indicates the amount of time remaining before an automatic rollback will occur.

```
Switch> enable
Switch# show issu rollback-timer
        Rollback Process State = In progress
      Configured Rollback Time = 45:00
       Automatic Rollback Time = 38:30

Switch# issu acceptversion 2 bootflash:new_image
% Rollback timer stopped. Please issue the commitversion command.
```

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

Loading New Cisco IOS Software on the New Standby Supervisor Engine

This task explains how to load new version of Cisco IOS software to the new standby supervisor engine.

Perform the following steps at the active supervisor engine.

SUMMARY STEPS

1. **enable**
2. **issu commitversion** *standby-slot-number [standby-image-new]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Switch# issu commitversion standby-slot-number [standby-image-new]`	Allows the new Cisco IOS software image to be loaded into the standby supervisor engine.
Step 3	`Switch# show redundancy [states]`	Displays redundancy facility state information.
Step 4	`Switch# show issu state [detail]`	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that a switchover occurs to slot 2.

This example shows how to reset and reload the current standby supervisor engine (slot 1) with the new Cisco IOS software version. After issuing the **commitversion** command, the standby supervisor engine will boot in the Standby Hot state.

```
Switch> enable
Switch# issu commitversion 1 slavebootflash:new_image

Wait till standby supervisor is reloaded with the new image. Then apply the following:

Switch# show redundancy states
00:17:12: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Secondary
        Unit ID = 2


Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State              = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
  Communications = Up

   client count = 39
 client_notification_TMR = 240000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 18
           RF debug mask = 0x0

Switch# show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 41 minutes
Switchovers system experienced = 1
              Standby failures = 1
        Last switchover reason = user forced

                 Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
     Operating Redundancy Mode = Stateful Switchover
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 2
        Current Software state = ACTIVE
       Uptime in current state = 29 minutes
                 Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
                          BOOT = bootflash:new_image,12;bootflash:old_image,1;
        Configuration register = 0x822

Peer Processor Information :
----------------------------
              Standby Location = slot 1
        Current Software state = STANDBY HOT
       Uptime in current state = 12 minutes
                 Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
                          BOOT = bootflash:new_image,12;bootflash:old_image,1;
        Configuration register = 0x822

Switch# show issu state detail
                          Slot = 2
                      RP State = Active
                    ISSU State = Init
                 Boot Variable = bootflash:new_image,12;bootflash:old_image,1;
                Operating Mode = Stateful Switchover
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = bootflash:new_image

                          Slot = 1
                      RP State = Standby
                    ISSU State = Init
                 Boot Variable = bootflash:new_image,12;bootflash:old_image,1;
                Operating Mode = Stateful Switchover
               Primary Version = N/A
             Secondary Version = N/A
               Current Version = bootflash:new_image
```

The ISSU process has been completed. At this stage, any further Cisco IOS software version upgrade or downgrade will require that a new ISSU process be invoked.

Aborting a Software Upgrade During ISSU

You can abort the ISSU process at any stage manually (prior to issuing the **issu commitversion** command) by issuing the **issu abortversion** command. The ISSU process also aborts on its own if the software detects a failure.

Note Issuing the **issu abortversion** command before the standby supervisor engine becomes hot might disrupt the traffic

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

If the process is aborted after you enter either the **issu runversion** or **issu acceptversion** command, then a second switchover is performed to the new standby supervisor engine that is still running the original software version. The supervisor engine that had been running the new software is reset and reloaded with the original software version.

Note Ensure that the standby supervisor is fully booted *before* issuing the **abortversion** command on an active sup command.

The following task describes how to abort the ISSU process before you complete the ISSU process with the **issu commitversion** command.

Perform the following task on the active supervisor engine.

SUMMARY STEPS

1. **enable**
2. **issu abortversion** *active-slot [active-image-new]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Switch# issu abortversion active slot [active-image-new]`	Cancels the ISSU upgrade or downgrade process in progress and restores the router to its state before the process had started.

This example shows how to abort the ISSU process on slot number 2, the slot for the current active supervisor engine.:

```
Switch> enable
Switch# issu abortversion 2
```

Configuring the Rollback Timer to Safeguard Against Upgrade Issues

Cisco IOS software maintains an ISSU rollback timer, to safeguard against an upgrade that may leave the new active supervisor engine in a state in which communication with the standby supervisor engine is severed.

You may want to configure the rollback timer to fewer than 45 minutes (the default) so that the user need not wait in case the new software is not committed or the connection to the switch was lost while it was in runversion mode. A user may want to configure the rollback timer to more than 45 minutes in order to have enough time to verify the operation of the new Cisco IOS software before committing the new image.

Note The valid timer value range is from 0 to 7200 seconds (two hours). A value of 0 seconds disables the rollback timer.

Once you are satisfied that the ISSU process has been successful and you want to remain in the current state, you must indicate acceptance by issuing the **issu acceptversion** command, which stops the rollback timer. Therefore, entering the **issu acceptversion** command is extremely important to moving the ISSU process forward.

Issuing the **issu commitversion** command at this stage is equal to entering both the **issu acceptversion** and the **issu commitversion** commands. Use the **issu commitversion** command if you do not intend to run in the current state for a period of time and are satisfied with the new software version.

Note The rollback timer can be configured only in the ISSU Init state.

This task explains how to configure the rollback timer.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **issu set rollback-timer** *hh::mm::ss*
4. **show issu rollback-timer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Switch# configure terminal`	Enters global configuration mode.
Step 3	`Switch(config)# issu set rollback-timer` *hh::mm::ss*	Configures the rollback timer value.
Step 4	`Switch(config)# exit`	Returns the user to privileged EXEC mode.
Step 5	`Switch# show issu rollback-timer`	Displays the current setting of the ISSU rollback timer.

This example shows how to set the rollback timer to 3600 seconds:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# issu set rollback-timer 3600
% Rollback timer value set to [ 3600 ] seconds

Switch(config)# exit

Switch# show issu rollback-timer
        Rollback Process State = Not in progress
      Configured Rollback Time = 60:00
```

The Rollback Timer cannot be set in LV state, as the following example illustrates:

```
Switch# show issu state detail
                          Slot = 1
                      RP State = Active
                    ISSU State = Load Version
                 Boot Variable = bootflash:old_image,12
                Operating Mode = RPR
               Primary Version = bootflash:old_image
             Secondary Version = bootflash:new_image
               Current Version = bootflash:old_image

                          Slot = 2
                      RP State = Standby
                    ISSU State = Load Version
                 Boot Variable = bootflash:new_image,12;bootflash:old_image,12
                Operating Mode = RPR
               Primary Version = bootflash:old_image
             Secondary Version = bootflash:new_image
               Current Version = bootflash:new_image

Switch# show issu rollback-timer
        Rollback Process State = Not in progress
      Configured Rollback Time = 60:00

Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# issu set rollback-timer 20
% ISSU state should be [ init ] to set the rollback timer
```

Displaying ISSU Compatibility Matrix Information

The ISSU compatibility matrix contains information about other software images about the version in question. This compatibility matrix represents the compatibility of the two software versions, one running on the active and the other on the standby supervisor engine, and the matrix allows the system to determine the highest operating mode it can achieve. This information helps the user identify whether or not to use ISSU.

This task shows how to display information about the ISSU compatibility matrix.

SUMMARY STEPS

1. **enable**
2. **show issu comp-matrix** {**negotiated** | **stored** | **xml**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Switch# show issu comp-matrix {negotiated | stored | xml}`	Displays information regarding the ISSU compatibility matrix. • negotiated - Displays negotiated compatibility matrix information • stored - Displays negotiated compatibility matrix information . • xml - Displays negotiated compatibility matrix information in XML format.

This example shows how to display negotiated information regarding the compatibility matrix:

```
Switch> enable
Switch# show issu comp-matrix negotiated

CardType: WS-C4507R(112), Uid: 2,  Image Ver: 12.2(31)SGA
Image Name: cat4500-ENTSERVICES-M

Cid      Eid      Sid      pSid     pUid       Compatibility
=======================================================
2        1        262151   3        1          COMPATIBLE
3        1        262160   5        1          COMPATIBLE
4        1        262163   9        1          COMPATIBLE
5        1        262186   25       1          COMPATIBLE
7        1        262156   10       1          COMPATIBLE
8        1        262148   7        1          COMPATIBLE
9        1        262155   1        1          COMPATIBLE
10       1        262158   2        1          COMPATIBLE
11       1        262172   6        1          COMPATIBLE
100      1        262166   13       1          COMPATIBLE
110      113      262159   14       1          COMPATIBLE
200      1        262167   24       1          COMPATIBLE
2002     1        -        -        -          UNAVAILABLE
2003     1        262185   23       1          COMPATIBLE
```

```
2004    1       262175  16      1         COMPATIBLE
2008    1       262147  26      1         COMPATIBLE
2008    1       262168  27      1         COMPATIBLE
2010    1       262171  32      1         COMPATIBLE
2012    1       262180  31      1         COMPATIBLE
2021    1       262170  41      1         COMPATIBLE
2022    1       262152  42      1         COMPATIBLE
2023    1       -       -       -         UNAVAILABLE
2024    1       -       -       -         UNAVAILABLE
2025    1       -       -       -         UNAVAILABLE
2026    1       -       -       -         UNAVAILABLE
2027    1       -       -       -         UNAVAILABLE
2028    1       -       -       -         UNAVAILABLE
2054    1       262169  8       1         COMPATIBLE
2058    1       262154  29      1         COMPATIBLE
2059    1       262179  30      1         COMPATIBLE
2067    1       262153  12      1         COMPATIBLE
2068    1       196638  40      1         COMPATIBLE
2070    1       262145  21      1         COMPATIBLE
2071    1       262178  11      1         COMPATIBLE
2072    1       262162  28      1         COMPATIBLE
2073    1       262177  33      1         COMPATIBLE
2077    1       262165  35      1         COMPATIBLE
2078    1       196637  34      1         COMPATIBLE
2079    1       262176  36      1         COMPATIBLE
2081    1       262150  37      1         COMPATIBLE
2082    1       262161  39      1         COMPATIBLE
2083    1       262184  20      1         COMPATIBLE
2084    1       262183  38      1         COMPATIBLE
4001    101     262181  17      1         COMPATIBLE
4002    201     262164  18      1         COMPATIBLE
4003    301     262182  19      1         COMPATIBLE
4004    401     262146  22      1         COMPATIBLE
4005    1       262149  4       1         COMPATIBLE

Message group summary:
Cid     Eid     GrpId     Sid     pSid    pUid    Nego Result
=============================================================
2       1       1         262151  3       1       Y
3       1       1         262160  5       1       Y
4       1       1         262163  9       1       Y
5       1       1         262186  25      1       Y
7       1       1         262156  10      1       Y
8       1       1         262148  7       1       Y
9       1       1         262155  1       1       Y
10      1       1         262158  2       1       Y
11      1       1         262172  6       1       Y
100     1       1         262166  13      1       Y
110     113     115       262159  14      1       Y
200     1       1         262167  24      1       Y
2002    1       2         -       -       -       N - did not negotiate
2003    1       1         262185  23      1       Y
2004    1       1         262175  16      1       Y
2008    1       1         262147  26      1       Y
2008    1       2         262168  27      1       Y
2010    1       1         262171  32      1       Y
2012    1       1         262180  31      1       Y
2021    1       1         262170  41      1       Y
2022    1       1         262152  42      1       Y
2023    1       1         -       -       -       N - did not negotiate
2024    1       1         -       -       -       N - did not negotiate
2025    1       1         -       -       -       N - did not negotiate
2026    1       1         -       -       -       N - did not negotiate
2027    1       1         -       -       -       N - did not negotiate
```

```
2028    1       1         -       -       -       N - did not negotiate
2054    1       1         262169  8       1       Y
2058    1       1         262154  29      1       Y
2059    1       1         262179  30      1       Y
2067    1       1         262153  12      1       Y
2068    1       1         196638  40      1       Y
2070    1       1         262145  21      1       Y
2071    1       1         262178  11      1       Y
2072    1       1         262162  28      1       Y
2073    1       1         262177  33      1       Y
2077    1       1         262165  35      1       Y
2078    1       1         196637  34      1       Y
2079    1       1         262176  36      1       Y
2081    1       1         262150  37      1       Y
2082    1       1         262161  39      1       Y
2083    1       1         262184  20      1       Y
2084    1       1         262183  38      1       Y
4001    101     1         262181  17      1       Y
4002    201     1         262164  18      1       Y
4003    301     1         262182  19      1       Y
4004    401     1         262146  22      1       Y
4005    1       1         262149  4       1       Y

List of Clients:
Cid       Client Name              Base/Non-Base
================================================
2         ISSU Proto client        Base
3         ISSU RF                  Base
4         ISSU CF client           Base
5         ISSU Network RF client   Base
7         ISSU CONFIG SYNC         Base
8         ISSU ifIndex sync        Base
9         ISSU IPC client          Base
10        ISSU IPC Server client   Base
11        ISSU Red Mode Client     Base
100       ISSU rfs client          Base
110       ISSU ifs client          Base
200       ISSU Event Manager clientBase
2002      CEF Push ISSU client     Base
2003      ISSU XDR client          Base
2004      ISSU SNMP client         Non-Base
2008      ISSU Tableid Client      Base
2010      ARP HA                   Base
2012      ISSU HSRP Client         Non-Base
2021      XDR Int Priority ISSU cliBase
2022      XDR Proc Priority ISSU clBase
2023      FIB HWIDB ISSU client    Base
2024      FIB IDB ISSU client      Base
2025      FIB HW subblock ISSU clieBase
2026      FIB SW subblock ISSU clieBase
2027      Adjacency ISSU client    Base
2028      FIB IPV4 ISSU client     Base
2054      ISSU process client      Base
2058      ISIS ISSU RTR client     Non-Base
2059      ISIS ISSU UPD client     Non-Base
2067      ISSU PM Client           Base
2068      ISSU PAGP_SWITCH Client  Non-Base
2070      ISSU Port Security clientNon-Base
2071      ISSU Switch VLAN client  Non-Base
2072      ISSU dot1x client        Non-Base
2073      ISSU STP                 Non-Base
2077      ISSU STP MSTP            Non-Base
2078      ISSU STP IEEE            Non-Base
2079      ISSU STP RSTP            Non-Base
```

```
2081      ISSU DHCP Snooping clientNon-Base
2082      ISSU IP Host client      Non-Base
2083      ISSU Inline Power client Non-Base
2084      ISSU IGMP Snooping clientNon-Base
4001      ISSU C4K Chassis client  Base
4002      ISSU C4K Port client     Base
4003      ISSU C4K Rkios client    Base
4004      ISSU C4K HostMan client  Base
4005      ISSU C4k GaliosRedundancyBase
```

This example shows how to display stored information regarding the compatibility matrix:

```
Switch# show issu comp-matrix stored

Number of Matrices in Table = 1

        (1) Matrix for cat4500-ENTSERVICES-M(112) - cat4500-ENTSERVICES-M(112)
        ==========================================
        Start Flag (0xDEADBABE)

                My Image ver:  12.2(905.7)HAEFT
                Peer Version    Compatability
                ------------    -------------
                12.2(31)SGA             Base(2)
                12.2(31)SGA1            Base(2)
                12.2(905.7)HAEFT        Comp(3)
```

CHAPTER 6

Configuring Interfaces

This chapter describes how to configure interfaces for the Catalyst 4500 series switches. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of Interface Configuration

By default, all interfaces are enabled. The 10/100-Mbps Ethernet interfaces autonegotiate connection speed and duplex. The 10/100/1000-Mbps Ethernet interfaces negotiate speed, duplex, and flow control. The 1000-Mbps Ethernet interfaces negotiate flow control only. Autonegotiation automatically selects the fastest speed possible on that port for the given pair. If a speed is explicitly stated for an interface, that interface will default to half duplex unless it is explicitly set for full duplex.

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

- Interface type:

  - Fast Ethernet (use the **fastethernet** keyword)
  - Gigabit Ethernet (use the **gigabitethernet** keyword)
  - 10-Gigabit Ethernet (use the **tengigabitethernet** keyword)
- Slot number—The slot in which the interface module is installed. Slots are numbered starting with 1, from top to bottom.
- Interface number—The interface number on the module. The interface numbers always begin with 1. When you are facing the front of the switch, the interfaces are numbered from left to right.

You can identify interfaces by physically checking the slot/interface location on the switch. You can also use the Cisco IOS **show** commands to display information about a specific interface or all the interfaces.

Using the interface Command

These general instructions apply to all interface configuration processes:

Step 1 At the privileged EXEC prompt, enter the **configure terminal** command to enter global configuration mode:

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

Step 2 In global configuration mode, enter the **interface** command. Identify the interface type and the number of the connector on the interface card. The following example shows how to select Fast Ethernet, slot 5, interface 1:

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

Step 3 Interface numbers are assigned at the factory at the time of installation or when modules are added to a system. Enter the **show interfaces** EXEC command to see a list of all interfaces installed on your switch. A report is provided for each interface that your switch supports, as shown in this display:

```
Switch(config-if)#Ctrl-Z
Switch#show interfaces
Vlan1 is up, line protocol is down
  Hardware is Ethernet SVI, address is 0004.dd46.7aff (bia 0004.dd46.7aff)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 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 packets output, 0 bytes, 0 underruns
     0 output errors, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
GigabitEthernet1/1 is up, line protocol is down
  Hardware is Gigabit Ethernet Port, address is 0004.dd46.7700 (bia 0004.dd46.7700)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 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
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
GigabitEthernet1/2 is up, line protocol is down
  Hardware is Gigabit Ethernet Port, address is 0004.dd46.7701 (bia 0004.dd46.7701)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 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
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
--More--
<...output truncated...>
```

Step 4 To begin configuring Fast Ethernet interface 5/5, as shown in the following example, enter the **interface** keyword, interface type, slot number, and interface number in global configuration mode:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/5
Switch(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 5 Follow each **interface** command with 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 exit interface configuration mode and return to privileged EXEC mode.

Step 6 After you configure an interface, check its status by using the EXEC **show** commands listed in the "Monitoring and Maintaining the Interface" section on page 6-22.

Configuring a Range of Interfaces

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

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

Command	Purpose
`Switch(config)# interface range {vlan vlan_ID - vlan_ID} | {{fastethernet | gigabitethernet | tengigabitethernet | macro macro_name} slot/interface - interface} [, {vlan vlan_ID - vlan_ID} {{fastethernet | gigabitethernet | tengigabitethernet | macro macro_name} slot/interface - interface}]`	Selects the range of interfaces to be configured. Note the following: • You are required to enter a space before the dash. • You can enter up to five comma-separated ranges. • You are not required to enter spaces before or after the comma.

Note When you use the **interface range** command, you must add a space between the **vlan**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**, or **macro** keyword and the dash. For example, the command **interface range fastethernet 5/1 - 5** specifies a valid range; the command **interface range fastethernet 1-5** does not contain a valid range command.

Note The **interface range** command works only with VLAN interfaces that have been configured with the **interface vlan** command (the **show running-configuration** command displays the configured VLAN interfaces). VLAN interfaces that are not displayed by the **show running-configuration** command cannot be used with the **interface range** command.

This example shows how to reenable all Fast Ethernet interfaces 5/1 to 5/5:

```
Switch(config)# interface range fastethernet 5/1 - 5
Switch(config-if-range)# no shutdown
Switch(config-if-range)#
*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
Switch(config-if)#
```

This example shows how to use a comma to add different interface type strings to the range to re-enable all Fast Ethernet interfaces ranging from 5/1 to 5/5 and both Gigabit Ethernet interfaces 1/1 and 1/2:

```
Switch(config-if)# interface range fastethernet 5/1 - 5, gigabitethernet 1/1 - 2
Switch(config-if)# no shutdown
Switch(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
Switch(config-if)#
```

If you enter multiple configuration commands while you are in interface-range configuration mode, each command is run as it is entered (they are not batched together and run after you exit interface-range configuration mode). If you exit interface-range configuration mode while the commands are being run, some commands might not be run on all interfaces in the range. Wait until the command prompt is displayed 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:

Table 6-1

Command	Purpose
`Switch(config)# `**`define interface-range`** *`macro_name`* {**`vlan`** *`vlan_ID - vlan_ID`*} | {{**`fastethernet`** | **`gigabitethernet`**} *`slot/interface`* **`-`** *`interface`*} [**`,`** {**`vlan`** *`vlan_ID - vlan_ID`*} {{**`fastethernet`** | **`gigabitethernet`**} *`slot/interface`* **`-`** *`interface`*}]	Defines the interface-range macro and saves it in the running configuration file.

This example shows how to define an interface-range macro named **enet_list** to select Fast Ethernet interfaces 5/1 through 5/4:

```
Switch(config)# define interface-range enet_list fastethernet 5/1 - 4
```

To show the defined interface-range macro configuration, perform this task:

Table 6-2

Command	Purpose
`Switch# `**`show running-config`**	Shows the defined interface-range macro configuration.

This example shows how to display the defined interface-range macro named **enet_list**:

```
Switch# show running-config | include define
define interface-range enet_list FastEthernet5/1 - 4
Switch#
```

To use an interface-range macro in the **interface range** command, perform this task:

Table 6-3

Command	Purpose
`Switch(config)# `**`interface range macro`** *`name`*	Selects the interface range to be configured using the values saved in a named interface-range macro.

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

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

Deploying 10-Gigabit Ethernet and a Gigabit Ethernet SFP Ports

Note On a Catalyst 4510R series switch, if you enable both the 10-Gigabit Ethernet and Gigabit Ethernet SFP uplink ports, you must re-boot the switch. On the Catalyst 4503, 4506, and 4507R series switches, this capability is automatically enabled.

Prior to Cisco IOS Release 12.2(25)SG, the Cisco Catalyst 4500 Supervisor Engine V-10GE allowed you to enable either the dual wire-speed 10-Gigabit Ethernet ports, or four alternatively wired Gigabit Ethernet SFP uplink ports. With Cisco IOS Release 12.2(25)SG, you can simultaneously deploy the dual 10 Gigabit Ethernet ports and the four Gigabit Ethernet SFP ports on the Catalyst 4503, Catalyst 4506, and Catalyst 4507R chassis.

When you deploy a Catalyst 4510R chassis, one of the following configurations is supported:

- Dual 10-Gigabit Ethernet ports (X2 optics) only.
- Four Gigabit Ethernet ports (SFP optics) only.
- Both the dual 10-Gigabit Ethernet and the four Gigabit Ethernet ports, with the understanding that the tenth slot (Flex-Slot) will only support a 2-port gigabit interface converter (GBIC) line card (WS-X4302-GB) when in this mode.

To select the 10-Gigabit Ethernet or the Gigabit Ethernet SFP uplink port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Establishes global configuration mode.
Step 2	`Switch(config)# hw-module uplink select [all	gigabitethernet	tengigabitethernet]`	Selects the port type to enable.

Note On a Supervisor Engine V-10GE (WS-X4516-10GE) in a 10 slot chassis (Catalyst 4510R and 4510RE), if a startup configuration with a new uplink mode is copied into flash memory and the system is power cycled, the system will not come up with the new uplink mode. After copying the startup configuration with the new uplink mode into flash memory, the uplink mode must be changed to the new uplink mode through the command interface before the system is power cycled. This ensures that the system comes up in the new uplink mode.

The following example shows how to enable both 10-Gigabit Ethernet and Gigabit Ethernet SFP uplink ports on a Catalyst 4510R series switch:

```
Switch# configure terminal
Switch(config)# hw-module uplink select all
Warning: This configuration mode will place slot 10 in flex slot mode
```

Deploying 10-Gigabit Ethernet or Gigabit Ethernet Ports on WS-X4606-10GE-E and Supervisor Engine 6-E

To increase the flexibility of X2 ports on both Supervisor Engine 6-E and WS-X4606-10GE-E, the Catalyst 4500 switch supports TwinGig Convertor modules. When you plug a TwinGig Convertor module into an X2 hole, it converts a single X2 hole (capable of holding one pluggable X2 optic) into two SFP holes (capable of holding two pluggable SFP optics). This enables you to have 10 Gigabit ports and 1-Gigabit ports on the same linecard. It also allows you to use Gigabit ports, and then switch to a 10-Gigabit port, when needed.

Topics include:

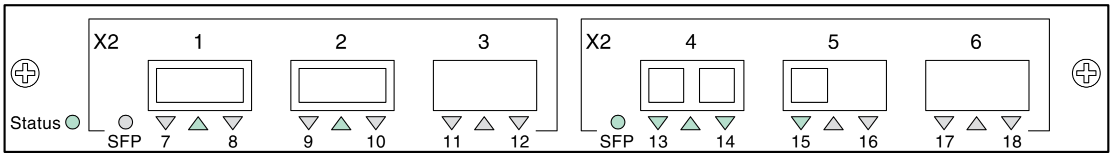
- Port Numbering TwinGig Convertors, page 6-8
- Limitations on Using a TwinGig Convertor, page 6-8

- Selecting X2/TwinGig Convertor Mode, page 6-8

Port Numbering TwinGig Convertors

When a TwinGig Convertor is enabled or disabled, the number and type of ports on the linecard change dynamically. The terminology must reflect this behavior. In Cisco IOS, 10-Gigabit ports are named *TenGigabit* and 1-Gigabit ports are named *Gigabit*. Starting with Cisco IOS Release 12.2(40)SG, to avoid having two ports named TenGigabit1/1 and Gigabit1/1, the 10-Gigabit and 1-Gigabit port numbers are independent. For example, for a WS-X4606-10GE-E module with six X2 holes, the X2 ports are named *TenGigabit slot-num/<1-6>*, and the SFP ports are named *Gigabit slot-num/<7-18>*.

Figure 6-1 Faceplate for WS-X4606-10GE



In Cisco IOS, ports 1 through 18 always exist. This means that you can apply configurations on them and they display in the CLI output. However, only the X2 or the SFP ports can be active at any particular time. For example, if an X2 is plugged into the second hole, the X2 port 2 is active and SFP ports 9 and 10 are inactive. If a TwinGig Convertor is plugged into the second hole, the X2 port 2 is inactive, and the SFP ports 9 and 10 are active. The inactive ports are treated analogously to the inactive ports on Supervisor Engines IV and V-10GE, where at no time are all of the uplinks are connected to the switching ASICs.

Note When using both TwinGig and X2 transceivers on the WS-X4606-X2-E module, keep them grouped with ports 1-3 in one group and ports 4-6 in another. Inserting a TwinGig or X2 transciever in any port will affect the capabilities of its partner ports, and all three will be set to handle the same type automatically. Mixing within a port group will not work. As an example, you would not be able to have an X2 in port 1 and a TwinGig in port 2 and expect both of them to function.

Limitations on Using a TwinGig Convertor

In a Supervisor Engine 6-E system, the ports are connected to the switching engine through a stub ASIC. This stub ASIC imposes some limitations on the ports: Gig and 10 Gig ports cannot be mixed on a single stub ASIC; they must either be all 10 Gig (X2), or all Gig (TwinGig Converter and SFP). The faceplates of X2 modules show this stub port grouping, either with actual physical grouping with a box drawn around a grouping.

Selecting X2/TwinGig Convertor Mode

The default configuration mode is X2. So, if you plan to deploy 10-Gigabit interfaces, you don’t need to configure anything. However, if you want to deploy Gigabit interfaces (that is, use TwinGig Convertors), you must configure the associated port-group:

- To determine how the X2 holes on a module are grouped, enter the **show hw-module module <*m*> port-group <*p*>** command.

  For a WS-X4606-10GE-E chassis, the output is similar to the following:

```
Switch# show hw-module module 1 port-group
Module Port-group Active                          Inactive
-------------------------------------------------------------
   1        1     Te1/1-3                         Gi1/7-12
   1        2     Te1/4-6                         Gi1/13-18

Switch# show int status mod 1

Port      Name               Status       Vlan       Duplex  Speed Type
Te1/1                        notconnect   1            full    10G 10GBase-LR
Te1/2                        connected    1            full    10G 10GBase-LR
Te1/3                        notconnect   1            full    10G No X2
Te1/4                        notconnect   1            full    10G No X2
Te1/5                        notconnect   1            full    10G No X2
Te1/6                        notconnect   1            full    10G No X2
Gi1/7                        inactive     1            full   1000 No Gbic
Gi1/8                        inactive     1            full   1000 No Gbic
Gi1/9                        inactive     1            full   1000 No Gbic
Gi1/10                       inactive     1            full   1000 No Gbic
Gi1/11                       inactive     1            full   1000 No Gbic
Gi1/12                       inactive     1            full   1000 No Gbic
Gi1/13                       inactive     1            full   1000 No Gbic
Gi1/14                       inactive     1            full   1000 No Gbic
Gi1/15                       inactive     1            full   1000 No Gbic
Gi1/16                       inactive     1            full   1000 No Gbic
Gi1/17                       inactive     1            full   1000 No Gbic
Gi1/18                       inactive     1            full   1000 No Gbic
Switch#
```

- To configure the modes of operation for each X2 port group in which you want to deploy Gigabit, enter the **hw-module module <*m*> port-group <*p*> select gigabitethernet** command. This configuration is preserved across power cycles and reloads.

  To deploy Gigabit Ethernet interfaces using the TwinGig Convertor, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Establishes global configuration mode.
Step 2	`Switch(config)# hw-module module m port-group p select [gigabitethernet	tengigabitethernet]`	Selects the mode of operation for each X2 port-group. Default is TenGigabit Ethernet (x2).
Step 3	`Switch(config)# exit`	Exits configuration mode.
Step 4	`Switch# show int status mod n`	Verifies the setting.

This example shows how to select Gigabit Ethernet interfaces on a WS-X4606-10GE-E using the TwinGig Convertor:

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# hw-module module 1 port-group 1 select gigabitethernet
Switch(config)# exit
Switch# show int status mod 1
Port      Name               Status       Vlan       Duplex  Speed Type
Te1/1                        inactive     1            full    10G No X2
Te1/2                        inactive     1            full    10G No X2
Te1/3                        inactive     1            full    10G No X2
```

```
Te1/4                       notconnect   1            full    10G No X2
Te1/5                       notconnect   1            full    10G No X2
Te1/6                       notconnect   1            full    10G No X2
Gi1/7                       notconnect   1            full   1000 No Gbic
Gi1/8                       notconnect   1            full   1000 No Gbic
Gi1/9                       notconnect   1            full   1000 No Gbic
Gi1/10                      notconnect   1            full   1000 No Gbic
Gi1/11                      notconnect   1            full   1000 No Gbic
Gi1/12                      notconnect   1            full   1000 No Gbic
Gi1/13                      inactive     1            full   1000 No Gbic
Gi1/14                      inactive     1            full   1000 No Gbic
Gi1/15                      inactive     1            full   1000 No Gbic
Gi1/16                      inactive     1            full   1000 No Gbic
Gi1/17                      inactive     1            full   1000 No Gbic
Gi1/18                      inactive     1            full   1000 No Gbic
```

Digital Optical Monitoring Transceiver Support

Command line interface (CLI) commands (show inventory, show idprom interface) are used on transceivers to obtain serial number, model name, inventory information.

The following commands are specific to the transceivers that support the DOM capability:

- Displays current values and thresholds for all sensor on a particular interface transceiver:

  ```
  show interfaces <int-name> transceiver [detail] [threshold]
  ```

- Enables or disables the *entSensorThresholdNotification* for all sensors in all the transceivers:

  ```
  snmp-server enable trap transceiver
  ```

- Enables or disables transceiver monitoring:

  ```
  transceiver type all
  ```

Note This feature is only available when a DOM capable transceiver is present and configured for monitoring. The frequency at which the sensor information is refreshed depends on default values configured in the transceiver SEEPROM (Serial Electrically Erasable Programmable Read Only Memory).

Configuring Optional Interface Features

The following subsections describe optional procedures:

- Configuring Ethernet Interface Speed and Duplex Mode, page 6-11
- Configuring Flow Control, page 6-14
- Configuring Jumbo Frame Support, page 6-16
- Interacting with Baby Giants, page 6-19
- Configuring auto-MDIX on a Port, page 6-19

Configuring Ethernet Interface Speed and Duplex Mode

Speed and Duplex Mode Configuration Guidelines

Note You do not configure the client device for autonegotiation. Rather, you configure the switch with the speed, or range of speeds, that you want to autonegotiate.

You can configure the interface speed and duplex mode parameters to **auto** and allow the Catalyst 4500 series switch to negotiate the interface speed and duplex mode between interfaces. If you decide to configure the interface **speed** and **duplex** commands manually, consider the following:

- If you enter the **no speed** command, the switch automatically configures both interface **speed** and **duplex** to **auto**.
- When you set the interface speed to **1000** (Mbps) or **auto 1000**, the duplex mode is full duplex. You cannot change the duplex mode.
- If the interface speed is set to **10** or **100**, the duplex mode is set to half duplex by default unless you explicitly configure it.

Caution Changing the interface speed and duplex mode configuration might shut down and restart the interface during the reconfiguration.

Setting the Interface Speed

If you set the interface speed to **auto** on a 10/100-Mbps Ethernet interface, speed and duplex are autonegotiated. The forced 10/100 autonegotiation feature allows you to limit interface speed auto negotiation up to 100 Mbps on a 10/100/1000BASE-T port.

To set the port speed for a 10/100-Mbps Ethernet interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface fastethernet slot/interface`	Specifies the interface to be configured.
Step 2	`Switch(config-if)# speed [10 | 100 | auto [10 | 100]]`	Sets the interface speed of the interface.

This example shows how to set the interface speed to 100 Mbps on the Fast Ethernet interface 5/4:

```
Switch(config)# interface fastethernet 5/4
Switch(config-if)# speed 100
```

This example shows how to allow Fast Ethernet interface 5/4 to autonegotiate the speed and duplex mode:

```
Switch(config)# interface fastethernet 5/4
Switch(config-if)# speed auto
```

Note This is analogous to specifying **speed auto 10 100**.

This example shows how to limit the interface speed to 10 and 100 Mbps on the Gigabit Ethernet interface 1/1 in auto-negotiation mode:

```
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# speed auto 10 100
```

This example shows how to limit speed negotiation to 100 Mbps on the Gigabit Ethernet interface 1/1:

```
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# speed auto 100
```

Note Turning off autonegotiation on a Gigabit Ethernet interface will result in the port being forced into 1000 Mbps and full-duplex mode.

To turn off the port speed autonegotiation for Gigabit Ethernet interface 1/1, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface gigabitethernet1/1`	Specifies the interface to be configured.
Step 2	`Switch(config-if)# speed nonegotiate`	Disables autonegotiation on the interface.

To restore autonegotiation, enter the **no speed nonegotiate** command in the interface configuration mode.

Note For the blocking ports on the WS-X4416 module, do not set the speed to autonegotiate.

Setting the Interface Duplex Mode

Note When the interface is set to 1000 Mbps, you cannot change the duplex mode from full duplex to half duplex.

To set the duplex mode of a Fast Ethernet interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface fastethernet` *slot/interface*	Specifies the interface to be configured.
Step 2	`Switch(config-if)# duplex [auto	full	half]`	Sets the duplex mode of the interface.

This example shows how to set the interface duplex mode to full on Fast Ethernet interface 5/4:

```
Switch(config)# interface fastethernet 5/4
Switch(config-if)# duplex full
```

Displaying the Interface Speed and Duplex Mode Configuration

To display the interface speed and duplex mode configuration for an interface, perform this task:

Command	Purpose
`Switch# show interfaces [fastethernet	gigabitethernet	tengigabitethernet] slot/interface`	Displays the interface speed and duplex mode configuration.

This example shows how to display the interface speed and duplex mode of Fast Ethernet interface 6/1:

```
Switch# show interface fastethernet 6/1
FastEthernet6/1 is up, line protocol is up
  Hardware is Fast Ethernet Port, address is 0050.547a.dee0 (bia 0050.547a.dee0)
  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 00:00:54, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 50/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     50 packets input, 11300 bytes, 0 no buffer
     Received 50 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
     1456 packets output, 111609 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     1 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Switch#
```

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, enter the following command:

Command	Purpose
`Switch(config-if)# description string`	Adds a description for an interface.

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

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

Configuring Flow Control

Gigabit Ethernet ports use Flow Control to slow-down the transmission of incoming packets. If a buffer on a Gigabit Ethernet port runs out of space, the port transmits a special packet that requests remote ports to delay sending packets for a period of time. The port can also receive this special packet from its link-partner for the same purpose. This special packet is called a *pause frame*.

The default settings for Gigabit Ethernet interfaces are as follows:

- Sending pause frames is off—non-oversubscribed Gigabit Ethernet interfaces.
- Receiving pause frames is desired—non-oversubscribed Gigabit Ethernet interfaces.
- Sending pause frames is on—Oversubscribed Gigabit Ethernet interfaces.
- Receiving pause frames is desired—Oversubscribed Gigabit Ethernet interfaces

The default settings for Tengigabit Ethernet interfaces are as follows:

- Sending pause frames is off.
- Receiving pause frames is on.

Note **desired** is not a flow control option on the Tengigabit Ethernet interfaces.

To configure flow control, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and specifies the interface to be enabled for flowcontrol.
Step 3	`Switch(config-if)#` **`flowcontrol`** `{`**`receive`** `|` **`send`**`} {`**`off`** `|` **`on`** `|` **`desired`**`}`	Configures a Gigabit Ethernet port to send or receive pause frames.
Step 4	`Switch(config-if)#` **`end`**	Returns to configuration mode.
Step 5	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.

This example shows how to configure flow control on an oversubscribed Gigabit Ethernet port 7/5:

```
Switch# configure terminal
Switch(config)# interface g7/5
Switch(config-if)# flowcontrol send on
Switch(config-if)# end
Switch)# show interfaces gigabitEthernet 7/5 capabilities
GigabitEthernet7/5
  Model:                  WS-X4548-GB-RJ45-RJ-45
  Type:                   10/100/1000-TX
  Speed:                  10,100,1000,auto
  Duplex:                 half,full,auto
  Trunk encap. type:      802.1Q,ISL
  Trunk mode:             on,off,desirable,nonegotiate
  Channel:                yes
  Broadcast suppression: percentage(0-100), hw
  Flowcontrol:            rx-(off,on,desired),tx-(off,on,desired)
  VLAN Membership:        static, dynamic
  Fast Start:             yes
  Queuing:                rx-(N/A), tx-(1p3q1t, Sharing/Shaping)
  CoS rewrite:           yes
  ToS rewrite:           yes
  Inline power:          no
  SPAN:                  source/destination
  UDLD:                  yes
  Link Debounce:         no
  Link Debounce Time:    no
  Port Security:         yes
  Dot1x:                 yes
  Maximum MTU:           1552 bytes (Baby Giants)
  Multiple Media Types:  no
  Diagnostic Monitoring: N/A

Switch)# show flowcontrol interface GigabitEthernet 7/5
Port       Send FlowControl  Receive FlowControl  RxPause TxPause
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------- -------
Gi7/5      on       off      desired  off         0       0
```

This example shows the output of the **show interfaces** and **show flowcontrol** commands on an non-overscribed Gigabit Ethernet port 5/5:

```
Switch# show interfaces gigabitEthernet 5/5 capabilities
GigabitEthernet5/5
  Model:                 WS-X4306-GB-Gbic
  Type:                  No Gbic
  Speed:                 1000
  Duplex:                full
  Trunk encap. type:     802.1Q,ISL
  Trunk mode:            on,off,desirable,nonegotiate
  Channel:               yes
  Broadcast suppression: percentage(0-100), hw
  Flowcontrol:           rx-(off,on,desired),tx-(off,on,desired)
  VLAN Membership:       static, dynamic
  Fast Start:            yes
  Queuing:               rx-(N/A), tx-(1p3q1t, Sharing/Shaping)
  CoS rewrite:           yes
  ToS rewrite:           yes
  Inline power:          no
  SPAN:                  source/destination
  UDLD:                  yes
  Link Debounce:         no
  Link Debounce Time:    no
  Port Security:         yes
  Dot1x:                 yes
  Maximum MTU:           9198 bytes (Jumbo Frames)
  Multiple Media Types:  no
  Diagnostic Monitoring: N/A

Switch# show flowcontrol interface gigabitEthernet 5/5
Port       Send FlowControl  Receive FlowControl  RxPause TxPause
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------- -------
Gi5/5      off      off      desired  off         0       0
```

This example shows the output of the **show interfaces** and **show flowcontrol** commands on an unsupported Fast Ethernet port 3/5:

```
Switch# show interfaces fa3/5 capabilities
FastEthernet3/5
  Model:                 WS-X4148-RJ-45
  Type:                  10/100BaseTX
  Speed:                 10,100,auto
  Duplex:                half,full,auto
  Trunk encap. type:     802.1Q,ISL
  Trunk mode:            on,off,desirable,nonegotiate
  Channel:               yes
  Broadcast suppression: percentage(0-100), sw
  Flowcontrol:           rx-(none),tx-(none)
  VLAN Membership:       static, dynamic
  Fast Start:            yes
  Queuing:               rx-(N/A), tx-(1p3q1t, Shaping)
  CoS rewrite:           yes
  ToS rewrite:           yes
  Inline power:          no
  SPAN:                  source/destination
  UDLD:                  yes
  Link Debounce:         no
  Link Debounce Time:    no
  Port Security:         yes
  Dot1x:                 yes
  Maximum MTU:           1552 bytes (Baby Giants)
  Multiple Media Types:  no
  Diagnostic Monitoring: N/A

Switch# show flowcontrol interface fa3/5
Port       Send FlowControl  Receive FlowControl  RxPause TxPause
           admin    oper     admin    oper
---------  -------- -------- -------- --------    ------- -------
Fa3/5      Unsupp.  Unsupp.  Unsupp.  Unsupp.      0       0
```

Configuring Jumbo Frame Support

These subsections describe jumbo frame support:

- Ports and Modules that Support Jumbo Frames, page 6-16
- Understanding Jumbo Frame Support, page 6-17
- Configuring MTU Sizes, page 6-18

Ports and Modules that Support Jumbo Frames

The following ports and modules support jumbo frames:

- Supervisor uplink ports
- WS-X4306-GB: all ports
- WS-X4232-GB-RJ: ports 1-2
- WS-X4418-GB: ports 1-2
- WS-X4412-2GB-TX: ports 13-14
- the 4648-GB-RJ45V
- WS-X4648-GB+RJ45V
- WS-X4706-10GE

  Each of the last three modules has two non-blocking ports that can support jumbo frames. Other ports are over-subscribed ports and cannot support jumbo frames.

Understanding Jumbo Frame Support

These sections describe jumbo frame support:

- Understanding Maximum Transmission Units, page 6-17
- Jumbo Frame Support Overview, page 6-17
- Ethernet Ports, page 6-18
- VLAN Interfaces, page 6-18

Understanding Maximum Transmission Units

The Catalyst 4500 series switch allows you to configure a maximum of 32 different maximum transmission unit (MTU) sizes systemwide. This means that the maximum number of different MTU sizes that you can configure with the **system mtu**, **mtu**, **ip mtu**, and **ipv6 mtu** command on all Layer 2 and Layer 3 interfaces combined is 32.

Also, the system stores the ipv4 and ipv6 MTU sizes configured on an interface separately. So, for every **system mtu** command or per interface **mtu** command, two separate MTU values are stored, one for ipv4 and one for ipv6. This further reduces the number of slots available (out of 32). However, only a single MTU value is stored for each **ip mtu** and **ipv6 mtu** commands.

If the new MTU value you are configuring is already present in the system (that is, configured on some other interface), then no new slot(s) will be allocated to store it again.

If the maximum limit of 32 is reached and an attempt is made to configure a new MTU size on a new interface, the system will only allow configuration to proceed if the new MTU size has previously been configured on some interface. Otherwise, an error message will be displayed and the default MTU size will be assigned to the interface being configured.

Jumbo Frame Support Overview

A jumbo frame is a frame larger than the default Ethernet size. Enable jumbo frame support by configuring a larger-than-default MTU size on a port or interface.

Catalyst 4500 series switch Ethernet LAN ports configured with a nondefault MTU size accept frames containing packets with a size between 1500 and 9198 bytes. With a nondefault MTU size configured, the packet size of ingress frames is checked. If the packet is larger than the configured MTU, it is dropped.

For traffic that needs to be routed, the MTU of the egress port is checked. If the MTU is smaller than the packet size, the packet is forwarded to the CPU. If the "do not fragment bit" is not set, it is fragmented. Otherwise, the packet is dropped.

Note Jumbo frame support does not fragment Layer 2 switched packets.

The Catalyst 4500 series switch does not compare the packet size with the MTU at the egress port, but jumbo frames are dropped in ports that do not support them. The frames can be transmitted in ports that do support jumbo frames, even though the MTU is not configured to jumbo size.

Note Jumbo frame support is only configured per interface; jumbo frame support cannot be configured globally.

Ethernet Ports

These sections describe configuring nondefault MTU sizes on Ethernet ports:

- Ethernet Port Overview, page 6-18
- Layer 3 and Layer 2 EtherChannels, page 6-18

Ethernet Port Overview

With Cisco IOS Release 12.2(25)EW, configuring a nondefault MTU size on certain Ethernet ports limits the size of ingress packets. The MTU does not impact the egress packets.

With releases earlier than Cisco IOS Release 12.1(13)EW, you can configure the MTU size only on Gigabit Ethernet .

Layer 3 and Layer 2 EtherChannels

With Release Cisco IOS Release 12.2(25)EW and later releases, you can configure all the interfaces in an EtherChannel provided that they have the same MTU. Changing the MTU of an EtherChannel changes the MTU of all member ports. If the MTU of a member port cannot be changed to the new value, that port is suspended (administratively shut down). A port cannot join an EtherChannel if the port has a different MTU. If a member port of an EtherChannel changes MTU, the member port is suspended.

VLAN Interfaces

If switch ports reside in the same VLAN, either configure all of the switch ports to handle jumbo frames and support the same MTU size, or configure none of them. However, such uniformity of MTU size in the same VLAN is not enforced.

When a VLAN has switch ports with different MTU size, packets received from a port with a larger MTU might be dropped when they are forwarded to a port with a smaller MTU.

If the switch ports in a VLAN have jumbo frames enabled, the corresponding SVI can have jumbo frames enabled. The MTU of an SVI should always be smaller than the smallest MTU among all the switch ports in the VLAN, but this condition is not enforced.

The MTU of a packet is not checked on the ingress side for an SVI; it is checked on the egress side of an SVI. If the MTU of a packet is larger than the MTU of the egress SVI, the packet will be sent to the CPU for fragmentation processing. If the “do not fragment” bit is not set, the packet is fragmented. Otherwise, the packet is dropped.

Configuring MTU Sizes

To configure the MTU size, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`interface`** `{{`**`vlan`** *`vlan_ID`*`} | {{`*`type`*[1] *`slot/port`*`} | {`**`port-channel`** *`port_channel_number`*`}` *`slot/port`*`}}`	Selects the interface to configure.
Step 2	`Switch(config-if)#` **`mtu`** *`mtu_size`*	Configures the MTU size.
	`Switch(config-if)#` **`no mtu`**	Reverts to the default MTU size (1500 bytes).
Step 3	`Switch(config-if)#` **`end`**	Exits configuration interface mode.

	Command	Purpose
Step 4	Switch(config)# **end**	Exits configuration mode.
Step 5	Switch# **show running-config interface** [{**fastethernet** | **gigabitethernet**} *slot/port*]	Verifies the running configuration.

[1] *type* = **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

Note When a linecard is *removed*, the MTU values configured on ports of that line card are unconfigured. This means that upon re-insertion of the linecard, you need to reconfigure all previous MTUs for ports of that line card from the CLI.

Note When configuring the MTU size for VLAN interfaces and Layer 3 and Layer 2 Ethernet ports, note that the supported MTU values are from 1500 to 9198 bytes.

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

```
switch# conf terminal
switch(config)# interface gi1/1
switch(config-if)# mtu 9198
switch(config-if)# end
switch(config)# end
switch# 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...>
switch#
```

For details on how to configure IP MTU size, refer to Configuring IP MTU Sizes, page 26-8.

Interacting with Baby Giants

The baby giants feature, introduced in Cisco IOS Release 12.1(12c)EW, uses the global command **system mtu <size>** to set the global baby giant MTU. This feature also allows certain interfaces to support Ethernet payload size of up to 1552 bytes.

Both the **system mtu** command and the per-interface **mtu** command can operate on interfaces that can support jumbo frames, but the per-interface **mtu** command takes precedence.

For example, before setting the per-interface MTU for interface gi1/1, you issue the **system mtu 1550** command to change the MTU for gi1/1 to 1550 bytes. Next, you issue the per-interface **mtu** command to change the MTU for gi1/1 to 9198 bytes. Now, if you change the baby giant MTU to 1540 bytes with the command **system mtu 1540**, the MTU for gi1/1 remains unchanged at 9198 bytes.

Configuring auto-MDIX on a Port

Note Supervisor Engine 6-E does not support auto-MDIX.

When automatic medium-dependent-interface crossover (auto-MDIX) is enabled on an port, the port automatically detects the required cable connection type (straight through or crossover) and configures the connection appropriately. When connecting switches without the auto-MDIX feature, you must use straight-through cables to connect to devices such as servers, workstations, or routers and crossover cables to connect to other switches or repeaters. With auto-MDIX enabled, you can use either type of cable to connect to other devices, and the interface automatically corrects for any incorrect cabling. For more information about cabling requirements, see the hardware installation guide.

Auto-MDIX is enabled by default. When you enable auto-MDIX, you must also set the speed on the port to **auto** so that for the feature to operate correctly. auto-MDIX is supported on copper media ports. It is not supported on fiber media ports.

Note The following linecards support Auto-MDIX by default, when port auto-negotiation is enabled: WS-X4424-GB-RJ45, WS-X4448-GB-RJ45 and WS-X4548-GB-RJ45. You cannot disable them with the **mdix** command.

Note The following linecards do not support Auto-MDIX, neither by default nor by CLI: WS-X4548-GB-RJ45V, WS-X4524-GB-RJ45V, and WS-X4506-GB-T.

Note The following linecards support Auto-MDIX through the CLI on their copper media ports: WS-X4124-RJ45, WS-X4148-RJ45 (hardware revision 3.0 or higher), and WS-X4232-GB-RJ45 (hardware revision 3.0 or higher).

Table 6-1 shows the link states that results from auto-MDIX settings and correct and incorrect cabling.

Table 6-1 Link Conditions and auto-MDIX Settings

Local Side auto-MDIX	Remote Side auto-MDIX	With Correct Cabling	With Incorrect Cabling
On	On	Link up	Link up
On	Off	Link up	Link up
Off	On	Link up	Link up
Off	Off	Link up	Link down

To configure auto-MDIX on a port, perform the following task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode for the physical interface to be configured.
Step 3	`Switch(config-if)# speed auto`	Configures the port to autonegotiate speed with the connected device.
Step 4	`Switch(config-if)# mdix auto`	Enables auto-MDIX on the port.
Step 5	`Switch(config-if)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 6	`Switch# show interfaces interface-id`	Verifies the configuration of the auto-MDIX feature on the interface.
Step 7	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To disable auto-MDIX, use the **no mdix auto** interface configuration command.

This example shows how to enable auto-MDIX on a port:

```
Switch# configure terminal
Switch(config)# interface fastethernet 6/5
Switch(config-if)# speed auto
Switch(config-if)# mdix auto
Switch(config-if)# end
```

Displaying the Interface auto-MDIX Configuration

To display the interface speed and duplex mode configuration for an interface, perform this task:

	Command	Purpose
Step 1	`Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	`Switch# show interfaces type slot/interface`	Displays the interface auto-MDIX configuration setting and operational state.

Depending on how the **speed auto** and the **mdix auto** commands are configured on a supported linecard interface, the **show interfaces** command displays the following possible auto-MDIX statuses:

Table 6-2 shows the auto-MDIX setting and operational state and the status of auto-MDIX.

Table 6-2 Auto-MDIX and Operational State

Auto-MDIX Setting And Operational State on an Interface	Description
Auto-MDIX on (operational: on)	Auto-MDIX is enabled and is fully functioning.
Auto-MDIX on (operational: off)	Auto-MDIX is enabled on this interface but it is not functioning. To allow auto-MDIX feature to function properly, you must also set the interface speed to be autonegotiated.
Auto-MDIX off	Auto-MDIX has been disabled with the **no mdix auto** command.

This example show s how to display the auto-MDIX configuration setting and its operational state on Fast Ethernet interface 6/1:

```
Switch# show interfaces fastethernet 6/1
FastEthernet6/1 is up, line protocol is up (connected)
  Hardware is Fast Ethernet Port, address is 0001.64fe.e5d0 (bia 0001.64fe.e5d0)
  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, link type is auto, media type is 10/100BaseTX
  input flow-control is unsupported output flow-control is unsupported
  Auto-MDIX on (operational: on)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:16, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     511 packets input, 74464 bytes, 0 no buffer
     Received 511 broadcasts (511 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     3552 packets output, 269088 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     1 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Switch#
```

Understanding Online Insertion and Removal

The online insertion and removal (OIR) feature supported on the Catalyst 4500 series switch allows you to remove and replace modules while the system is online. You can shut down the module before removal and restart it after insertion without causing other software or interfaces to shut down.

You do not need to enter a command to notify the software that you are going to remove or install a module. The system notifies the supervisor engine that a module has been removed or installed and scans the system for a configuration change. The newly installed module is initialized, and each interface type is verified against the system configuration; then the system runs diagnostics on the new interface. There is no disruption to normal operation during module insertion or removal.

If you remove a module and then replace it, or insert a different module of the same type into the same slot, no change to the system configuration is needed. An interface of a type that has been configured previously will be brought online immediately. If you remove a module and insert a module of a different type, the interface(s) on that module will be administratively up with the default configuration for that module.

Monitoring and Maintaining the Interface

The following sections describe how to monitor and maintain the interfaces:

- Monitoring Interface and Controller Status, page 6-23
- Clearing and Resetting the Interface, page 6-23
- Shutting Down and Restarting an Interface, page 6-24
- Configuring Interface Link Status and Trunk Status Events, page 6-24
- Resetting the Interface to the Default Configuration, page 6-26

Monitoring Interface and Controller Status

The Cisco IOS software for the Catalyst 4500 series switch 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, the controller status, and statistics about the interfaces. The following table lists some of the interface monitoring commands. (You can display the full list of **show** commands by entering the **show ?** command at the EXEC prompt.) These commands are fully described in the *Interface Command Reference*.

To display information about the interface, perform this task:

	Command	Purpose
Step 1	`Switch#` **`show interfaces`** `[`*`type slot/interface`*`]`	Displays the status and configuration of all interfaces or of a specific interface.
Step 2	`Switch#` **`show running-config`**	Displays the configuration currently running in RAM.
Step 3	`Switch#` **`show protocols`** `[`*`type slot/interface`*`]`	Displays the global (system-wide) and interface-specific status of any configured protocol.
Step 4	`Switch#` **`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 interface 5/5:

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

Clearing and Resetting the Interface

To clear the interface counters shown with the **show interfaces** command, enter the following command:

Command	Purpose
`Switch# clear counters` `{`*`type slot/interface`*`}`	Clears interface counters.

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

```
Switch# clear counters fastethernet 5/5
Clear "show interface" counters on this interface [confirm] y
Switch#
*Sep 30 08:42:55: %CLEAR-5-COUNTERS: Clear counter on interface FastEthernet5/5
by vty1 (171.69.115.10)
Switch#
```

The **clear counters** command (without any arguments) clears all the current interface counters from all interfaces.

Note The **clear counters** command does not clear counters retrieved with SNMP; it clears only those counters displayed with the EXEC **show interfaces** command.

Shutting Down and Restarting an Interface

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

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

	Command	Purpose
Step 1	`Switch(config)#` **`interface`** `{`**`vlan`** *`vlan_ID`*`}	{{`**`fastethernet`**`	`**`gigabitethernet`**`	`**`tengigabitethernet`**`}` *`slot/port`*`}	{`**`port-channel`** *`port_channel_number`*`}`	Specifies the interface to be configured.
Step 2	`Switch(config-if)#` **`shutdown`**	Shuts down the interface.
Step 3	`Switch(config-if)#` **`no shutdown`**	Reenables the interface.

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

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

This example shows how to reenable Fast Ethernet interface 5/5:

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

To verify whether an interface is disabled, enter the EXEC **show interfaces** command. An interface that has been shut down will appear as “administratively down.”

Configuring Interface Link Status and Trunk Status Events

You can configure interface link status and trunk status events. On the Catalyst 4500 series switch, the following interface logging event notifications are supported both globally and per interface:

- Enable/disable notification on the interface whenever its data link status is changed.
- Enable/disable notification on the trunk interface whenever its trunking status is changed.

Use the **[no] logging event link-status [use-global]** command to enable/disable the interface link status event. Use the **[no] logging event trunk-status [use-global]** command to enable/disable the interface trunk status event.

Each interface link status logging event can be configured in one of the following states:

- **logging event link-status** - Link status logging event is enabled explicitly on the interface regardless of the switch global setting.
- **no logging event link-status** - Link status logging event is disabled explicitly on the interface regardless of the switch global setting.

- **logging event link-status use-global** - This is the default link status logging event configuration on the interface; its configuration should follow the switch global link status logging event setting.

The interface trunk status logging event can be configured in the same configuration states.

Configuring Link Status Event Notification for an Interface

To enable/disable a link status logging event, enter one of the following commands:

Command	Purpose
`Switch(config-if)# logging event link-status`	Enables interface link status logging.
`Switch(config-if)# no logging event link-status`	Disables interface link status logging.
`Switch(config-if)# logging event link-status use-global`	Specifies the global default setting for interface link status logging.

Global Settings

You can also provide a global configuration for the corresponding logging event. A global configuration provides default logging settings for all interfaces. The **[no] logging event link-status global** command lets you enable/disable the interface link status logging for the entire switch. The **[no] logging event trunk-status global** command lets you enable/disable interface trunk status logging for the entire switch.

Each interface link status logging event, if not configured at the interface level, will use the following global logging event setting:

- **logging event link-status global** - Link status logging event is enabled, if not configured on the interface.
- **no logging event link-status global** - Link status logging event is disabled, if not configured on the interface.

The interface trunk status logging event has similar global configurations.

Configuring a Switch Global Link Status Logging Event

To enable/disable the global link status logging event, enter one of the following commands:

Command	Purpose
`Switch(config-if)# logging event link-status global`	Enables global link status logging.
`Switch(config-if)# no logging event link-status global`	Disables global link status logging.

Result

The following example displays a summary of the operating states for the interface logging event under different combinations of global and interface logging settings:

```
global setting       interface setting       actual logging state
--------------       -----------------       --------------------
     on                   on                         on
     off                  on                         on
     on                   off                        off
     off                   off                       off
     on                    default(use-global)       on
     off                   default(use-global)       off
```

The following example displays the configuration and logging message output for link status and trunk status logging events:

```
//
// The global link status and trunk status logging events are enabled.
//
Switch# show running | include logging
show running | include logging
logging event link-status global
logging event trunk-status global
Switch#

//
// The interface link status and trunk status logging settings
// are set to default values, which follow regardless of the global
// setting.
//
Switch# show running interface g1/4
Building configuration...

Current configuration: 97 bytes
!
interface GigabitEthernet1/4
 switchport trunk encapsulation dot1q
 switchport mode trunk
end
Switch#

//
// The trunk status logging messages for the interface are
// displayed whenever the interface trunking status is changed.
// Here we change the other end node's trunking encapsulation
// from dot1q to isl.
//
3d00h: %DTP-5-ILGLCFG: Illegal config(on,isl--on,dot1q) on Gi1/4
3d00h: %DTP-5-ILGLCFG: Illegal config(on,isl--on,dot1q) on Gi1/4
3d00h: %DTP-5-ILGLCFG: Illegal config(on,isl--on,dot1q) on Gi1/4

//
// The link and trunk status logging message for the interface
// are displayed whenever the interface link status is changed.
// Here we do a "shut" and "no shut" on the other end link node.
//
3d00h: %DTP-5-NONTRUNKPORTON: Port Gi1/4 has become non-trunk
3d00h: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/4, changed state to down
3d00h: %LINK-3-UPDOWN: Interface GigabitEthernet1/4, changed state to
down
3d00h: %LINK-3-UPDOWN: Interface GigabitEthernet1/4, changed state to up
3d00h: %DTP-5-TRUNKPORTON: Port Gi1/4 has become dot1q trunk
3d00h: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/4, changed state to up
```

Resetting the Interface to the Default Configuration

If you have configured a interface with many command lines and you want to clear all the configuration on that interface, you can use the **default interface** global configuration command, as follows:

```
Switch(config)# default interface fastEthernet 3/5
Interface FastEthernet3/5 set to default configuration
```

This command will clear all the configurations and shutdown the interface:

```
Switch# show run interface fastethernet 3/5
Building configuration...

Current configuration : 58 bytes
!
interface FastEthernet3/5
 no ip address
 shutdown
end
```

CHAPTER 7

Checking Port Status and Connectivity

This chapter describes how to check switch port status and connectivity on the Catalyst 4500 series switch.

This chapter includes the following major sections:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Checking Module Status

The Catalyst 4500 series switch is a multimodule system. You can see which modules are installed, as well as the MAC address ranges and version numbers for each module, by using the **show module** command. You can use the [*mod_num*] argument to specify a particular module number to see detailed information on that module.

This example shows how to check module status for all modules on your switch:

```
Switch# show module all

Mod  Ports Card Type                              Model            Serial No.
----+-----+--------------------------------------+-----------------+-----------
 1     2   1000BaseX (GBIC) Supervisor Module     WS-X4014         JAB012345AB
 5    24   10/100/1000BaseTX (RJ45)               WS-X4424-GB-RJ45 JAB045304EY
 6    48   10/100BaseTX (RJ45)                    WS-X4148         JAB023402QK

 M MAC addresses                    Hw  Fw               Sw              Stat
--+--------------------------------+---+-----------------+---------------+-----
 1 0004.dd46.9f00 to 0004.dd46.a2ff 0.0 12.1(10r)EW(1.21) 12.1(10)EW(1)    Ok
 5 0050.3e7e.1d70 to 0050.3e7e.1d87 0.0                                    Ok
 6 0050.0f10.2370 to 0050.0f10.239f 1.0                                    Ok
Switch#
```

Checking Interfaces Status

You can view summary or detailed information on the switch ports using the **show interfaces status** command. To see summary information on all ports on the switch, enter the **show interfaces status** command with no arguments. Specify a particular module number to see information on the ports on that module only. Enter both the module number and the port number to see detailed information about the specified port.

To apply configuration commands to a particular port, you must specify the appropriate logical module. For more information, see the "Checking Module Status" section on page 7-1.

This example shows how to display the status of all interfaces on a Catalyst 4500 series switch, including transceivers. Output of this command displays "Unapproved GBIC" for non-Cisco transceivers:

```
Switch#show interfaces status

Port    Name               Status       Vlan       Duplex  Speed Type
Gi1/1                      notconnect   1            auto   auto No Gbic
Gi1/2                      notconnect   1            auto   auto No Gbic
Gi5/1                      notconnect   1            auto   auto 10/100/1000-TX
Gi5/2                      notconnect   1            auto   auto 10/100/1000-TX
Gi5/3                      notconnect   1            auto   auto 10/100/1000-TX
Gi5/4                      notconnect   1            auto   auto 10/100/1000-TX
Fa6/1                      connected    1          a-full  a-100 10/100BaseTX
Fa6/2                      connected    2          a-full  a-100 10/100BaseTX
Fa6/3                      notconnect   1            auto   auto 10/100BaseTX
Fa6/4                      notconnect   1            auto   auto 10/100BaseTX

Switch#
```

This example shows how to display the status of interfaces in error-disabled state:

```
Switch# show interfaces status err-disabled
Port    Name               Status         Reason
Fa9/4                      err-disabled   link-flap
informational error message when the timer expires on a cause
--------------------------------------------------------------
5d04h:%PM-SP-4-ERR_RECOVER:Attempting to recover from link-flap err-disable state on Fa9/4
Switch#
```

Displaying MAC Addresses

In addition to displaying the MAC address range for a module using the **show module** command, you can display the MAC address table information of a specific MAC address or a specific interface in the switch using the **show mac-address-table address** and **show mac-address-table interface** commands.

This example shows how to display MAC address table information for a specific MAC address:

```
Switch# show mac-address-table address 0050.3e8d.6400
vlan   mac address     type    protocol  qos            ports
-----+---------------+--------+---------+---+--------------------------------
 200  0050.3e8d.6400  static   assigned  --  Switch
 100  0050.3e8d.6400  static   assigned  --  Switch
   5  0050.3e8d.6400  static   assigned  --  Switch
   4  0050.3e8d.6400  static        ipx  --  Switch
   1  0050.3e8d.6400  static        ipx  --  Switch
   1  0050.3e8d.6400  static   assigned  --  Switch
   4  0050.3e8d.6400  static   assigned  --  Switch
   5  0050.3e8d.6400  static        ipx  --  Switch
 100  0050.3e8d.6400  static        ipx  --  Switch
 200  0050.3e8d.6400  static        ipx  --  Switch
 100  0050.3e8d.6400  static      other  --  Switch
 200  0050.3e8d.6400  static      other  --  Switch
   5  0050.3e8d.6400  static      other  --  Switch
   4  0050.3e8d.6400  static         ip  --  Switch
   1  0050.3e8d.6400  static         ip  --  Route
   1  0050.3e8d.6400  static      other  --  Switch
   4  0050.3e8d.6400  static      other  --  Switch
   5  0050.3e8d.6400  static         ip  --  Switch
 200  0050.3e8d.6400  static         ip  --  Switch
 100  0050.3e8d.6400  static         ip  --  Switch
Switch#
```

This example shows how to display MAC address table information for a specific interface:

```
Switch# show mac-address-table interface gigabit 1/1
Multicast Entries
 vlan    mac address     type    ports
-------+---------------+-------+------------------------------------------
   1    ffff.ffff.ffff system Switch,Gi6/1,Gi6/2,Gi6/9,Gi1/1
Switch#
```

Checking Cable Status Using Time Domain Reflectometer

Note Time Domain Reflectometer is *not* supported on Supervisor Engine 6-E.

You can use the Time Domain Reflectometer (TDR) feature to determine if cabling is at fault when you cannot establish a link.

Note This test is especially important when replacing an existing switch, upgrading to Gigabit Ethernet, or installing new cable plants.

Overview

With TDR, you can check the status of copper cables on the 48-port 10/100/1000 BASE-T modules for the Catalyst 4500 series switch (WS-X4548-GB-RJ45, WS-X4548-GB-RJ45V, WS-X4524-GB-RJ45V, WS-X4013+TS, WS-C4948, and WS-C4948-10GE). TDR detects a cable fault by sending a signal through the cable and reading the signal that is reflected back. All or part of the signal can be reflected back either by cable defects or by the end of the cable.

Note There are four pairs of standard category 5 cable. Each pair can assume one of the following states: open (not connected), broken, shorted, or terminated. The TDR test detects all four states and displays the first three as “Fault” conditions, and displays the fourth as “Terminated.” Although the CLI output is shown, the cable length is shown only if the state is “Faulty.”

Running the TDR Test

To start the TDR test, perform this task in privileged mode:

	Command	Purpose

Step 1	`Switch#` **`test cable-diagnostics tdr`**
	{***`interface`*** {*interface interface-number*}}
		Start the TDR test.

Step 2	`Switch#` **`show cable-diagnostics tdr`**
	{***`interface`*** {*interface interface-number*}}
		Show the TDR test counter information.

This example shows how to start the TDR test on port 1 on module 2:

```
Switch# test cable-diagnostics tdr int gi2/1
Switch#
```

This example shows the message that displays when the TDR test is not supported on a module:

```
Switch# test cable-diagnostics tdr int gi2/1
00:03:15:%C4K_IOSDIAGMAN-4-TESTNOTSUPPORTEDONMODULE: Online cable
diag tdr test is not supported on this module
Switch#
```

This example shows how to display TDR test results for a port:

```
Switch# show cable-diagnostics tdr interface gi4/13
Interface Speed  Local pair Cable length Remote channel Status
Gi4/13    0Mbps   1-2         102 +-2m     Unknown       Fault
                  3-6         100 +-2m     Unknown       Fault
                  4-5         102 +-2m     Unknown       Fault
                  7-8         102 +-2m     Unknown       Fault
```

Note This command will be deprecated in future releases of Cisco IOS software. Please use the diagnostic start and the **show diagnostic result** commands to run the TDR test and display the test results.

Note TDR is a port test; the port can not handle traffic for the duration of the test (generally, 1 minute).

Guidelines

The following guidelines apply to the use of TDR:

- If you connect a port undergoing a TDR test to an Auto-MDIX enabled port, the TDR result might be invalid. In those instances, the port on the WS-X4148-RJ45V should be administratively down before the start of the TDR test.
- If you connect a port undergoing a TDR test to a 100BASE-T port such as that on the WS-X4148-RJ45V, the unused pairs (4-5 and 7-8) will be reported as faulty because the remote end does not terminate these pairs.
- Do not change the port configuration while the TDR test is running.
- Due to cable characteristics, you should run the TDR test multiple times to get accurate results.
- Do not change port status (i.e. remove the cable at the near or far end), as this might make the results inaccurate.

Using Telnet

You can access the switch command-line interface (CLI) using Telnet. In addition, you can use Telnet from the switch to access other devices in the network. You can have up to eight simultaneous Telnet sessions.

Before you can open a Telnet session to the switch, you must first set the IP address (and in some cases the default gateway) for the switch. For information about setting the IP address and default gateway, see Chapter 1, “Configuring the Switch for the First Time.”

Note To establish a Telnet connection to a host by using the hostname, configure and enable DNS.

To establish a Telnet connection to another device on the network from the switch, perform this task:

Command	Purpose
Switch# **telnet** *host* [*port*]	Opens a Telnet session to a remote host.

This example shows how to establish a Telnet connection from the switch to the remote host named labsparc:

```
Switch# telnet labsparc
Trying 172.16.10.3...
Connected to labsparc.
Escape character is '^]'.

UNIX(r) System V Release 4.0 (labsparc)

login:
```

Changing the Logout Timer

The logout timer automatically disconnects a user from the switch when the user is idle for longer than the specified time. To set the logout timer, perform this task:

Command	Purpose
Switch# **logoutwarning** *number*	Changes the logout timer value (a timeout value of 0 prevents idle sessions from being disconnected automatically). Use the **no** keyword to return to the default value.

Monitoring User Sessions

You can display the currently active user sessions on the switch using the **show users** command. The command output lists all active console port and Telnet sessions on the switch.

To display the active user sessions on the switch, perform this task in privileged EXEC mode:

Command	Purpose
Switch# **show users [all]**	Displays the currently active user sessions on the switch.

This example shows the output of the **show users** command when local authentication is enabled for console and Telnet sessions (the asterisk [*] indicates the current session):

```
Switch# show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                idle                 00:00:00

  Interface      User        Mode                     Idle     Peer Address

Switch# show users all
    Line       User       Host(s)              Idle       Location
*  0 con 0                idle                 00:00:00
   1 vty 0                                     00:00:00
   2 vty 1                                     00:00:00
   3 vty 2                                     00:00:00
   4 vty 3                                     00:00:00
   5 vty 4                                     00:00:00

  Interface      User        Mode                     Idle     Peer Address
Switch#
```

To disconnect an active user session, perform this task:

Command	Purpose
Switch# **disconnect** {**console** | *ip_addr*}	Disconnects an active user session on the switch.

This example shows how to disconnect an active console port session and an active Telnet session:

```
Switch> disconnect console
Console session disconnected.
Console> (enable) disconnect tim-nt.bigcorp.com
Telnet session from tim-nt.bigcorp.com disconnected. (1)
Switch# show users
  Session  User             Location
  -------- ---------------- -------------------------
  telnet   jake             jake-mac.bigcorp.com
```

```
* telnet   suzy            suzy-pc.bigcorp.com
Switch#
```

Using Ping

These sections describe how to use IP ping:

- Understanding How Ping Works, page 7-7
- Running Ping, page 7-7

Understanding How Ping Works

You can use the **ping** command to verify connectivity to remote hosts. If you attempt to ping a host in a different IP subnetwork, you must define a static route to the network or configure a router to route between those subnets.

The **ping** command is configurable from normal executive and privileged EXEC mode. Ping returns one of the following responses:

- Normal response—The normal response (*hostname* is alive) occurs in 1 to 10 seconds, depending on network traffic.
- Destination does not respond—If the host does not respond, a No Answer message is returned.
- Unknown host—If the host does not exist, an Unknown Host message is returned.
- Destination unreachable—If the default gateway cannot reach the specified network, a Destination Unreachable message is returned.
- Network or host unreachable—If there is no entry in the route table for the host or network, a Network or Host Unreachable message is returned.

To stop a ping in progress, press **Ctrl-C**.

Running Ping

To ping another device on the network from the switch, perform this task in normal executive and privileged EXEC mode:

Command	Purpose
Switch# **ping** *host*	Checks connectivity to a remote host.

This example shows how to ping a remote host from normal executive mode:

```
Switch# ping labsparc
labsparc is alive
Switch> ping 72.16.10.3
12.16.10.3 is alive
Switch#
```

This example shows how to enter a **ping** command in privileged EXEC mode specifying the number of packets, the packet size, and the timeout period:

```
Switch# ping
```

```
Target IP Address []: 12.20.5.19
Number of Packets [5]: 10
Datagram Size [56]: 100
Timeout in seconds [2]: 10
Source IP Address [12.20.2.18]: 12.20.2.18
!!!!!!!!!!

----12.20.2.19 PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 1/1/1
Switch
```

Using IP Traceroute

These sections describe how to use IP traceroute feature:

- Understanding How IP Traceroute Works, page 7-8
- Running IP Traceroute, page 7-8

Understanding How IP Traceroute Works

You can use IP traceroute to identify the path that packets take through the network on a hop-by-hop basis. The command output displays all network layer (Layer 3) devices, such as routers, that the traffic passes through on the way to the destination.

Layer 2 switches can participate as the source or destination of the **trace** command but will not appear as a hop in the **trace** command output.

The **trace** command uses the Time To Live (TTL) field in the IP header to cause routers and servers to generate specific return messages. Traceroute starts by sending a User Datagram Protocol (UDP) datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it drops the datagram and sends back an Internet Control Message Protocol (ICMP) Time-Exceeded message to the sender. Traceroute determines the address of the first hop by examining the source address field of the ICMP Time-Exceeded message.

To identify the next hop, traceroute sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the Time-Exceeded message to the source. This process continues until the TTL is incremented to a value large enough for the datagram to reach the destination host or until the maximum TTL is reached.

To determine when a datagram reaches its destination, traceroute sets the UDP destination port in the datagram to a very large value that the destination host is unlikely to be using. When a host receives a datagram with an unrecognized port number, it sends an ICMP Port Unreachable error message to the source. The Port Unreachable error message indicates to traceroute that the destination has been reached.

Running IP Traceroute

To trace the path that packets take through the network, perform this task in EXEC or privileged EXEC mode:

Command	Purpose
Switch# **trace** [*protocol*] [*destination*]	Runs IP traceroute to trace the path that packets take through the network.

This example shows use the **trace** command to display the route a packet takes through the network to reach its destination:

```
Switch# trace ip ABA.NYC.mil

Type escape sequence to abort.
Tracing the route to ABA.NYC.mil (26.0.0.73)
  1 DEBRIS.CISCO.COM (192.180.1.6) 1000 msec 8 msec 4 msec
  2 BARRNET-GW.CISCO.COM (192.180.16.2) 8 msec 8 msec 8 msec
  3 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 8 msec 4 msec 4 msec
  4 BB2.SU.BARRNET.NET (192.200.254.6) 8 msec 8 msec 8 msec
  5 SU.ARC.BARRNET.NET (192.200.3.8) 12 msec 12 msec 8 msec
  6 MOFFETT-FLD-MB.in.MIL (192.52.195.1) 216 msec 120 msec 132 msec
  7 ABA.NYC.mil (26.0.0.73) 412 msec 628 msec 664 msec
Switch#
```

Using Layer 2 Traceroute

The Layer 2 traceroute feature allows the switch to identify the physical path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses. It determines the path by using the MAC address tables of the switches in the path. When the switch detects a device in the path that does not support Layer 2 traceroute, the switch continues to send Layer 2 trace queries and lets them time out.

If you want the switch to trace the path from a host on a source device to a host on a destination device, the switch can identify only the path from the source device to the destination device. It cannot identify the path that a packet takes from source host to the source device or from the destination device to the destination host.

These sections describe how to use the Layer 2 traceroute feature:

- Layer 2 Traceroute Usage Guidelines, page 7-9
- Running Layer 2 Traceroute, page 7-10

Layer 2 Traceroute Usage Guidelines

These are the Layer 2 traceroute usage guidelines:

- CDP must be enabled on all the devices in the network. For Layer 2 traceroute to functional properly, do not disable CDP.

  If any devices in the physical path are transparent to CDP, the switch cannot identify the path through these devices.

  Note For more information about enabling CDP, see Chapter 22, “Configuring CDP.”

- All switches in the physical path must have IP connectivity. When a switch is reachable from another switch, you can test connectivity by using the **ping** command in privileged EXEC mode.

- The maximum number of hops identified in the path is ten.
- You can enter the **traceroute mac** or the **traceroute mac ip** command in privileged EXEC mode on a switch that is not in the physical path from the source device to the destination device. All switches 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 switch uses Address Resolution Protocol (ARP) to associate the IP address with the corresponding MAC address and the VLAN ID.
  - If an ARP entry exists for the specified IP address, the switch uses the associated MAC address and identifies the physical path.
  - If an ARP entry does not exist, the switch 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 feature is not supported. When more than one CDP neighbor is detected on a port, the Layer 2 path is not identified, and an error message appears.
- This feature is not supported in Token Ring VLANs.

Running Layer 2 Traceroute

To display the physical path that a packet takes from a source device to a destination device, perform either one of these tasks in privileged EXEC mode:

Command	Purpose
Switch# **traceroute mac** {*source-mac-address*} {*destination-mac-address*}	Runs Layer 2 traceroute to trace the path that packets take through the network.

or

Command	Purpose
Switch# **traceroute mac ip** {*source-mac-address*} {*destination-mac-address*}	Runs IP traceroute 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:

```
Switch# 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
Switch#

Switch# traceroute mac ip 2.2.66.66 2.2.22.22 detail
Translating IP to mac .....
2.2.66.66 => 0000.0201.0601
2.2.22.22 => 0000.0201.0201

Source 0000.0201.0601 found on con6[WS-C2950G-24-EI] (2.2.6.6)
con6 / WS-C2950G-24-EI / 2.2.6.6 :
        Fa0/1 [auto, auto] => Fa0/3 [auto, auto]
con5 / WS-C2950G-24-EI / 2.2.5.5 :
        Fa0/3 [auto, auto] => Gi0/1 [auto, auto]
con1 / WS-C3550-12G / 2.2.1.1 :
        Gi0/1 [auto, auto] => Gi0/2 [auto, auto]
con2 / WS-C3550-24 / 2.2.2.2 :
        Gi0/2 [auto, auto] => Fa0/1 [auto, auto]
Destination 0000.0201.0201 found on con2[WS-C3550-24] (2.2.2.2)
Layer 2 trace completed.
Switch#
```

Configuring ICMP

Internet Control Message Protocol (ICMP) provides many services that control and manage IP connections. ICMP messages are sent by routers or access servers to hosts or other routers when a problem is discovered with the Internet header. For detailed information on ICMP, refer to RFC 792.

Enabling ICMP Protocol Unreachable Messages

If the Cisco IOS software receives a nonbroadcast packet that uses an unknown protocol, it sends an ICMP Protocol Unreachable message back to the source.

Similarly, if the software receives a packet that it is unable to deliver to the ultimate destination because it knows of no route to the destination address, it sends an ICMP Host Unreachable message to the source. This feature is enabled by default.

To enable the generation of ICMP Protocol Unreachable and Host Unreachable messages, enter the following command in interface configuration mode:

Command	Purpose
`Switch (config-if)# [`**`no`**`]` **`ip unreachables`**	Enables ICMP destination unreachable messages. Use the **no** keyword to disable the ICMP destination unreachable messages.

Caution If you issue the **no ip unreachables** command, you will break “path MTU discovery” functionality. Routers in the middle of the network might be forced to fragment packets.

To limit the rate that Internet Control Message Protocol (ICMP) destination unreachable messages are generated, perform this task:

Command	Purpose
`Switch (config)# [no] ip icmp rate-limit unreachable [df] milliseconds`	Limits the rate that ICMP destination messages are generated. Use the **no** keyword to remove the rate limit and reduce the CPU usage.

Enabling ICMP Redirect Messages

Data routes are sometimes less than optimal. For example, it is possible for the router to be forced to resend a packet through the same interface on which it was received. If this occurs, the Cisco IOS software sends an ICMP Redirect message to the originator of the packet telling the originator that the router is on a subnet directly connected to the receiving device, and that it must forward the packet to another system on the same subnet. The software sends an ICMP Redirect message to the packet's originator because the originating host presumably could have sent that packet to the next hop without involving this device at all. The Redirect message instructs the sender to remove the receiving device from the route and substitute a specified device representing a more direct path. This feature is enabled by default.

However, when Hot Standby Router Protocol (HSRP) is configured on an interface, ICMP Redirect messages are disabled (by default) for the interface. For more information on HSRP, refer to the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt1/1cdip.htm

To enable the sending of ICMP Redirect messages if the Cisco IOS software is forced to resend a packet through the same interface on which it was received, enter the following command in interface configuration mode:

Command	Purpose
`Switch (config-if)# [no] ip redirects`	Enables ICMP Redirect messages. Use the **no** keyword to disable the ICMP Redirect messages and reduce CPU usage.

Enabling ICMP Mask Reply Messages

Occasionally, network devices must know the subnet mask for a particular subnetwork in the internetwork. To obtain this information, devices can send ICMP Mask Request messages. These messages are responded to by ICMP Mask Reply messages from devices that have the requested information. The Cisco IOS software can respond to ICMP Mask Request messages if the ICMP Mask Reply function is enabled.

To have the Cisco IOS software respond to ICMP mask requests by sending ICMP Mask Reply messages, perform this task:

Command	Purpose
`Switch (config-if)# [no] ip mask-reply`	Enables response to ICMP destination mask requests. Use the **no** keyword to disable this functionality.

CHAPTER 8

Configuring Supervisor Engine Redundancy Using RPR and SSO

Catalyst 4500 series switches allow a redundant supervisor engine to take over if the active supervisor engine fails. In software, supervisor engine redundancy is enabled by running the redundant supervisor engine in route processor redundancy (RPR) or stateful switchover (SSO) operating mode.

Note Stateful Switchover is *not* supported on Supervisor Engine 6-E.

Note The minimum ROMMON requirement for running SSO is Cisco IOS Release 12.1(20r)EW1 or Cisco IOS Release 12.2(20r)EW1.

This chapter describes how to configure supervisor engine redundancy on the Catalyst 4507R and Catalyst 4510R switches.

Note For information on Cisco nonstop forwarding (NSF) with SSO, see Chapter 1, "Configuring Cisco NSF with SSO Supervisor Engine Redundancy."

This chapter contains these major sections:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Understanding Supervisor Engine Redundancy

These sections describe supervisor engine redundancy:

- Overview, page 8-2
- RPR Operation, page 8-3
- SSO Operation, page 8-3
- "Understanding Supervisor Engine Redundancy Synchronization" section on page 8-5

Overview

With supervisor engine redundancy enabled, if the active supervisor engine fails or if a manual switchover is performed, the redundant supervisor engine becomes the active supervisor engine. The redundant supervisor engine has been automatically initialized with the startup configuration of the active supervisor engine, shortening the switchover time (30 seconds or longer in RPR mode, depending on the configuration; subsecond in SSO mode).

In addition to the reduced switchover time, supervisor engine redundancy supports the following:

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

  Supervisor engine redundancy allows OIR of the redundant supervisor engine for maintenance. When the redundant supervisor engine is inserted, the active supervisor engine detects its presence, and the redundant supervisor engine boots into a partially-initialized state in RPR mode and a fully-initialized state in SSO mode.

- Software upgrade. (See the "Performing a Software Upgrade" section on page 8-13.)

  To minimize down time during software changes on the supervisor engine, load the new image on the redundant supervisor engine, and conduct a switchover.

When power is first applied to a switch, the supervisor engine that boots first becomes the active supervisor engine and remains active until a switchover occurs.

A switchover will occur when one or more of the following events take place:

- The active supervisor engine fails (due to either hardware or software function) or is removed.
- A user forces a switchover.
- A user reloads the active supervisor engine.

Table 8-1 provides information about chassis and supervisor engine support for redundancy.

Table 8-1 Chassis and Supervisor Support

Chassis (Product Number)	Supported Supervisor Engines
Catalyst 4507R (WS-C4507R)	Supports redundant Supervisor Engine II-Plus (WS-X4013+), redundant Supervisor Engine II-Plus (WS-X4013+GE), Supervisor Engine IV (WS-X4515), redundant Supervisor Engine V (WS-X4516), and redundant Supervisor Engine V (WS-X4516-10GE)
Catalyst 4510R (WS-C4510R)	Supports redundant Supervisor Engine V (WS-X4516) and redundant Supervisor Engine V (WS-X4516-10GE)

RPR Operation

RPR is supported in Cisco IOS Release 12.2(12c)EW and later releases. When a redundant supervisor engine runs in RPR mode, it starts up in a partially-initialized state and is synchronized with the persistent configuration of the active supervisor engine.

Note Persistent configuration includes the following components: startup-config, boot variables, config-register, and VLAN database.

The redundant supervisor engine pauses the startup sequence after basic system initialization, and in the event that the active supervisor engine fails, the redundant supervisor engine becomes the new active supervisor engine.

In a supervisor engine switchover, traffic is disrupted because in the RPR mode all of the physical ports restart since there is no state maintained between supervisor engines relating to module types and statuses. When the redundant supervisor engine completes its initialization, it will read hardware information directly from the module.

SSO Operation

Note Stateful Switchover is *not* supported on Supervisor Engine 6-E.

SSO is supported in Cisco IOS Release 12.2(20)EWA and later releases. When a redundant supervisor engine runs in SSO mode, the redundant supervisor engine starts up in a fully-initialized state and synchronizes with the persistent configuration and the running configuration of the active supervisor engine. It subsequently maintains the state on the protocols listed below, and all changes in hardware and software states for features that support stateful switchover are kept in sync. Consequently, it offers zero interruption to Layer 2 sessions in a redundant supervisor engine configuration.

Because the redundant supervisor engine recognizes the hardware link status of every link, ports that were active before the switchover will remain active, including the uplink ports. However, because uplink ports are physically on the supervisor engine, they will be disconnected if the supervisor engine is removed.

If the active supervisor engine fails, the redundant supervisor engine become active. This newly active supervisor engine uses existing Layer 2 switching information to continue forwarding traffic. Layer 3 forwarding will be delayed until the routing tables have been repopulated in the newly active supervisor engine.

SSO supports stateful switchover of the following Layer 2 features. The state of these features is preserved between both the active and redundant supervisor engines:

- 802.3
- 802.3u
- 802.3x (Flow Control)
- 802.3ab (GE)
- 802.3z (Gigabit Ethernet including CWDM)
- 802.3ad (LACP)
- 802.1p (Layer 2 QoS)

- 802.1q
- 802.1X (Authentication)
- 802.1D (Spanning Tree Protocol)
- 802.3af (Inline power)
- PAgP
- VTP
- Dynamic ARP Inspection
- DHCP snooping
- IP source guard
- IGMP snooping (versions 1 and 2)
- DTP (802.1q and ISL)
- MST
- PVST+
- Rapid-PVST
- PortFast/UplinkFast/BackboneFast
- BPDU guard and filtering
- Voice VLAN
- Port security
- Unicast MAC filtering
- ACL (VACLS, PACLS, RACLS)
- QOS (DBL)
- Multicast storm control/broadcast storm control

SSO is compatible with the following list of features. However, the protocol database for these features is not synchronized between the redundant and active supervisor engines:

- 802.1Q tunneling with Layer 2 Protocol Tunneling (L2PT)
- Baby giants
- Jumbo frame support
- CDP
- Flood blocking
- UDLD
- SPAN/RSPAN
- NetFlow

The following features are learned on the redundant supervisor engine if the SSO feature is enabled:

- All Layer 3 protocols on Catalyst 4500 series switches (Switch Virtual Interfaces)

Understanding Supervisor Engine Redundancy Synchronization

During normal operation, the persistent configuration (RPR and SSO) and the running configuration (SSO only) are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.

Note You cannot enter CLI commands on the redundant supervisor engine console.

These sections describe supervisor engine redundancy synchronization:

- RPR Supervisor Engine Configuration Synchronization, page 8-5
- SSO Supervisor Engine Configuration Synchronization, page 8-5

RPR Supervisor Engine Configuration Synchronization

Because the redundant supervisor engine is only partially initialized in RPR mode, it interacts with the active supervisor engine only to receive configuration changes at startup and upon saving the configuration changes.

When a redundant supervisor engine is running in RPR mode, the following events trigger synchronization of the configuration information:

- When the redundant supervisor engine boots, the **auto-sync** command synchronizes the persistent configuration. This command is enabled by default. For details, refer to "Synchronizing the Supervisor Engine Configurations" section on page 8-11.
- When the active supervisor engine detects the redundant supervisor engine, the configuration information is synchronized from the active supervisor engine to the redundant supervisor engine. This synchronization overwrites any existing startup configuration file on the redundant supervisor engine.
- When you make changes to the configuration, you must use the **write** command to save and synchronize the startup configuration of the redundant supervisor engine.

SSO Supervisor Engine Configuration Synchronization

Note Stateful Switchover is *not* supported on Supervisor Engine 6-E.

When a redundant supervisor engine runs in SSO mode, the following events trigger synchronization of the configuration information:

- When the active supervisor detects the redundant supervisor engine, synchronization of the persistent and running configuration takes place, allowing the redundant supervisor engine to arrive at a fully-initiated state.
- When real-time changes occur, the active supervisor engine synchronizes the running-config and (or) the persistent configuration (if necessary) with the redundant supervisor engine.
- When you change the configuration, you must use the **write** command to allow the active supervisor engine to save and synchronize the startup configuration of the redundant supervisor engine.

Supervisor Engine Redundancy Guidelines and Restrictions

The following guidelines and restrictions apply to supervisor engine redundancy:

- RPR requires Cisco IOS Release 12.1(12c)EW, Release 12.1(19)E or later releases. SSO requires Cisco IOS Release 12.2(20)EWA or later releases.
- The Catalyst 4507R switch and the 4510R switch are the only Catalyst 4500 series switches that support supervisor engine redundancy.
- The Catalyst 4510R series switch only supports the WS-X4516 and WS-X4516-10GE supervisor engines. The Catalyst 4507R series switch supports supervisor engines WS-X4013+, WS-X4013+10GE, WS-X4515, WS-X4516, and WS-X4516-10GE.
- In Cisco IOS Release 12.2(25)SG and later releases on a Catalyst 4507R series switch, TenGigabit Ethernet and Gigabit Ethernet uplinks are concurrently usable on the Supervisor Engine V-10GE (WS-X4516-10GE) and the Supervisor Engine II+10GE (WS-4013+10GE). In Cisco IOS releases earlier than 12.2(25)SG, you need to use the **hw-module uplink select** configuration command to select either the TenGigabit Ethernet or Gigabit Ethernet uplinks.
- In Cisco IOS Release 12.2(25)SG and later releases, when using a Supervisor Engine V-10GE (WS-X4516-10GE) on a Catalyst 4510R series switch, you can select to use both the TenGigabit Ethernet and Gigabit Ethernet uplinks concurrently, but only with a WS-X4302-GB in slot 10. If either the 10 Gigabit Ethernet or Gigabit Ethernet uplinks are selected, then any linecard is allowed in slot 10. To select the uplinks, use the **hw-module uplink select** configuration command. In Cisco IOS releases earlier than 12.2(25)SG, you cannot use the TenGigabit Ethernet and Gigabit Ethernet uplinks concurrently.
- When you select TenGigabit Ethernet uplinks on WS-X4516-10GE and WS-X4013+10GE Supervisor Engines in RPR or SSO mode, only TenGigabitEthernet 1/1 and 2/1 interfaces are available. Similarly, when you select Gigabit Ethernet uplinks, only GigabitEthernet 1/3, 1/4, 2/3, and 2/4 interfaces are available. When you select to use both uplinks concurrently, TenGigabitEthernet 1/1 and 2/1 interfaces and GigabitEthernet 1/3, 1/4, 2/3, and 2/4 interfaces are available.
- Redundancy requires both supervisor engines in the chassis to be of the same supervisor engine model and to use the same Cisco IOS software image.
- When you use the WS-X4013+ and WS-X4515 supervisor engines in RPR or SSO mode, only the Gig1/1 and Gig2/1 Gigabit Ethernet interfaces are available, but the Gig1/2 and Gig2/2 uplink ports are unavailable.
- When the WS-X4516 active and redundant supervisor engines are installed in the same chassis, the four uplink ports (Gig1/1, Gig2/1, Gig 1/2, and Gig2/2) are available.
- The active and redundant supervisor engines in the chassis must be in slots 1 and 2.
- Each supervisor engine in the chassis must have its own Flash device and console port connections to operate the switch on its own.
- Each supervisor engine must have a unique console connection. Do not connect a Y cable to the console ports.
- Supervisor engine redundancy does not provide supervisor engine load balancing.
- The Cisco Express Forwarding (CEF) table is cleared on a switchover. As a result, routed traffic is interrupted until route tables reconverge. This reconvergence time is minimal because the SSO feature reduces the supervisor engine redundancy switchover time from 30+ seconds to subsecond, so Layer 3 also has a faster failover time if the switch is configured for SSO.

- Static IP routes are maintained across a switchover because they are configured from entries in the configuration file.
- Information about Layer 3 dynamic states that is maintained on the active supervisor engine is not synchronized to the redundant supervisor engine and is lost on switchover.
- Starting with Cisco IOS Release 12.2, if an unsupported condition is detected (such as when the active supervisor engine is running Cisco IOS Release 12.2(20)EW and the redundant supervisor engine is running Cisco IOS Release 12.1(20)EW), the redundant supervisor engine will be reset multiple times and then be placed in ROMMON mode. Therefore, it is important to follow the exact procedures outlined in the "Performing a Software Upgrade" section on page 8-13.
- If you are running (or upgrading to) Cisco IOS Release 12.2(20)EWA or Cisco IOS Release 12.2(25)EW and are using a single supervisor engine in a redundant chassis (Catalyst 4507R or Catalyst 4510R series switch), and you intend to use routed ports, do one of the following:
  - Use SVI's instead of routed ports.
  - Change the redundancy mode from SSO to RPR.
- Configuration changes made to the redundant supervisor engine through SNMP synchronization and SNMP set operations in SSO mode are not synchronized to the redundant supervisor engine. Even though you can still perform SNMP set operations in SSO mode, you might experience unexpected behaviour.

  After you configure the switch through SNMP in SSO mode , 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. Then, reload the redundant supervisor engine so that new configuration is applied on the redundant supervisor engine.
- You cannot perform configuration changes during the startup (bulk) synchronization. If you attempt to make configuration changes during this process, the following message is generated:

  ```
  Config mode locked out till standby initializes
  ```
- If configuration changes occur at the same time as a supervisor engine switchover, these configuration changes are lost.

Configuring Supervisor Engine Redundancy

These sections describe how to configure supervisor engine redundancy:

Configuring Redundancy

To configure redundancy, perform this task:

	Command	Purpose
Step 1	`Switch(config)# redundancy`	Enters redundancy configuration mode.
Step 2	`Switch(config-red)# mode {sso | rpr}`	Configures SSO or RPR. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO or RPR mode.
Step 3	`Switch# show running-config`	Verifies that SSO or RPR is enabled.
Step 4	`Switch# show redundancy [clients | counters | history | states]`	Displays the redundancy information (counter, state, and so on) for the active and redundant supervisor engines.

When configuring redundancy, note the following:

- The **sso** keyword is supported in Cisco IOS Release 12.2(20)EWA and later releases.
- The **rpr** keyword is supported in Cisco IOS Release 12.1(12c)EW and later releases.

This example shows how to configure the system for SSO and display the redundancy facility information:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# redundancy
Switch(config-red)# mode sso
Switch(config-red)# end
Switch# show redundancy
Redundant System Information :
------------------------------
       Available system uptime = 2 days, 2 hours, 39 minutes
Switchovers system experienced = 0
              Standby failures = 0
        Last switchover reason = none

                 Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
     Operating Redundancy Mode = Stateful Switchover
              Maintenance Mode = Disabled
                Communications = Up

Current Processor Information :
-------------------------------
               Active Location = slot 1
        Current Software state = ACTIVE
       Uptime in current state = 2 days, 2 hours, 39 minutes
                 Image Version = Cisco Internetwork Operating System Software
IOS (tm) Catalyst 4000 L3 Switch Software (cat4000-I5S-M), Version 12.2(20)EWA(3
.92), CISCO INTERNAL USE ONLY ENHANCED PRODUCTION VERSION
Copyright (c) 1986-2004 by cisco Systems, Inc.
Compiled Wed 14-Jul-04 04:42 by esi
                          BOOT = bootflash:cat4000-i5s-mz.122_20_EWA_392,1
        Configuration register = 0x2002

Peer Processor Information :
----------------------------
              Standby Location = slot 2
        Current Software state = STANDBY HOT
       Uptime in current state = 2 days, 2 hours, 39 minutes
                 Image Version = Cisco Internetwork Operating System Software
IOS (tm) Catalyst 4000 L3 Switch Software (cat4000-I5S-M), Version 12.2(20)EWA(3
.92), CISCO INTERNAL USE ONLY ENHANCED PRODUCTION VERSION
Copyright (c) 1986-2004 by cisco Systems, Inc.
Compiled Wed 14-Jul-04 0
                          BOOT = bootflash:cat4000-i5s-mz.122_20_EWA_392,1
        Configuration register = 0x2002

Switch#
```

This example shows how to display redundancy facility state information:

```
Switch# show redundancy states
my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
           Unit = Primary
        Unit ID = 2

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
     Split Mode = Disabled
   Manual Swact = Enabled
 Communications = Up

   client count = 21
 client_notification_TMR = 240000 milliseconds
          keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 18
           RF debug mask = 0x0
Switch#
```

This example shows how to change the system configuration from RPR to SSO mode:

```
Switch(config)# redundancy
Switch(config-red)# mode
Switch(config-red)# mode sso
Changing to sso mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
Switch#
*Aug  1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug  1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost
```

This example shows how to change the system configuration from SSO to RPR mode:

```
Switch(config)# redundancy
Switch(config-red)# mode rpr
Changing to rpr mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
*Aug  1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug  1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost
```

Virtual Console for Standby Supervisor Engine

Catalyst 4500 series switches can be configured with 2 supervisor engines to provide redundancy. When the switch is powered, one of the supervisor engines becomes active and remains active until a switchover occurs. The other supervisor engine remains in standby mode.

Each supervisor engine has its own console port. Access to the standby supervisor engine is possible only through the console port of the standby supervisor engine. Therefore, you must connect to the standby console to access, monitor or debug the standby supervisor.

Virtual Console for Standby Supervisor Engine enables you to access the standby console from the active supervisor engine without requiring a physical connection to the standby console. It uses IPC over EOBC to communicate with the standby supervisor engine and thus emulate the standby console on the active supervisor engine. Only one active standby console session is active at any time.

The Virtual Console for Standby Supervisor Engine allows users who are logged onto the active supervisor engine to remotely execute show commands on the standby supervisor engine and view the results on the active supervisor engine. Virtual Console is available only from the active supervisor engine.

You can access the standby virtual console from the active supervisor engine with the **attach module**, **session module**, or **remote login** commands on the active supervisor engine. You must be in privilege EXEC mode (level 15) to run these commands to access the standby console.

Once you enter the standby virtual console, the terminal prompt automatically changes to "<hostname>-standby-console#" where hostname is the configured name of the switch. The prompt is restored back to the original prompt when you exit the virtual console.

You exit the virtual console with the **exit** or **quit** commands. When the inactivity period of the terminal on the active supervisor engine where you logged in exceeds the configured idle time, you are automatically logged out of the terminal on the active supervisor engine. In such a case, the virtual console session is also terminated. Virtual console session is also automatically terminated when the standby is rebooted. After the standby boots up, you need to create another virtual console session.

To login to the standby supervisor engine using a virtual console, do the following:

```
Switch# session module 2
Connecting to standby virtual console
Type "exit" or "quit" to end this session

Switch-standby-console# exit
Switch#
```

If the standby console is not enabled, the following message appears.

```
Switch-standby-console#
Standby console disabled.
Valid commands are: exit, logout
```

Note The standby virtual console provides the standard features that are available from the supervisor console such as command history, command completion, command help and partial command keywords.

The following limitations apply to the standby virtual console:

- All commands on the virtual console run to completion. It does not provide the auto-more feature; it behaves as if the **terminal length 0** command has been executed. It is also non-interactive. Therefore, a running command cannot be interrupted or aborted by any key sequence on the active supervisor engine. Therefore if a command produces considerable output, the virtual console displays it on the supervisor screen.
- The virtual console is non-interactive. Because the virtual console does not detect the interactive nature of a command, any command that requires user interaction causes the virtual console to wait until the RPC timer aborts the command.

The virtual console timer is set to 60 seconds. The virtual console returns to its prompt after 60 seconds. During this time, you cannot abort the command from the key board. You must wait for the timer to expire before you continue.

- You cannot use virtual console to view debug and syslog messages that are being displayed on the standby supervisor engine. The virtual console only displays the output of commands that are executed from the virtual console. Other information that is displayed on the real standby console does not appear on the virtual console.

Synchronizing the Supervisor Engine Configurations

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

	Command	Purpose
Step 1	`Switch(config)#` **`redundancy`**	Enters redundancy configuration mode.
Step 2	`Switch(config-red)#` **`main-cpu`**	Enters main-cpu configuration submode.
Step 3	`Switch(config-r-mc)#` **`auto-sync`** `{`**`startup-config`** `|` **`config-register`** `|` **`bootvar`** `|` **`standard`**`}`	Synchronizes the configuration elements.
Step 4	`Switch(config-r-mc)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Switch#` **`copy running-config startup-config`**	Synchronizes the running configuration in dynamic random-access memory (DRAM) to the startup configuration file in NVRAM. **Note** This step is not required to synchronize the running configuration file in (DRAM).

Note Configuration changes made to the active supervisor engine through SNMP are not synchronized to the redundant supervisor engine. For information on how to handle this situation, see the "Supervisor Engine Redundancy Guidelines and Restrictions" section on page 8-6.

Note The **auto-sync** command controls the synchronization of the config-reg, bootvar, and startup/private configuration files only. The calendar and VLAN database files are always synchronized when they change. In SSO mode, the running-config is always synchronized.

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. Updates for the boot variables are automatic and cannot be disabled.

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

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

Note When you configure the auto-sync standard, the individual sync options such as no auto-sync startup-config are ignored.

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

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# no auto-sync standard
Switch(config-r-mc)# auto-sync config-register
Switch(config-r-mc)# end
```

Performing a Manual Switchover

This section describes how to perform a manual switchover (from the active supervisor engine to the redundant supervisor engine) for test purposes. We recommend that you perform a manual switchover prior to deploying SSO in your production environment.

Note This discussion assumes that SSO has been configured as the redundant mode.

To perform a manual switchover, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	`Switch# show redundancy`	Verifies that the peer state is in the STANDBY HOT state. See the example of the **show redundancy states** command on page 6-10.
Step 2	`Switch# redundancy force-switchover`	Launches switchover from the active supervisor engine to the redundant supervisor engine. If the state of the redundant supervisor engine is not standby hot, this command will not execute.

Be aware of these usage guidelines:

- To force a switchover, the redundant supervisor engine must be in a standby hot state. You can verify the state with the **show redundancy** command. If the state is not standby hot, the **redundancy force-switchover** command will not execute.
- Use the **redundancy force-switchover** command, rather than the **reload** command, to initiate a switchover. The **redundancy force-switchover** command will first check that the redundant supervisor engine is in the correct state. If you issue the **reload** command and the status is not standby hot, the **reload** command will reset the current supervisor engine only.

After an initial switchover, there might be occasions when you want to make the supervisor engine in slot 1 of the chassis the active supervisor engine. If the image on supervisor engine 1 is the one you intend to run on both supervisor engines, it is not necessary to re-boot the image on the supervisor engine in slot 1 to make it redundant. Instead, you can force another switchover. However, if you want a newer version of the image to run on both supervisor engines, follow the steps under “Performing a Software Upgrade” on page 13. Use the **show module** command to see which slot contains the active supervisor engine, and force another switchover if necessary.

Performing a Software Upgrade

The software upgrade procedure supported by supervisor engine redundancy allows you to reload the Cisco IOS software image on the redundant supervisor engine, and once complete, reload the active supervisor engine once.

If the active supervisor engine is running Cisco IOS Release 12.2(x)S, the standby supervisor engine cannot run Cisco IOS Release 12.1(x)E. This would reset the switch immediately after the system boot of the standby supervisor engine. The reverse configuration, where the standby engine is running Cisco IOS Release 12.2(x)S and the active supervisor engine is running Cisco IOS Release 12.1(x)E, is fully supported.

To perform a software upgrade, perform this task:

	Command	Purpose
Step 1	Switch# **copy** *source_device***:***source_filename* **slot0:***target_filename* Or: Switch# **copy** *source_device***:***source_filename* **bootflash:***target_filename*	Copies the new Cisco IOS software image to bootflash on the supervisor engine.
Step 2	Switch# **copy** *source_device***:***source_filename* **slaveslot0:***target_filename* Or: Switch# **copy** *source_device***:***source_filename* **slavebootflash:***target_filename*	Copies the new image to a slave device (such as slavebootflash and slaveslot0).
Step 3	Switch# **config terminal** Switch(config)# **config-register 0x2** Switch(config)# **boot system flash** *device*:*file_name*	Configures the supervisor engines to boot the new image. If your system was configured to autoboot an earlier image, issue the following command string to boot the new image instead: **no boot system flash** *device*:*old_file_name*
Step 4	Switch(config)# **redundancy**	Enters redundancy configuration mode.
Step 5	Switch(config-red)# **main-cpu**	Enters main-cpu configuration submode.
Step 6	Switch(config-r-mc)# **auto-syn standard**	Synchronizes the configuration elements.
Step 7	Switch(config-r-mc)# **end**	Returns to privileged EXEC mode.
Step 8	Switch# **copy running-config start-config**	Saves the configuration.

	Command	Purpose
Step 9	`Switch# redundancy reload peer`	Reloads the redundant supervisor engine and brings it back online (using the new release of the Cisco IOS software). **Note** Before proceeding to Step 10, ensure that the switch is operating in RPR mode.
Step 10	`Switch# redundancy force-switchover`	Conducts a manual switchover to the redundant supervisor engine. The redundant supervisor engine becomes the new active supervisor engine using the new Cisco IOS software image. The old active supervisor engine reboots with the new image and becomes the redundant supervisor engine.

This example shows how to perform a software upgrade:

```
Switch# config terminal
Switch(config)# config-register 0x2
Switch(config)# boot system flash slot0:cat4000-i5s-mz.122-20.EWA
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-syn standard
Switch(config-r-mc)# end
Switch# copy running-config start-config
Switch# redundancy reload peer
Switch# redundancy force-switchover
Switch#
```

This example illustrates how to verify that the running configuration on the active supervisor engine has successfully synchronized with the redundant supervisor engine:

```
Switch# config terminal
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The bootvar has been successfully synchronized to the
standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The config-reg has been successfully synchronized to
the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The startup-config has been successfully synchronized
to the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The private-config has been successfully synchronized
to the standby supervisor
```

The example above shows that the boot variable, the config-register, and the startup configuration from the active supervisor engine have successfully synchronized to the redundant supervisor engine.

Manipulating Bootflash on the Redundant Supervisor Engine

Note The console port on the redundant supervisor engine is not available.

To manipulate the redundant supervisor engine bootflash, perform one or more of the following tasks:

Command	Purpose

`Switch# dir slaveslot0:target_filename`
or:
`Switch# dir slavebootflash:target_filename`
	Lists the contents of the **slot0:** device on the redundant supervisor engine.
	Lists the contents of the **bootflash:** device on the redundant supervisor engine.

`Switch# delete slaveslot0:target_filename`
or:
`Switch# delete slave bootflash:target_filename`
	Deletes specific files from the **slot0:** device on the redundant supervisor engine.
	Deletes specific files from the **bootflash:** device on the redundant supervisor engine.

`Switch# squeeze slaveslot0:target_filename`
or:
`Switch# squeeze slavebootflash:target_filename`
	Squeezes the **slot0:** device on the redundant supervisor engine.
	Squeezes the **bootflash:** device on the redundant supervisor engine.

`Switch# format slaveslot0:target_filename`
or:
`Switch# format slavebootflash:target_filename`
	Formats the **slot0:** device on the redundant supervisor engine.
	Formats the **bootflash:** device on the redundant supervisor engine.

`Switch# copy source_device:source_filename slaveslot0:target_filename`
or:
`Switch# copy source_device:source_filename slavebootflash:target_filename`
	Copies a file from the active supervisor engine to the **slot0:** device on the redundant supervisor engine.
	Copies a file to the **bootflash:** device on a redundant supervisor engine.
	Note Source could be the active supervisor engine or a TFTP server.

CHAPTER 9

Configuring Cisco 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 Nonstop Forwarding is *not* supported on Supervisor Engine 6-E.

This chapter consists of these sections:

- Understanding NSF with SSO Supervisor Engine Redundancy, page 9-1
- Configuring NSF with SSO Supervisor Engine Redundancy, page 9-9

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference*, Release 12.2(37)SG and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Understanding NSF with SSO Supervisor Engine Redundancy

These sections describe supervisor engine redundancy using NSF with SSO:

- Understanding Cisco IOS NSF-Aware and NSF-Capable Support, page 9-2
- NSF with SSO Supervisor Engine Redundancy Overview, page 9-3
- SSO Operation, page 9-4
- NSF Operation, page 9-4
- Cisco Express Forwarding, page 9-5
- Routing Protocols, page 9-5
- NSF Guidelines and Restrictions, page 9-9

Understanding Cisco IOS NSF-Aware and NSF-Capable Support

Cisco IOS Nonstop Forwarding (NSF) has two primary components:

NSF-awareness—If neighboring router devices detect that an NSF router can still forward packets when a supervisor engine switchover happens, this capability is referred to as NSF-awareness. Cisco IOS enhancements to the Layer 3 routing protocols (OSPF, BGP, EIGRP and IS-IS) are designed to prevent route-flapping so that the CEF routing table does not time out or the NSF router does not drop routes. An NSF-aware router helps to send routing protocol information to the neighboring NSF router.

NSF-capability—NSF works with SSO to minimize the amount of time that a Layer 3 network is unavailable following a supervisor engine switchover by continuing to forward IP packets. Reconvergence of Layer 3 routing protocols (BGP, EIGRP, OSPF v2, and IS-IS) is transparent to the user and happens automatically in the background. The routing protocols recover routing information from neighbor devices and rebuild the Cisco Express Forwarding (CEF) table.

Note NSF does not support VRF and IPv6.

Note NSF capable devices include Catalyst 4500 series switches, Catalyst 6500 series switches, Cisco 7500 series routers, Cisco 10000 series routers, and Cisco 12000 series routers.

A typical topology for NSF and NSF-aware routers is given below.

Figure 9-1 Topology for NSF and NSF-Capable Switches

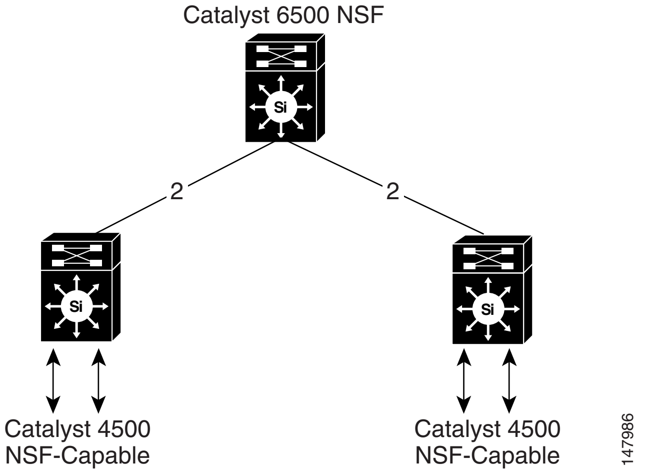


Table 9-1 lists the supervisor engines and Catalyst 4500 series switches that support NSF-awareness:

Table 9-1 NSF-Aware Supervisor Engines

NSF-Aware Supervisor Engine	Switch Support
Supervisor Engine II-Plus (WS-X4013)	Catalyst 4507R series switch, Catalyst 4506 series switch, and Catalyst 4503 series switch
Supervisor Engine II-Plus+TS (WS-X4013+TS)	Catalyst 4507R series switch, Catalyst 4506 series switch, and Catalyst 4503 series switch
Supervisor Engine II-Plus+10GE (WS-X4013+10GE)	Catalyst 4507R series switch, Catalyst 4506 series switch, and Catalyst 4503 series switch
Supervisor Engine IV (WS-X4515)	Catalyst 4507R series switch, Catalyst 4506 series switch, and Catalyst 4503 series switch
Supervisor Engine V (WS-X4516)	Catalyst 4507R series switch and Catalyst 4510R series switch
Supervisor Engine V-10GE (WS-X4516-10GE)	Catalyst 4506 series switch, Catalyst 4507R series switch, and Catalyst 4510R series switch
Fixed Switchs (WS-C4948 and WS-C4948-10GE)	Catalyst 4948 and 4948-10GE switches

Starting with Cisco IOS Release 12.2(20)EWA, the Catalyst 4500 series switch supported NSF-awareness for the EIGRP, IS-IS, OSPF and BGP protocols. Starting with Cisco IOS Release 12.2(31)SG, the Catalyst 4500 series switch supported NSF-awareness for the EIGRP-stub in IP Base image for all supervisor engines. NSF-awareness is turned on by default for EIGRP-stub, EIGRP, IS-IS and OSPF protocols. For BGP, you need to turned it on manually.

If the supervisor engine is configured for BGP (with the **graceful-restart** command), EIGRP, OSPF or IS-IS routing protocols, routing updates are automatically sent during the supervisor engine switchover of a neighboring NSF capable switch (typically a Catalyst 6500 series switch).

Starting with Cisco IOS Release 12.2(31)SG, the Catalyst 4500 series switch supports NSF-capability. Table 9-2 lists the supervisor engines and Catalyst 4500 series switches that support NSF-capable:

Table 9-2 NSF-Capable Supervisor Engines

NSF-Capable Supervisor Engine	Switch Support
Supervisor Engine IV (WS-X4515)	Catalyst 4507R series switch
Supervisor Engine V (WS-X4516)	Catalyst 4507R series switch and Catalyst 4510R series switch
Supervisor Engine V-10GE (WS-X4516-10GE)	Catalyst 4507R series switch and Catalyst 4510R series switch

NSF with SSO Supervisor Engine Redundancy Overview

Catalyst 4500 series switches support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. NSF works with SSO to minimize the amount of time a network is unavailable to its users following a switchover.

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, which were 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 during 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 during a switchover, routing flaps are avoided.
- Maintains user sessions established prior to the switchover

Catalyst 4500 series switches also support route processor redundancy (RPR). For information about these redundancy modes, see Chapter 1, “Configuring Supervisor Engine Redundancy Using RPR and SSO.”

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.

In networking devices running SSO, both supervisor engines must be running the same Cisco IOS software version and ROMMON version 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.

Note Be aware that you can use the [**no**] **service slave-log** configuration command to forward all error messages from the standby supervisor engine to the active engine. By default, this capability is enabled. For details, refer to the *Catalyst 4500 Series Switch Cisco IOS System Error Message Guide*, Release 12.2(37)SG.

NSF Operation

NSF always runs with SSO and provides redundancy for Layer 3 traffic. NSF is supported by the BGP, OSPF, IS-IS, and EIGRP routing protocols 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 rebuilds 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 forwarding engines, CEF keeps the forwarding engine on the redundant supervisor engine current with changes that are sent to it by CEF on the active supervisor engine. The forwarding engine can 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 cause prefix-by-prefix updates to CEF, which it uses to update the FIB and adjacency databases. Existing and new entries receive the new version (“epoch”) number, indicating that they have been refreshed. The forwarding information is updated on the 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.

Routing Protocols

Note **Use of the routing protocols require the Enterprise Services Cisco IOS Software image for the Catalyst 4500 series switch.**

The routing protocols run only on the active supervisor engine, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby 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. 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 capable of a graceful restart.

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 ignores the graceful restart capability in an OPEN message but establishes a BGP session with the NSF-capable device. This function allows interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers is not capable of a graceful restart.

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 does not establish an NSF-capable session with that neighbor. All other neighbors that have graceful restart capability 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 support in 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 disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware routers 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 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 aborts 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 attempts a second NSF restart. During this time, the new redundant supervisor engine boots up and synchronizes 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 attempts a second NSF restart. During this time, the new redundant supervisor engine boots up and synchronizes 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 is not 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 re-boots after 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 uses a new Restart (RS) bit in the EIGRP packet header to indicate a restart. The RS bit is 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 recognizes the restarting peer in its peer list and maintains the adjacency with the restarting router. The neighbor then sends it topology table to the restarting router with the RS bit set in the first update packet indicating that it is NSF-aware and is helping out the restarting router. The neighbor does not set the RS bit in their hello packets, unless it is also a NSF restarting neighbor.

Note A router may be NSF-aware but may not be helping the NSF restarting neighbor because booting 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 notifies the RIB of convergence. EIGRP waits for the RIB convergence signal and then floods its topology table to all awaiting NSF-aware peers.

NSF Guidelines and Restrictions

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; NSF with SSO does not support IPv6 traffic.
- 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.
- 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.

Configuring NSF with SSO Supervisor Engine Redundancy

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

- Configuring SSO, page 9-10

- Configuring CEF NSF, page 9-11
- Verifying CEF NSF, page 9-11
- Configuring BGP NSF, page 9-11
- Verifying BGP NSF, page 9-12
- Configuring OSPF NSF, page 9-13
- Verifying OSPF NSF, page 9-13
- Configuring IS-IS NSF, page 9-14
- Verifying IS-IS NSF, page 9-15
- Configuring EIGRP NSF, page 9-16
- Verifying EIGRP NSF, page 9-16

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	`Switch(config)# redundancy`	Enters redundancy configuration mode.
Step 2	`Switch(config-red)# mode sso`	Configures SSO. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO mode.
Step 3	`Switch(config-red)# end`	Returns to EXEC mode.
Step 4	`Switch# show running-config`	Verifies that SSO is enabled.
Step 5	`Switch# show redundancy states`	Displays the operating redundancy mode.

Note The **sso** keyword is supported in Cisco IOS Release 12.2(20)EWA and later releases.

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

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# redundancy
Switch(config-red)# mode sso
Switch(config-red)# end
Switch# 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
Switch#
```

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:

```
Switch# 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	`Switch# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 2	`Switch(config)#` **`router bgp`** *`as-number`*	Enables a BGP routing process, which places the switch in switch configuration mode.
Step 3	`Switch(config-router)#` **`bgp graceful-restart`**	Enables the BGP graceful restart capability, starting BGP NSF. If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor. Use this command on the restarting switch and all of its peers.

Verifying BGP NSF

To verify BGP NSF, you must check that BGP graceful restart 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 switch by entering the **show running-config** command:

```
Switch# 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, BGP NSF does not occur either:

```
Switch# 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 when you install an NSF software image on the device.

To configure OSPF NSF, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# router ospf processID`	Enables an OSPF routing process, which places the switch in router configuration mode.
Step 3	`Switch(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:

```
Switch# show running-config

route 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:

```
Switch> 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	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# router isis [tag]`	Enables an IS-IS routing process, which places the switch in router configuration mode.
Step 3	`Switch(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	`Switch(config-router)# nsf interval [minutes]`	(Optional) Specifies the minimum time between NSF restart attempts. The default time between *consecutive* NSF restart attempts is 5 minutes.
Step 5	`Switch(config-router)# nsf t3 {manual [seconds] | adjacency}`	(Optional) Specifies the time IS-IS waits 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 switch that is restarting obtains its wait time from neighboring devices.
Step 6	`Switch(config-router)# nsf interface wait seconds`	(Optional) Specifies how long an IS-IS NSF restart waits 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 shows 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:

```
Switch# 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 differs 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”:

```
Switch# 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”:

```
Switch# 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:

```
Switch# 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	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`router eigrp`** *`as-number`*	Enables an EIGRP routing process, which places the switch in router configuration mode.
Step 3	`Switch(config-router)#` **`nsf`**	Enables EIGRP NSF. Use this command on the “restarting” switch 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:

```
Switch# 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:

```
Switch# 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
```

CHAPTER 10

Environmental Monitoring and Power Management

Note Before reading this chapter, read the "Preparing for Installation” section of the *Catalyst 4500 Series Installation Guide*. It is important to ensure that your installation site has enough power and cooling to accommodate the additional electrical load and heat introduced by Power over Ethernet.

This chapter describes power management and environmental monitoring features in the Catalyst 4500 series switches. It provides guidelines, procedures, and configuration examples.

This chapter consists of the following major sections:

- Understanding Environmental Monitoring, page 10-1
- Power Management, page 10-6

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Understanding Environmental Monitoring

This section contains the following subsections:

- Using CLI Commands to Monitor your Environment, page 10-2
- Displaying Environment Conditions, page 10-2
- Emergency Actions, page 10-3
- System Alarms, page 10-4

Environmental monitoring of chassis components provides early warning indications of possible component failure. This warning helps you to ensure the safe and reliable operation of your system and avoid network interruptions.

This section describes how to monitor critical system components so that you can identify and rapidly correct hardware-related problems.

Using CLI Commands to Monitor your Environment

Use the **show environment** CLI command to monitor the system. This section gives a basic overview of the command and keywords you will need.

Enter the **show environment** [**alarm** | **status** | **temperature**] command to display system status information. Keyword descriptions are listed in Table 10-1.

Table 10-1   show environment Keyword Descriptions

Keyword	Purpose
alarm	Displays environmental alarms for the system.
status	Displays field-replaceable unit (FRU) operational status and power and power supply fan sensor information.
temperature	Displays temperature of the chassis.

Displaying Environment Conditions

Topics include:

- Conditions on Supervisor Engines II-Plus to V-10GE, page 10-2
- Conditions on Supervisor Engine 6-E, page 10-2

Conditions on Supervisor Engines II-Plus to V-10GE

The following example shows how to display the environment condition on Supervisor Engines II-Plus to V-10GEs. The output indicates that the power supplies differ. The switch uses only one power supply and disables the other.

```
Switch# show environment
no alarm

Chassis Temperature                    = 35 degrees Celsius
Chassis Over Temperature Threshold     = 75 degrees Celsius
Chassis Critical Temperature Threshold = 95 degrees Celsius

Power                                             Fan     Inline
Supply  Model No          Type       Status       Sensor  Status
------  ----------------  ---------  -----------  ------  ------
PS1     PWR-C45-2800AC    AC 2800W   good         good    good
➔ PS2     PWR-C45-1000AC    AC 1000W   err-disable  good    n.a.

*** Power Supplies of different types have been detected***
Switch#
```

Conditions on Supervisor Engine 6-E

Supervisor Engine 6-E and its associated linecards support multiple temperature sensors per card. The environment condition output includes the temperature reading from each sensor and the temperature thresholds for each sensor. These linecards support three thresholds: warning, critical, and shutdown. (Supervisor Engines II-Plus to V-10GE support two threshold.)

The following example illustrates how to display the environment condition on a Supervisor Engine 6-E. The thresholds appear within parentheses.

```
Switch# show environment
no temperature alarms


Module Sensor                     Temperature          Status
------+---------------------------+--------------------+-----------
2      air inlet                  23C (51C,65C,68C)              ok
2      air outlet                 29C (69C,83C,86C)              ok
5      air inlet                  38C (51C,65C,68C)              ok
5      air outlet                 38C (69C,83C,86C)              ok
6      air inlet                  34C (51C,65C,68C)              ok
6      air outlet                 37C (69C,83C,86C)              ok

Power                                             Fan      Inline
Supply  Model No          Type       Status       Sensor   Status
------  ----------------  ---------  -----------  -------  -------
PS1     PWR-C45-2800AC    AC 2800W   good         good     good
PS2     none              --         --           --       --

Power supplies needed by system    : 1
Power supplies currently available : 1

Chassis Type : WS-C4510R-E

Power consumed by backplane : 40 Watts

Switch Bandwidth Utilization : 0%

Supervisor Led Color : Green

Module  2 Status Led Color  : Green
Module  5 Status Led Color  : Green
Module  6 Status Led Color  : Orange
Module 10 Status Led Color  : Green

Fantray : Good

Power consumed by Fantray : 80 Watts
```

Emergency Actions

Chassis with Supervisor Engine 6-E can power down a single card, providing a detailed response to over-temperature conditions on linecards. However, Supervisor Engine 6-E *cannot* safely operate when the temperature of the supervisor itself exceeds the critical threshold. Therefore, the supervisor will turn off the chassis’ power supplies to protect itself from overheating. When this happens, you can recover the switch only by cycling the power on/off switches on the power supplies or by cycling the AC or DC inputs to the power supplies.

Critical and shutdown temperature emergencies trigger the same action. Therefore, the following table (Table 10-2) lists temperature emergencies but does not distinguish between critical and shutdown emergencies.

Table 10-2 Emergency and Action for Supervisor Engines 6-E

Case 1. Complete fan failure emergency.	Power down the chassis.
Case 2. Temperature emergency on a linecard.	Power down the linecard.
Case 3. Temperature emergency on the standby supervisor engine.	Power down the standby supervisor.
Case 4. Temperature emergency on the active supervisor engine with the standby supervisor engine in the hot standby or cold standby redundancy state.	Reset the active supervisor engine.
Case 5. Temperature emergency on the active supervisor engine with no standby supervisor engine or with a standby supervisor engine that is not in hot standby or cold standby redundancy state.	Power down the chassis.

In Case 4, the standby supervisor engine takes over when the active engine resets itself. Then, if the temperature emergency remains, the newly-active supervisor engine resets the now-standby supervisor engine.

Case 5 applies to nonredundant chassis and to chassis with a standby supervisor engine that has been shutdown or which has not fully booted.

System Alarms

Any system has two types of alarms: major and minor. A major alarm indicates a critical problem that could lead to system shutdown. A minor alarm is informational—it alerts you to a problem that could become critical if corrective action is not taken.

Table 10-3 lists the possible environment alarms.

Table 10-3 Possible Environmental Alarms

A temperature sensor over its warning threshold	minor
A temperature sensor over its critical threshold	major
A temperature sensor over its shutdown threshold	major
A partial fan failure	minor
A complete fan failure	major

Fan failure alarms are issued as soon as the fan failure condition is detected and are canceled when the fan failure condition clears. Temperature alarms are issued as soon as the temperature reaches the threshold temperature and are canceled when the temperature drops more than 5 degree C below the threshold. 5 degree C is a hysteresis value designed to prevent toggling alarms.

An LED on the supervisor engine indicates whether an alarm has been issued.

When the system issues a major alarm, it starts a timer whose duration depends on the alarm. If the alarm is not canceled before the timer expires, the system takes emergency action to protect itself from the effects of overheating. The timer values and the emergency actions depend on the type of supervisor

Note Refer to the *Catalyst 4500 Series Switch Module Installation Guide* for information on LEDs, including the startup behavior of the supervisor engine system LED.

Table 10-4 describes the alarms on Supervisor Engines II-Plus to V-10GE.

Table 10-4 Alarms on Supervisor Engines II-Plus to V-10GE

Event	Alarm Type	Supervisor LED Color	Timeout	Description and Action
Chassis temperature exceeds the critical threshold.	Major	Red	5 min	Syslog message displays when the alarm is issued. linecards are put in reset when the timeout expires.
Supervisor fails power on self-test (POST).	Major	Red	—	Syslog message is displayed. The supervisor fails to come up.
Chassis fan tray fails.	Major	Red	4 min	Syslog message displays when the alarm is issued. linecards are put in reset when the timeout expires.
Chassis temperature exceeds the warning threshold.	Minor	Orange	—	Syslog message displays when the alarm is issued.
Chassis fan tray experiences partial failure.	Minor	Orange	—	Syslog message displays when the alarm is issued.

Table 10-5 describes the alarms on Supervisor Engine 6-E.

Table 10-5 Alarms on Supervisor Engine 6-E

Event	Alarm Type	Supervisor LED Color	Timeout	Description and Action
Card temperature exceeds the critical threshold.	Major	Red	15 min	Syslog message displays when the alarm is issued. SeeTable 10-2 for the action on timeout.
Card temperature exceeds the shutdown threshold.	Major	Red	30 sec	Syslog message displays when the alarm is issued. SeeTable 10-2 for the action on timeout.
Supervisor fails power-on self-test (POST).	Major	Red	—	Syslog message displays. Supervisor fails to come up.
Chassis fan tray fails.	Major	Red	30 sec	Syslog message displays when the alarm is issued. SeeTable 10-2 for the action on timeout.

Table 10-5 Alarms on Supervisor Engine 6-E

Event	Alarm Type	Supervisor LED Color	Timeout	Description and Action
Chassis temperature exceeds the warning threshold.	Minor	Orange	—	Syslog message when the alarm is issued.
Chassis fan tray experiences partial failure.	Minor	Orange	—	Syslog message when the alarm is issued.

Power Management

This section describes the power management feature in the Catalyst 4500 series switches. It includes the following topics:

- Power Management for the Catalyst 4500 Series Switches, page 10-6
- Powering Down a Module, page 10-20
- Power Management for the Catalyst 4948 Switches, page 10-20

Note For power consumption of all Catalyst 4000/4500 family modules, see *"Appendix A, Specifications,"* in the *Catalyst 4500 Series Module Installation Guide*. Enter the **show power** command to display the current power redundancy and the current system power usage.

Power Management for the Catalyst 4500 Series Switches

This section includes the following subsections:

- Supported Power Supplies, page 10-6
- Power Management Modes for the Catalyst 4500 Switch, page 10-8
- Selecting a Power Management Mode, page 10-8
- Power Management Limitations in Catalyst 4500 Series Switches, page 10-9
- Available Power for Catalyst 4500 Series Switches Power Supplies, page 10-13
- Insufficient Inline Power Handling for Supervisor Engine II-TS, page 10-18
- Combined Mode Power Resiliency, page 10-15
- Special Considerations for the 1400 W DC Power Supply, page 10-16
- Special Considerations for the 1400 W DC SP Triple Input Power Supply, page 10-17
- Insufficient Inline Power Handling for Supervisor Engine II-TS, page 10-18
- Power Management Modes for the Catalyst 4948 Switch, page 10-20

Supported Power Supplies

You can select from several different power supplies to ensure that you have enough power for the modules installed in your switch.

Note You should select a power supply based on the modules and the amount of PoE desired using the Cisco Power Calculator. The choice between 1000AC and 1400AC should depend on the type of linecards that the customer plans to use in the chassis.

The Catalyst 4500 series switches support the following power supplies:

- Fixed Wattage—These power supplies always deliver a fixed amount of PoE and system power.
  - 1000 W AC—Supports up to 1050 W of system power. (Not recommended on the Catalyst 4510R switch, PoE not supported)
  - 1400 W AC—Supports up to 1400 W system power. (PoE not supported)
  - 2800 W AC—Supports up to 1400 W of system power and up to 1400 W of PoE.
- Variable Wattage—These power supplies automatically adjust the wattage to accommodate PoE and system power requirements.
  - 1300 W AC—Supports up to 1050 W of system power and 800 W of PoE, limited to a total of 1300 W.
  - 1400 W DC—Supports up to 1400 W of system power and variable amounts of PoE, depending on the input feed to the power supply. See "Special Considerations for the 1400 W DC Power Supply" section on page 10-16 for more information.
  - 1400 W DC Service Provider—Uses up to three lines (12.5 A, 15 A, 15 A) of DC input and delivers varying amounts of system power ranging from 400 W to 1400 W depending on the lines powered. See "Special Considerations for the 1400 W DC SP Triple Input Power Supply" section on page 10-17 for more information. (PoE not supported)
  - 4200 W AC—Supports varying amounts of system power and PoE depending on the number of inputs powered and input voltage.

Note All Catalyst 4500 series switch AC-input power supplies require single-phase source AC. The source AC can be out of phase between multiple power supplies or multiple AC-power plugs on the same power supply because all AC power supply inputs are isolated. Each chassis power supply should ideally have its own dedicated branch circuit sized to local and national codes.

When you insert power supplies in your switch, use power supplies that are of the same wattage. Multi-input power supplies such as 1400 W DC triple-input and 4200 W AC have additional restrictions. Read the sections on special considerations for these power supplies. If you mix power supplies, the switch uses the one with the lower wattage and ignores the other power supply. The power supply status displays as err-disable and the summary displays as all zeros (0) for wattage values in the output for the **show power** command.

The following example shows the output for the **show power** command for mixed power supplies:

```
Switch# show power
Power                                             Fan     Inline
Supply  Model No          Type       Status       Sensor  Status
------  ----------------  ---------  -----------  ------  ------
PS1     PWR-C45-2800AC    AC 2800W   good         good    good
→ PS2     PWR-C45-1000AC    AC 1000W   err-disable  good    n.a.

*** Power Supplies of different type have been detected***

Power supplies needed by system    :1
Power supplies currently available :1
Power Summary                        Maximum
 (in Watts)              Used      Available
----------------------   ----      ---------
System Power (12V)        328           1360
Inline Power (-50V)         0           1400
Backplane Power (3.3V)     10             40
----------------------   ----
Total Used                338 (not to exceed Total Maximum Available = 750)
Switch#
```

Power Management Modes for the Catalyst 4500 Switch

The Catalyst 4500 series switches support two power management modes:

- Redundant mode—Redundant mode uses one power supply as a primary power supply and the second power supply as a back-up. If the primary power supply fails, the second power supply immediately supports the switch without any disruption in the network. Both power supplies must be the same wattage. A single power supply must have enough power to support the switch configuration.
- Combined mode—Combined mode uses the power from all installed power supplies to support the switch configuration power requirements. However, combined mode has no power redundancy. If a power supply fails, one or more modules might shut down.

Note On the Catalyst 4510R switch, the 1000 W AC power supply is not enough to support redundant mode for all possible configurations. It is able to support redundant mode for limited configurations that require less than 1050 W.

Note The 1400 W DC power supply supports combined mode for data power. It does not support combined mode for PoE power.

Selecting a Power Management Mode

By default, a switch is set to redundant mode. In the **show power** command, if the **power supplies needed by system** is 1, the switch is in redundant mode; if the **power supplies needed by system** is 2, the switch is in combined mode.

Your switch hardware configuration will dictate which power supply or supplies you should use. For example, if your switch configuration requires more power than a single power supply provides, use the combined mode. In combined mode, however, the switch has no power redundancy. Consider the following possibilities:

- The supervisor engine consumes 110 W, the fan boxes for the Catalyst 4503 switch consume 30 W each, the fan boxes for the Catalyst 4506 and Catalyst 4507 switches consume 50 W each, the backplane for the Catalyst 4503 and Catalyst 4506 switches consumes 10 W, and the backplane for the Catalyst 4507 switch consumes 40 W.
- 1000 W can support a fully loaded Catalyst 4503 switch with no powered device support.
- 1300 W can support a fully loaded Catalyst 4503 switch with Cisco powered devices.
- Each PoE port on a WS-X4148-RJ45V module requires 6.3 W. Five fully loaded WS-X4148-RJ45V modules in a switch comprise 240 ports. This configuration requires 1512 W of PoE, plus 300 W for the modules.

Power Management Limitations in Catalyst 4500 Series Switches

Limitation 1

It is possible to configure a switch that requires more power than the power supplies provide. The two ways you could configure a switch to exceed the power capabilities are as follows:

- The power requirements for the installed modules exceed the power provided by the power supplies.

  If you insert a single power supply and then set the switch to combined mode, the switch displays this error message:

  ```
  Insufficient power supplies present for specified configuration.
  ```

  This error message also displays in the output for the **show power** command. This error message displays because, by definition, combined mode requires that two working power supplies be installed in your switch.

  If the power requirements for the installed modules exceeds the power provided by the power supplies, the switch displays this error message:

  ```
  Insufficient power available for the current chassis configuration.
  ```

  This error message also appears in the **show power** command output.

  If you attempt to insert additional modules into your switch and exceed the power supply, the switch immediately places the newly inserted module into reset mode, and the switch displays these error messages:

  ```
  Module has been inserted
  Insufficient power supplies operating.
  ```

  Additionally, if you power down a functioning switch and insert an additional module or change the module configuration so that the power requirements exceed the available power, one or more modules enter reset mode when you power on the switch again.

- The power requirements for the PoE exceed the PoE provided by the power supplies.

  If you have too many IP phones drawing power from the system, power to IP phones is cut, and some phones may be powered down to reduce the power requirements to match the power supplies.

In the first scenario (power requirements exceed the power supplied), the system attempts to resolve this power usage limitation by evaluating the type and number of modules installed. During the evaluation cycle, beginning from the bottom of the chassis, the system puts the modules that it is unable to support (for lack of power) into reset mode. The supervisor engine and modules for which there is adequate power always remain enabled, with no disruption of network connectivity. Modules placed in reset mode still consume some power and can be removed from the chassis to further reduce power requirements. If you configure the chassis correctly, the system will not enter the evaluation cycle.

A module in reset mode continues to draw power as long as it is installed in the chassis; you can use the **show power module** command to determine how much power is required to bring the module online.

To compute the power requirements for your system and verify that your system has enough power, add the power consumed by the supervisor engine module(s), the fan box(es), and the installed modules (including PoE). For PoE, total the requirements for all the phones. See the “Powering Down a Module” section on page 10-20 for more information on the power consumption for the various components of your switch.

The 802.3af-compliant PoE modules can consume up to 20 W of PoE to power FPGAs and other hardware components on the module. Be sure to add at least 20 W to your PoE requirements for each 802.3af-compliant PoE module to ensure that the system has adequate power for the PDs connected to the switch.

On the WS-X4148-RJ45V PoE module, PoE consumption cannot be measured. Therefore, for all PoE calculations, the PoE consumption on this module is presumed to be equal to its administrative PoE.

You can use the **show module** command to verify which modules are active and which, if any, have been placed in reset.

The following example shows the **show module** command output for a system with inadequate power for all installed modules. The system does not have enough power for Module 5; the *Status* displays it as *PwrDeny.*

If the PoE that is consumed by the module is more than 50 W above the PoE you allocated using the **power inline consumption default** command, the Status displays as PwrOver. If the PoE consumed by the module is more than 50 W above the PoE module limit, the Status displays as PwrFault.

```
Switch# show module
Mod  Ports Card Type                              Model             Serial No.
----+-----+--------------------------------------+-----------------+-----------
 1      2  1000BaseX (GBIC) Supervisor(active)    WS-X4014          JAB054109GH
 2      6  1000BaseX (GBIC)                       WS-X4306          00000110
 3     18  1000BaseX (GBIC)                       WS-X4418          JAB025104WK
➔ 5     0  Not enough power for module            WS-X4148-FX-MT    00000000000
 6     48  10/100BaseTX (RJ45)                    WS-X4148          JAB023402RP

 M MAC addresses                    Hw  Fw           Sw               Status
--+--------------------------------+---+------------+----------------+---------
 1 005c.9d1a.f9d0 to 005c.9d1a.f9df 0.5 12.1(11br)EW 12.1(20020313:00 Ok
 2 0010.7bab.9920 to 0010.7bab.9925 0.2                               Ok
 3 0050.7356.2b36 to 0050.7356.2b47 1.0                               Ok
➔ 5 0001.64fe.a930 to 0001.64fe.a95f 0.0                               PwrDeny
 6 0050.0f10.28b0 to 0050.0f10.28df 1.0                               Ok
Switch#
```

Limitation 2

Certain configurations on the Cat4507R and Cat4510R chassis exceeds the maximum amount of data power available. These configurations include the combination of the follow PIDs:

- 7 Slot configuration:
- chassis: WS-C4507R-E, WS-C4510R-E
- Dual supervisors: WS-X45-Sup6-E
- one or more: WS-X4448-GB-RJ45 or WS-X4148-FX-MT

To maximize the 10/100/1000 port density of 7 and 10 slot chassis when using redundant Supervisor engine 6-E install WS-X4548-GB-RJ45 linecards instead of WS-X4448-GB-RJ45 linecards. If WS-X4448-GB-RJ45 linecards are required two options are available.

- Option 1

  Only 4 linecard slots can be used on the Cat4507R and 6 linecard slots on the Cat4510R chassis.
- Option 2

  When all slots are required only one WS-X4448-GB-RJ45 linecard can be used.

To maximize the 100-BASE-FX port density of 7 and 10 slot chassis when using Supervisor engine 6-E install WS-4248-FE-SFP linecards with FX optics instead of WS-X4148-FX-MT linecards. If WS-X4148-FX-MT linecards are required two options are available.

- Option 1

  Only 4 linecard slots can be used on the Cat4507R and 6 linecard slots on the Cat4510R chassis.

- Option 2

  When all slots are required only one WS-X4448-GB-RJ45 linecard can be used.

Configuring Redundant Mode on a Catalyst 4500 Series Switch

By default, the power supplies in a Catalyst 4500 series switch are set to operate in redundant mode. To effectively use redundant mode, follow these guidelines:

- Use two power supplies of the same type.
- If you have the power management mode set to redundant mode and only one power supply installed, your switch will accept the configuration but operates without redundancy.

Caution If you have power supplies with different types or different wattages installed in your switch, the switch will not recognize one of the power supplies and will not have power redundancy.

- For fixed power supplies, choose a power supply that by itself is powerful enough to support the switch configuration.
- For variable power supplies, choose a power supply that provides enough power so that the chassis and PoE requirements are less than the maximum available power. Variable power supplies automatically adjust the power resources at startup to accommodate the chassis and PoE requirements. Modules are brought up first, followed by IP phones.
- The maximum available power for chassis and PoE for each power supply are listed in Table 10-6 on page 10-13.

To configure redundant mode on your Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters configuration mode.
Step 2	Switch(config)# **power redundancy-mode redundant**	Sets the power management mode to redundant mode.
Step 3	Switch(config)# **end**	Exits configuration mode.
Step 4	Switch# **show power supplies**	Verifies the power redundancy mode for the switch.

The following example shows how to set the power management mode to redundant mode.

```
Switch (config)# power redundancy-mode redundant
Switch (config)# end
Switch#
```

The following example shows how to display the current power redundancy mode. The power supplies needed by system: 1 indicates that the switch is in redundant mode.

```
Switch# show power supplies
Power supplies needed by system:1
```

```
Switch#
```

An option in the combined mode provides a form of redundancy available with only the 4200 W AC power supply. Refer to the section “Combined Mode Power Resiliency” on page 15.

Configuring Combined Mode on a Catalyst 4500 Series Switch

If your switch configuration requires more power than a single power supply can provide, set the power management mode to combined mode. Combined mode utilizes the available power for both power supplies; however, your switch will have no power redundancy.

To effectively use combined mode, follow these guidelines:

- Use power supplies of the same type and wattage (fixed or variable and AC or DC).
- If you use power supplies with different types or wattages, the switch will utilize only one of the power supplies.
- For variable power supplies, choose a power supply that provides enough power so that the chassis and PoE requirements are less than the maximum available power. Variable power supplies automatically adjust the power resources at startup to accommodate the chassis and PoE requirements.
- If you have the power management mode set to combined mode and only one power supply installed, your switch will accept the configuration, but power is available from only one power supply.
- When your switch is configured to combined mode, the total available power is not the mathematical sum of the individual power supplies. The power supplies have a predetermined current sharing ratio (See Table 10-6 on page 10-13 for more information.)
- The maximum available power for chassis and PoE for each power supply are listed in Table 10-6 on page 10-13.

To configure combined mode on your Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters configuration mode.
Step 2	Switch(config)# **power redundancy-mode combined**	Sets the power management mode to combined mode.
Step 3	Switch(config)# **end**	Exits configuration mode.
Step 4	Switch# **show power supplies**	Verifies the power redundancy mode for the switch.

The following example shows how to set the power management mode to combined mode.

```
Switch (config)# power redundancy-mode combined
Switch (config)# end
Switch#
```

The following example shows how to display the current power redundancy mode. The power supplies needed by system: 2 indicates that the switch is in combined mode.

```
Switch# show power supplies
Power supplies needed by system:2
Switch#
```

Available Power for Catalyst 4500 Series Switches Power Supplies

Table 10-6 lists the power available for use in the various Catalyst 4500 series switches power supplies. When your switch is configured to combined mode, the total available power in not the mathematical sum of the individual power supplies. The power supplies have a sharing ratio predetermined by the hardware. In combined mode, the total power available is P + (P * sharing-ratio), where P is the amount of power in the power supply.

Table 10-6 Available Power for Switch Power Supplies

Power Supply	Redundant Mode (W)	Combined Mode (W)	Sharing Ratio

1000 W AC	Chassis[1] = 1050
	PoE = 0
		Chassis = 1667
		PoE = 0
			2/3

1300 W AC	Chassis (max) = 1050 PoE (max) = 800 Chassis + PoE + Backplane ≤ 1300	Chassis (min) = 767 PoE (max) = 1333 Chassis (max) = 1667 PoE (min) = 533 Chassis + PoE + Backplane ≤ 2200	2/3
1400 W DC	Chassis (min) = 200 Chassis (max) = 1360 PoE (max)[2] = (DC Input[3] - [Chassis (min) + Backplane] / 0.75) * 0.96	Chassis = 2267[4] PoE[5]	Chassis—2/3 PoE—0

1400 W AC	Chassis = 1360
	PoE = 0[6]
		Chassis = 2473
		PoE = 0
			9/11

2800 W AC	Chassis = 1360
	PoE = 1400
		Chassis = 2473
		PoE = 2333
			Chassis[7]—9/11
			PoE[8]—2/3

[1] Chassis power includes power for the supervisor(s), all linecards, and the fan tray.

[2] The efficiency for the 1400 W DC power supply is 0.75, and 0.96 is applied to PoE.

[3] DC input can vary for the 1400 W DC power supply and is configurable. For more information, see “Special Considerations for the 1400 W DC Power Supply” on page 16.

[4] Not available for PoE.

[5] Not available for PoE.

[6] No voice power.

[7] Data-only.

[8] Inline power.

Special Considerations for the 4200 W AC Power Supply

The 4200 W AC power supply has two inputs: each can be powered at 110 or 220 V.

The output of the **show power** command for the 4200 W AC power supply is similar to that of 1400 W DC triple-input power supply (that is, the status of the sub-modules (multiple inputs) is displayed). With these two power supplies, you can distinguish sub-module “failed” versus “off,” and the status of the sub-modules (good, bad, or off):

```
Switch# show power
```

```
Power                                            Fan      Inline
Supply  Model No          Type       Status      Sensor   Status
------  ----------------  ---------  ----------- -------  -------
PS1     PWR-C45-4200ACV   AC 4200W   good        good     good
PS1-1                         220V   good
PS1-2                                   off
PS2     PWR-C45-4200ACV   AC 4200W   bad/off     good     bad/off
PS2-1                         220V   good
PS2-2                         220V   bad

Power supplies needed by system    : 1
Power supplies currently available : 2

Power Summary                      Maximum
 (in Watts)              Used     Available
----------------------   ----     ---------
System Power (12V)        140        1360
Inline Power (-50V)         0        1850
Backplane Power (3.3V)      0          40
----------------------   ----     ---------
Total                     140 (not to exceed Total Maximum Available = 2100)
Switch#
```

As with other power supplies, the two power supplies must be of the same type (4200 W AC or 1400 W DC). Otherwise, the right power supply will be put in err-disable state and the left one will be selected. In addition, all the inputs to the chassis must be at the same voltage. In redundant mode, the inputs to the left and right power supplies must be identical. If the left and right power supplies are powered in redundant mode, the power values will be based on the weaker of the two power supplies.

Note When the system is powered with a 4200 W power supply either in 110 V or 220 V combined mode operation, the available power is determined by the configuration of the system (the type of linecards, the number of linecards, number of ports consuming inline power etc.) and does not reflect the absolute maximum power.

Note In a matched redundant power supply configuration, if a power supply sub-module fails, the other (good) power supply will provide power to its full capability.

Table 10-7 illustrates how power supply is evaluated in redundant mode.

Table 10-7 Power Output in Redundant Mode

Power Supply	12 V	3.3 V	-50 V	Total
110 V	660	40	700	1050
110 V+110 V or 220 V	1360	40	1850	2100
220 V+220 V	1360	40	3700	4200

In combined mode, all the inputs to the chassis must be at the same voltage.

Table 10-8 illustrates how power supply is evaluated in combined mode.

Table 10-8 Power Output in Combined Mode

Power Supply	12 V	3.3 V	-50 V	Total
Both sides (bays) at 110 V	1200	40	1200	1873
One-side 110 V+110 V, other side 110 V	1360	40	2000	2728
Both sides at 110 V+110 V	1360	40	3100	3782
Both sides at 220 V	1360	40	3100	3782
One-side 220 V+220 V, other side 220 V	1360	40	4700	5493
Both sides at 220 V+220 V	1360	40	6800	7600

Combined Mode Power Resiliency

Note This feature only applies in combined mode when both power supply bays contain the 4200 W AC power supply.

Using the combined mode power resiliency feature, you can limit the power usage to a maximum of two or three (configurable) inputs.

With two 4200 W AC power supplies, a maximum of four inputs are available. This feature allows you to cap the power usage to that of two or three inputs. If one of the power supplies fails, no loss of power occurs because you have capped its usage to a smaller number of inputs.

To configure the combined mode resiliency feature, perform this task:

	Command	Purpose
Step 1	Switch# **`configure terminal`**	Enters configuration mode
Step 2	Switch(config)# **power redundancy combined max inputs {2 | 3}**	Limits the power usage to two or three inputs. **Note** The max inputs part of the command is ignored by all power supplies other than the 4200 W AC.
Step 3	`Switch(config)#` **`end`**	Exits configuration mode.

Let’s say that you have **max inputs 3** configured with 4 “good” (220 V) inputs and you limit the user to 5500 W instead of 7600 W with the following configuration. If one sub-unit fails or is powered off, the user would have three “good” inputs providing 5500 W and the chassis is powered at the same rate as it was prior to the failure event.

```
Switch# configuration terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# power redundancy combined max inputs 3
Switch(config)# end
Switch#
14:32:01: %SYS-5-CONFIG_I: Configured from console by console
```

Here is the output of the **show power** command prior to invoking this feature:

```
Switch# show power
sh power
Power                                             Fan      Inline
Supply  Model No          Type       Status       Sensor   Status
------  ----------------  ---------  -----------  -------  -------
PS1     PWR-C45-4200ACV   AC 4200W   good         good     good
PS1-1                         110V   good
PS1-2                         110V   good
PS2     PWR-C45-4200ACV   AC 4200W   good         good     good
PS2-1                         110V   good
PS2-2                         110V   good

Power supplies needed by system    : 1
Power supplies currently available : 2

Power Summary                      Maximum
 (in Watts)              Used     Available
----------------------   ----     ---------
System Power (12V)        140        1360
Inline Power (-50V)         0        1850
Backplane Power (3.3V)      0          40
----------------------   ----     ---------
Total                     140 (not to exceed Total Maximum Available = 2100)
```

Here is the output after invoking this features: Whereas before combined mode was indicated as **Power supplies needed = 2** in the output of the **show power** command, combined mode is now indicated by the phrase **Power supplies needed by system : 2 Maximum Inputs = 3**.

```
Switch# show power
sh power
Power                                             Fan      Inline
Supply  Model No          Type       Status       Sensor   Status
------  ----------------  ---------  -----------  -------  -------
PS1     PWR-C45-4200ACV   AC 4200W   good         good     good
PS1-1                         110V   good
PS1-2                         110V   good
PS2     PWR-C45-4200ACV   AC 4200W   good         good     good
PS2-1                         110V   good
PS2-2                         110V   good

Power supplies needed by system    : 2 Maximum Inputs = 3
Power supplies currently available : 2

Power Summary                      Maximum
 (in Watts)              Used     Available
----------------------   ----     ---------
System Power (12V)        140        2400
Inline Power (-50V)         0        2000
Backplane Power (3.3V)      0          40
----------------------   ----     ---------
Total                     140 (not to exceed Total Maximum Available = 2728)


Switch#
```

Special Considerations for the 1400 W DC Power Supply

Caution Do not mix the 1400 W DC power supply with any other power supply, even for a hot swap or other short-term emergency. Doing so can seriously damage your switch.

Keep in mind the following guidelines when using a 1400 W DC power supply with your Catalyst 4500 series switch:

- The 1400 W DC power supply works with a variety of DC sources. The DC input can vary from 300 W to 7500 W. Refer to the power supply documentation for additional information.
- The supervisor engine cannot detect the DC source plugged into the 1400 W DC power supply. If you are using the 1400 W DC power supply, use the **power dc input** command to set the DC input power. For more information on this command, see the "Configuring the DC Input for a Power Supply" section on page 10-17.
- The software automatically adjusts between system power (for modules, backplane, and fans) and PoE. Although PoE is 96 percent efficient, system power has only 75 percent efficiency. For example, each 120 W of system power requires 160 W from the DC input. This requirement is reflected in the "**Power Used"** column of the output for the **show power available** command.
- The 1400 W DC power supply has a separate power on or off switch for PoE. The power supply fan status and main power supply status are tied together. If either of them fails, both the power supply and its fan report as bad/off. You should verify that the main power is on before turning on the power for the inline switch. In addition, you should verify that the power for the inline switch is off before turning off the main power.

Configuring the DC Input for a Power Supply

To configure the DC input power for the 1400 W DC power supply or a power shelf, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters configuration mode
Step 2	Switch(config)# **power dc input** *watts*	Sets the capacity of the DC input source.
Step 3	Switch(config)# **end**	Exits configuration mode.

The same configuration is applied to both power slots. For example, if you set the **dc power input** to 1000 W, the switch expects 1000 W as the external DC source for both slot 1and slot 2 (if present) respectively.

The following example shows how to set the external DC power source to 1000 W:

```
Switch# configure terminal
Switch (config)# power dc input 1000
Switch (config)# end
Switch#
```

If you use the 1400 W DC SP power supply in combined mode, the inputs do not have to match.

Special Considerations for the 1400 W DC SP Triple Input Power Supply

Unlike the 1400 W DC power supply, the 1400 W DC SP power supply has sub-modules (multiple inputs) that can be powered on or off. With Cisco IOS Release 12.2(25)EW, the output of the **show power** command is modified to display the status of these sub-modules:

```
Switch# show power
Power                                            Fan      Inline
Supply  Model No          Type       Status       Sensor   Status
------  ----------------  ---------  -----------  -------  ------
PS1     PWR-C45-1400DC    DCSP1400W  good         good     n.a.
PS1-1                     12.5A      good
```

```
PS1-2                    15.0A      bad
PS1-3                    15.0A      off

PS2     none             --         --           --       --
```

Keep in mind the following guidelines when using a 1400 W DC SP power supply with your Catalyst 4500 series switch:

- When you use two 48 V power rails to drive two power supplies, you might employ cross-wiring to connect the power supplies (to rails) to minimize the "inrush" current drawn during an initial power up. In this situation, you should configure the switch in combined mode before you take a rail down for maintenance.
- Ordinarily, when configured for redundancy, two power supplies must be matched (have identical inputs). For example, you might provide power to inputs 1 and 3 on both PS1 and PS2. If power supplies are mismatched upon bootup, the right (second) power supply will be in err-disable state.

In a matched redundant power supply configuration, if a power supply sub-module fails, the other (good) power supply will provide power to its full capability.

Insufficient Inline Power Handling for Supervisor Engine II-TS

When the Supervisor Engine II-TS is used with the 1400 W DC power supply (PWR-C45-1400DC), and only one 12.5 A input of the power supply is used, the supervisor engine's power consumption may vary depending on the type of linecard used and on whether a linecard is inserted at slots 2 and 3. The power consumption varies between 155 W and 330 W, which also affects the maximum amount of available inline power through the supervisor engine (0 W to 175 W). Consequently, it is possible for the supervisor engine to deny inline power to a connected inline power device when one or more linecards are inserted into the chassis.

The output of the **show power detail** and **show power module** commands reveals the variable amount of power consumption attributable to the supervisor engine and summarizes the supervisor engine's inline power.

```
Switch# show power detail
show power detail
Power                                             Fan      Inline
Supply  Model No          Type       Status       Sensor   Status
------  ----------------  ---------  -----------  -------  -------
PS1     PWR-C45-1400DC    DCSP1400W  good         good     n.a.
PS1-1                     12.5A       good
PS1-2                     15.0A       off
PS1-3                     15.0A       off
PS2     none              --         --           --       --

Power supplies needed by system    : 1
Power supplies currently available : 1

Power Summary                      Maximum
 (in Watts)              Used     Available
----------------------   ----     ---------
System Power (12V)        360         360
Inline Power (-50V)         0           0
Backplane Power (3.3V)      0          40
----------------------   ----     ---------
Total                     360         400

Module Inline Power Summary (Watts)
(12V -> -48V on board conversion)
---------------------------------
                    Maximum
```

```
Mod      Used     Available
---      ----     ---------
1           5             25
---      ----     ---------

                           Watts Used of System Power (12V)
Mod   Model                currently  out of reset  in reset
----  -----------------   ---------  ------------  --------
 1    WS-X4013+TS           180         180         180
 2    WS-X4506-GB-T          60          60          20
 3    WS-X4424-GB-RJ45       90          90          50
 --   Fan Tray               30          --          --
-----------------------   ---------  ------------   -------
       Total               360         330         250

                        Watts used of Chassis Inline Power (-50V)
                        Inline Power Admin  Inline Power Oper
Mod   Model                PS     Device       PS     Device     Efficiency
----  -----------------  ----------------    ----------------    ----------
 2    WS-X4506-GB-T         0        0          0        0          89
 3    WS-X4424-GB-RJ45      -        -          -        -           -
-----------------------  ----------------    ----------------    ----------
       Total                0        0          0        0

                        Watts used of Module Inline Power (12V -> -50V)
                        Inline Power Admin  Inline Power Oper
Mod   Model                PS     Device       PS     Device     Efficiency
----  -----------------  ----------------    ----------------    ----------
 1    WS-X4013+TS           6        5          3        3          90
-----------------------  ----------------    ----------------    ----------

Switch# show power module
sh power module
                           Watts Used of System Power (12V)
Mod   Model                currently  out of reset  in reset
----  -----------------   ---------  ------------  --------
 1    WS-X4013+TS           180         180         180
 2    WS-X4506-GB-T          60          60          20
 3    WS-X4424-GB-RJ45       90          90          50
 --   Fan Tray               30          --          --
-----------------------   ---------  ------------   -------
       Total               360         330         250

                        Watts used of Chassis Inline Power (-50V)
                        Inline Power Admin  Inline Power Oper
Mod   Model                PS     Device       PS     Device     Efficiency
----  -----------------  ----------------    ----------------    ----------
 2    WS-X4506-GB-T         0        0          0        0          89
 3    WS-X4424-GB-RJ45      -        -          -        -           -
-----------------------  ----------------    ----------------    ----------
       Total                0        0          0        0

                        Watts used of Module Inline Power (12V -> -50V)
                        Inline Power Admin  Inline Power Oper
Mod   Model                PS     Device       PS     Device     Efficiency
----  -----------------  ----------------    ----------------    ----------
 1    WS-X4013+TS           6        5          3        3          90
-----------------------  ----------------    ----------------    ----------

Switch#
```

Powering Down a Module

If your system does not have enough power for all modules installed in the switch, you can power down a module, and place it in low power mode. To power down a module, perform this task:

Command	Purpose
`Switch(config)# no hw-module module num power`	Turns power down to the specified module by placing it in low power mode.

To power on a module that has been powered down, perform this task:

Command	Purpose
`Switch(config)# hw-module module num power`	Turns power on to the specified module.

This example shows how to power down module 6:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# no hw-module module 6 power
Switch(config)# end
Switch#
```

Power Management for the Catalyst 4948 Switches

You can select from AC or DC power supplies to ensure that you have enough power for your switch. The Catalyst 4948 switches support the following power supplies:

- 300 W AC
- 300 W DC

These power supplies are incompatible with Catalyst 4500 series switches. Since Power over Ethernet (PoE) is not supported on the Catalyst 4948 switch, only a limited wattage is needed. (For information on PoE, see Chapter 11, “Configuring Power over Ethernet.”) When you insert power supplies in your switch, the EEPROM on the power supplies can be read by the system software even if the supply is not powered on. You may mix AC and DC power supplies.

Power Management Modes for the Catalyst 4948 Switch

The Catalyst 4948 switches support the redundant power management mode. In this mode, if both power supplies are operating normally, each provides from 20/80 to 45/55 percent of the total system power requirements at all times. If one power supply fails, the other unit increases power to 100 percent of the total power requirement.

CHAPTER 11

Configuring Power over Ethernet

Note Before reading this chapter, read the "Preparing for Installation” section of the *Catalyst 4500 Series Installation Guide*. It is important to ensure that your installation site has enough power and cooling to accommodate the additional electrical load and heat introduced by PoE.

This chapter describes how to configure Power over Ethernet (PoE) on the Catalyst 4500 series switch.

This chapter contains the following sections:

- Overview, page 11-1
- Power Management Modes, page 11-2
- Configuring Power Consumption for Powered Devices on an Interface, page 11-4
- Displaying the Operational Status for an Interface, page 11-7
- Displaying the PoE Consumed by a Module, page 11-8

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview

The Catalyst 4500 series switch provides support for Power over Ethernet (PoE) for both Cisco Prestandard PoE and the IEEE 802.3af standard (ratified in 2003). PoE is supported by all Catalyst 4500 series chassis and requires a PoE module and power supply. The amount of PoE power available depends on the PoE capabilities of individual power supplies. Support for PoE enables the system to power inline devices, such as IP phones, IP video phones, and wireless access points over standard copper cabling (Category 5, 5e, or 6 cabling).

In addition, with PoE, you do not need to provide wall power for each PoE enabled device. This eliminates the cost for additional electrical cabling that would otherwise be necessary for connected devices. Moreover, PoE enables you to isolate critical devices on a single power system, enabling the entire system to be supported by UPS backup.

You typically deploy a Catalyst 4500 series switch in one of two deployment scenarios. The first scenario is data-only, which requires power to operate the switch and the associated modules. The second scenario supports data and PoE (also termed "inline power") for deployments where the attached device derives power from the Ethernet port.

Catalyst 4500 series switches can sense if a powered device is connected to a PoE module. They can supply PoE to the powered device if there is no power on the circuit. (If there is power on the circuit, the switch does not supply it.) The powered device can also be connected to an AC power source and supply its own power to the voice circuit.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/index.htm.

Hardware Requirements

To power a device using PoE, your chassis must use at least one of the power supplies listed in Table 11-1, and connect the device to at least one of the switching modules listed in Table 11-1.

Table 11-1 Hardware Requirements

Switching Modules	Power Supplies
WS-X4148-RJ45V	PWR-C45-1300ACV=
WS-X4224-RJ45V	PWR-C45-1400DCV=
WS-X4248-RJ21V	PWR-C45-2800ACV=
WS-X4248-RJ45V	PWR-C45-4200ACV=
WS-X4505-GB-T	
WS-X4524-GB-RJ45V	
WS-X4548-GB-RJ45V	

Power Management Modes

If your switch has a module capable of providing PoE to end stations, you can set each interface on the module to automatically detect and apply PoE if the end station requires power.

The Catalyst 4500 series switch has three PoE modes:

- **auto**—PoE interface. The supervisor engine directs the switching module to power up the interface *only* if the switching module discovers the phone and the switch has enough power. You can specify the maximum wattage that is allowed on the interface. If you do not specify a wattage, then the switch will deliver no more than the hardware-supported maximum value. This mode has no effect if the interface is not capable of providing PoE.
- **static**—High priority PoE interface. The supervisor engine preallocates power to the interface, even when nothing is connected, guaranteeing that there will be power for the interface. You can specify the maximum wattage that is allowed on the interface. If you do not specify a wattage, then the

switch preallocates the hardware-supported maximum value. If the switch does not have enough power for the allocation, the command will fail. The supervisor engine directs the switching module to power up the interface *only* if the switching module discovers the powered device.

- **never**—Data interface only The supervisor engine never powers up the interface, even if an unpowered phone is connected. This mode is only needed when you want to make sure power is never applied to a PoE-capable interface.

The switch can measure the actual PoE consumption for an 802.3af-compliant PoE module, and displays this in the **show power module** command.

PoE consumption cannot be measured on the WS-X4148-RJ45V PoE module. Therefore, for all PoE calculations, the PoE consumption on this module is presumed to be equal to its administrative PoE.

For more information, see the "Displaying the PoE Consumed by a Module" section on page 11-8.

For most users, the default configuration of "auto" works well, providing plug and play capability. No further configuration is required. However, to make an interface higher priority or data only, or to specify a maximum wattage, perform this task:

	Command	Purpose
Step 1	Switch(config)# **interface** {**fastethernet** | **gigabitethernet**} *slot/port*	Selects the interface to configure.
Step 2	Switch(config-if)# **power inline** {**auto** [**max** *milli-watts*] | **never** | **static** [**max** *milli-watts*]}	The **auto** keyword sets the interface to automatically detect and supply power to the powered device. This is the default configuration. The **static** keyword sets the interface to higher priority than auto. If necessary, you can use the **max** keyword to specify the maximum wattage allowed on the interface (4000 to 15400 milliwatts). Use the **never** keyword to disable detection and power for the PoE capable interface.
Step 3	Switch(config-if)# **end**	Exits configuration mode.
Step 4	Switch# **show power inline** {**fastethernet** | **gigabitethernet**} *slot/port*	Displays the PoE state for the switch.

Note If you set a non-PoE-capable interface to automatically detect and apply power, an error message indicates that the configuration is not valid.

The following example shows how to set the Fast Ethernet interface 4/1 to automatically detect PoE and send power through that interface, and to verify the PoE configuration:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 4/1
Switch(config-if)# power inline auto
Switch(config-if)# end
Switch# show power inline fastethernet 4/1
Available:677(w)  Used:11(w)  Remaining:666(w)

Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
```

```
--------- ------ ---------- ---------- ---------- ------------------- -----
Fa4/1     auto   on         11.2       10.0       Ieee PD             0

Interface  AdminPowerMax   AdminConsumption
             (Watts)           (Watts)
---------- --------------- --------------------
Fa4/1                15.4                 10.0
Switch#
```

The following example shows how to configure an interface so that it never supplies power through the interface:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/2
Switch(config-if)# power inline never
Switch(config-if)# end
Switch#
```

Intelligent Power Management

All Catalyst 4500 PoE-capable modules use Intelligent Power Management to provide power on each interface. When a powered device (PD) is attached to a PoE-capable port, the port will detect the PD and provision power accordingly. If a Cisco PD is used, the switch and PD negotiate power using CDP packets to determine the precise amount of power needed by the PD. If the PD is 802.3af compatible, the difference between what is mandated by the 802.3af class and what is actually needed by the PD is returned to the power budget for use by additional devices. In this way, power negotiation enables customers to stretch their power budget and use it more effectively.

Power negotiation also enables the interoperability of newer Cisco powered devices with older legacy PoE-capable ports from Cisco. Newer Cisco PDs do not consume more than what the switch port can provide.

Configuring Power Consumption for Powered Devices on an Interface

This section contains the following subsections:

- Overview, page 11-4
- PoE and Supported Cabling Topology, page 11-6

Overview

By default, when the switch detects a powered device on an interface, it assumes the powered device consumes the maximum the port can provide (7 W on a legacy Power over Ethernet (PoE) module and 15.4W on the IEEE PoE modules introduced in Cisco IOS Release 12.2(18)EW). Then, when the switch receives a CDP packet from the powered device, the wattage automatically adjusts downward to the specific amount required by that device. Normally, this automatic adjustment works well, and no further configuration is required or recommended. However, you can specify the powered device’s consumption for the entire switch (or for a particular interface) to provide extra functionality from your switch. This is useful when CDP is disabled or not available.

Note When manually configuring the consumption for powered devices, you need to account for the power loss over the cable between the switch and the powered device.

To change the power consumption for the entire switch, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] power inline consumption default milli-watts`	Sets the PoE consumption (in milliwatts) of all powered devices connected to the switch. The power consumption can range from 4000 to 15,400. To re-enable the automatic adjustment of consumption, either use the **no** keyword or specify 15,400 milliwatts.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show power inline consumption default`	Displays the administrative PoE consumption of powered devices connected to the switch. The administrative PoE is not the measured PoE value.

This example shows how to set the default PoE consumption of all powered devices connected to the switch to 5000 milliwatts, and to verify the PoE consumption:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# power inline consumption default 5000
Switch(config)# end
Switch# show power inline consumption default
Default PD consumption : 5000 mW
Switch#
```

To change the power consumption of a single powered device, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet	gigabitethernet} slot/port`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] power inline consumption milli-watts`	Sets the PoE consumption (in milliwatts) of the powered device connected to a specific interface. The power consumption can range from 4000 to 15,400. To re-enable the automatic adjustment of consumption, either use the **no** keyword or specify 15,400 milliwatts.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show power inline consumption {fastethernet	gigabitethernet} slot/port`	Displays the PoE consumption for the interface.

This example shows how to set the PoE consumption to 5000 milliwatts for interface gi 7/1 regardless what is mandated by the 802.3af class of the discovered device, or by any CDP packet received from the powered device. This example also verifies the PoE consumption on interface gi 7/1.

The following output displays the inital power consumption of the interface.

```
Switch# show power inline gi 7/1
Available:627(w)  Used:267(w)  Remaining:360(w)
```

```
Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Gi7/1     auto   on         7.9        7.0        IP Phone 7941       3

Interface  AdminPowerMax   AdminConsumption
             (Watts)           (Watts)
---------- --------------- --------------------

Gi7/1                15.4                 15.4

Switch# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# int gi 7/1
Switch(config-if)# power inline consumption 5000
Switch(config-if)# exit
Switch(config)# exit
```

The following output displays the power consumption after issuing the **power inline consumption** command against the interface:

```
Switch# sh power inline gi 7/1
Available:627(w)  Used:265(w)  Remaining:362(w)

Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Gi7/1     auto   on         5.6        5.0        Ieee PD             3

Interface  AdminPowerMax   AdminConsumption
             (Watts)           (Watts)
---------- --------------- --------------------

Gi7/1                15.4                  5.0
```

PoE and Supported Cabling Topology

When using PoE, pairs 2 and 3 (pins 1, 2, 3, and 6) of the four pairs in a standard UTP cable are used for both the Ethernet data signals and the DC power at the same time. In DC, PoE flows from pair 3 (pins 3 and 6) to the device using PoE and back to pair 2 (pins 1 and 2) while the Ethernet port transmits differential signals in pair 2 (between pins 1 and 2). This method of supplying DC power is sometimes called “phantom power” because the power signals travel over the same two pairs used to transmit Ethernet signals. The inline power signals are transparent to the Ethernet signals and do not interfere with each other. The main electrical parameter that affects inline power operation and performance is the DC resistance of the cable. The inline power method is designed to work with category 3 cable and above, up to 100 meters.

PoE has been tested and found to work with the IBM Token Ring STP cable (100 meters) when used with a Token Ring to Fast Ethernet adapter.

When you use PoE modules with type 1/2 shielded twisted pair (STP) cable configurations (90 and 125 meters), the modules perform the same as with Category 5 cable for the IEEE 802.3af standard at 10 and 100 Mbps.

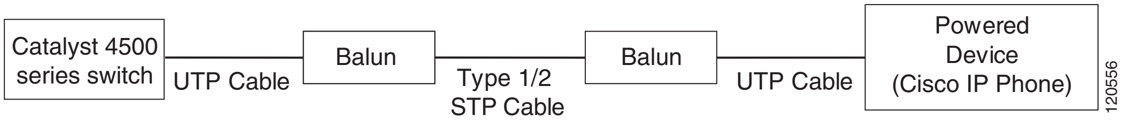
The following adapters have been tested and are the only ones supported by Cisco:

- LanTel Silver Bullet (SB-LN/VIP-DATA adapter)
- BIP-1236/S (BATM)

- RIT P/N 13712017
- RIT balun with integrated unshielded twisted pair (UTP) cable, 6 and 24 foot lengths

In Figure 11-1, a Catalyst 4500 series switch is connected to a balun through a short length of Category 5 UTP cable. Type 1 or Type 2 STP cable connects this balun to a second balun. A short length of Category 5 UTP cable connects the second balun to another Powered Device (such as a Cisco IP phone)

Figure 11-1 Supported Adapter Topology



Displaying the Operational Status for an Interface

Each interface has an operational status which reflects the PoE status for an interface. The operational status for an interface is defined as one of the following:

- on—Power is supplied by the port.
- off—Power is not supplied by the port. If a powered device is connected to an interface with external power, the switch does not recognize the powered device. The “Device” column in the **show power inline** command displays as n/a.
- Power-deny—The supervisor engine does not have enough power to allocate to the port, or the power that is configured for the port is less than the power required by the port; power is not being supplied by the port.
- err-disable—The port is unable to provide power to the connected device that is configured in static mode.
- faulty—The port failed diagnostics tests.

You can use the **show power inline** command to view the operational status for an interface.

This example shows how to display the operational status for all interfaces on module 3.

```
Switch# show power inline module 3
Available:677(w)  Used:117(w)  Remaining:560(w)

Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Fa3/1     auto   on         17.3       15.4       Ieee PD             0
Fa3/2     auto   on         4.5        4.0        Ieee PD             1
Fa3/3     auto   on         7.1        6.3        Cisco IP Phone 7960 0
Fa3/4     auto   on         7.1        6.3        Cisco IP Phone 7960 n/a
Fa3/5     auto   on         17.3       15.4       Ieee PD             0
Fa3/6     auto   on         17.3       15.4       Ieee PD             0
Fa3/7     auto   on         4.5        4.0        Ieee PD             1
Fa3/8     auto   on         7.9        7.0        Ieee PD             2
Fa3/9     auto   on         17.3       15.4       Ieee PD             3
Fa3/10    auto   on         17.3       15.4       Ieee PD             4
Fa3/11    auto   off        0          0          n/a                 n/a
```

```
Fa3/12    auto   off        0          0          n/a                n/a
Fa3/13    auto   off        0          0          n/a                n/a
Fa3/14    auto   off        0          0          n/a                n/a
Fa3/15    auto   off        0          0          n/a                n/a
Fa3/16    auto   off        0          0          n/a                n/a
Fa3/17    auto   off        0          0          n/a                n/a
Fa3/18    auto   off        0          0          n/a                n/a

--------- ------ ---------- ---------- ---------- ------------------- -----

Totals:          10   on    117.5      104.6
Switch#
```

This example shows how to display the operational status for Fast Ethernet interface 4/1:

```
Switch# show power inline fa4/1
Available:677(w)  Used:11(w)  Remaining:666(w)

Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Fa4/1     auto   on         11.2       10.0       Ieee PD             0

Interface  AdminPowerMax   AdminConsumption
             (Watts)           (Watts)
---------- --------------- --------------------

Fa4/1                15.4                 10.0
Switch#
```

Displaying the PoE Consumed by a Module

The switch can measure the actual PoE consumption for an 802.3af-compliant PoE module, and it displays the measured PoE in both the **show power module** and **show power detail** commands.

For all PoE calculations, the PoE consumption on the WS-X4148-RJ45V module is presumed to be equal to its administrative PoE.

The 802.3af-compliant PoE modules can consume up to 20 W of PoE to power FPGAs and other hardware components on the module. Be sure to add at least 20 W to your PoE requirements for each 802.3af-compliant PoE module to ensure that the system has adequate power for the PDs connected to the switch.

The example below displays the PoE consumption for an 802.3af-compliant module using the **show power module** command.

The “Inline Power Oper” column displays the amount of PoE consumed by the powered devices that are attached to the module, in addition to the PoE consumed by the FPGAs and other hardware components on the module. The “Inline Power Admin” column displays only the amount of PoE allocated by the powered devices attached to the module.

Note The operating PoE consumption for an 802.3af-compliant module can be non-zero, even when there are no powered devices attached to the module, because of the PoE consumed by FPGAs and other hardware components on the module. In addition, the operating PoE can vary due to fluctuations in the PoE consumed by the hardware components.

```
Switch# show power module

Watts Used of System Power (12V)
Mod   Model               currently  out of reset  in reset
----  -----------------   ---------  ------------  --------
 1    WS-X4013+TS             330         330         330
 2    WS-X4548-GB-RJ45V       60          60          20
 3    WS-X4548-GB-RJ45V       60          60          20
 --   Fan Tray                30          --          --
-----------------------   ---------  ------------   -------
       Total                 480         450         370

                        Watts used of Chassis Inline Power (-50V)
                        Inline Power Admin  Inline Power Oper
Mod   Model                PS     Device       PS     Device     Efficiency
----  -----------------  ----------------    ----------------    ----------
 2    WS-X4548-GB-RJ45V    138      123         73       65          89
 3    WS-X4548-GB-RJ45V      0        0         22       20          89
-----------------------  ----------------    ----------------    ----------
       Total               138      123         95       85

                        Watts used of Module Inline Power (12V -> -50V)
                        Inline Power Admin  Inline Power Oper
Mod   Model                PS     Device       PS     Device     Efficiency
----  -----------------  ----------------    ----------------    ----------
 1    WS-X4013+TS         128      128         63       63         100
-----------------------  ----------------    ----------------    ----------

Switch#
```

The example below displays the PoE consumption for an 802.3af-compliant module using the **show power detail** and **show power inline** commands.

The “Inline Power Oper” column displays the amount of PoE consumed by the powered devices that are attached to the module, in addition to the PoE consumed by the FPGAs and other hardware components on the module. The “Inline Power Admin” column displays only the amount of PoE allocated by the powered devices attached to the module.

```
Switch# show power detail

Power                                             Fan      Inline
Supply  Model No          Type       Status       Sensor   Status
------  ----------------  ---------  -----------  -------  -------
PS1     PWR-C45-1300ACV   AC 1300W   good         good     good
PS2     none              --         --           --       --

Power supplies needed by system    : 1
Power supplies currently available : 1

Power Summary                      Maximum
 (in Watts)              Used     Available
----------------------   ----     ---------
System Power (12V)        480        1000
Inline Power (-50V)       138         800
Backplane Power (3.3V)      0           0
----------------------   ----     ---------
Total                     618 (not to exceed Total Maximum Available = 1300)
```

```
Module Inline Power Summary (Watts)
(12V -> -48V on board conversion)
---------------------------------
                    Maximum
Mod      Used      Available
---      ----      ---------
1         128           158
---      ----      ---------

                           Watts Used of System Power (12V)
Mod   Model             currently  out of reset  in reset
----  -----------------   ---------  ------------  --------
 1    WS-X4013+TS          330         330         330
 2    WS-X4548-GB-RJ45V     60          60          20
 3    WS-X4548-GB-RJ45V     60          60          20
 --   Fan Tray              30          --          --
-----------------------   ---------  ------------   -------
       Total               480         450         370

                        Watts used of Chassis Inline Power (-50V)
                        Inline Power Admin  Inline Power Oper
Mod   Model               PS     Device       PS     Device     Efficiency
----  -----------------  ----------------    ----------------    ----------
 2    WS-X4548-GB-RJ45V    138      123         73       65          89
 3    WS-X4548-GB-RJ45V      0        0         22       20          89
-----------------------  ----------------    ----------------    ----------
       Total              138      123         95       85

                        Watts used of Module Inline Power (12V -> -50V)
                        Inline Power Admin  Inline Power Oper
Mod   Model               PS     Device       PS     Device     Efficiency
----  -----------------  ----------------    ----------------    ----------
 1    WS-X4013+TS         128      128         64       64         100
-----------------------  ----------------    ----------------    ----------

Switch# show power inline g1/1
Module 1 Inline Power Supply: Available:158(w)  Used:128(w)  Remaining:30(w)

Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Gi1/1     auto   on         10.3       10.3       CNU Platform        3

Interface  AdminPowerMax   AdminConsumption
             (Watts)           (Watts)
---------- --------------- --------------------

Gi1/1                15.4                 15.4

switch# show power inline g2/1
Chassis Inline Power Supply: Available:800(w)  Used:138(w)  Remaining:662(w)

Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Gi2/1     auto   on         11.5       10.2       CNU Platform        n/a

Interface  AdminPowerMax   AdminConsumption
             (Watts)           (Watts)
---------- --------------- --------------------

Gi2/1                15.4                 15.4
```

```
Switch# show power inline module 1
Module 1 Inline Power Supply: Available:158(w)  Used:128(w)  Remaining:30(w)

Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Gi1/1     auto   on         10.3       10.3       CNU Platform        3
Gi1/2     auto   on         10.3       10.3       CNU Platform        3
Gi1/3     auto   on         10.3       10.3       CNU Platform        3
Gi1/4     auto   on         10.3       10.3       CNU Platform        3
Gi1/5     auto   on         10.3       10.3       CNU Platform        3
Gi1/6     auto   on         10.3       10.3       CNU Platform        3
Gi1/7     auto   on         10.3       10.3       CNU Platform        3
Gi1/8     auto   on         10.3       10.3       CNU Platform        3
Gi1/9     auto   on         10.3       10.3       CNU Platform        3
Gi1/10    auto   on         15.4       15.4       Cisco/Ieee PD       3
Gi1/11    auto   on         10.3       10.3       CNU Platform        3
Gi1/12    auto   on         10.3       10.3       CNU Platform        3
--------- ------ ---------- ---------- ---------- ------------------- -----

Totals:          12   on    128.2      128.2
switch#

switch# show power inline module 2
Chassis Inline Power Supply: Available:800(w)  Used:138(w)  Remaining:662(w)
Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Gi2/1     auto   on         11.5       10.2       CNU Platform        n/a
Gi2/2     auto   on         11.5       10.2       CNU Platform        n/a
Gi2/3     auto   on         11.5       10.2       CNU Platform        n/a
Gi2/4     auto   on         11.5       10.2       CNU Platform        n/a
Gi2/5     auto   off        0.0        0.0        n/a                 n/a
Gi2/6     auto   off        0.0        0.0        n/a                 n/a
Gi2/7     auto   off        0.0        0.0        n/a                 n/a
Gi2/8     auto   off        0.0        0.0        n/a                 n/a
Gi2/9     auto   on         11.5       10.2       CNU Platform        3
Gi2/10    auto   on         11.5       10.2       CNU Platform        n/a
Gi2/11    auto   on         11.5       10.2       CNU Platform        n/a
Gi2/12    auto   on         11.5       10.2       CNU Platform        n/a
Gi2/13    auto   on         11.5       10.2       CNU Platform        3
Gi2/14    auto   on         11.5       10.2       CNU Platform        3
Gi2/15    auto   on         11.5       10.2       CNU Platform        3
Gi2/16    auto   on         11.5       10.2       CNU Platform        3
Gi2/17    auto   off        0.0        0.0        n/a                 n/a
Gi2/18    auto   off        0.0        0.0        n/a                 n/a
Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Gi2/19    auto   off        0.0        0.0        n/a                 n/a
Gi2/20    auto   off        0.0        0.0        n/a                 n/a
Gi2/21    auto   off        0.0        0.0        n/a                 n/a
Gi2/22    auto   off        0.0        0.0        n/a                 n/a
Gi2/23    auto   off        0.0        0.0        n/a                 n/a
Gi2/24    auto   off        0.0        0.0        n/a                 n/a
Gi2/25    auto   off        0.0        0.0        n/a                 n/a
```

```
Gi2/26    auto   off        0.0        0.0        n/a                 n/a
Gi2/27    auto   off        0.0        0.0        n/a                 n/a
Gi2/28    auto   off        0.0        0.0        n/a                 n/a
Gi2/29    auto   off        0.0        0.0        n/a                 n/a
Gi2/30    auto   off        0.0        0.0        n/a                 n/a
Gi2/31    auto   off        0.0        0.0        n/a                 n/a
Gi2/32    auto   off        0.0        0.0        n/a                 n/a
Gi2/33    auto   off        0.0        0.0        n/a                 n/a
Gi2/34    auto   off        0.0        0.0        n/a                 n/a
Gi2/35    auto   off        0.0        0.0        n/a                 n/a
Gi2/36    auto   off        0.0        0.0        n/a                 n/a
Gi2/37    auto   off        0.0        0.0        n/a                 n/a
Gi2/38    auto   off        0.0        0.0        n/a                 n/a
Gi2/39    auto   off        0.0        0.0        n/a                 n/a
Gi2/40    auto   off        0.0        0.0        n/a                 n/a
Interface Admin  Oper            Power(Watts)     Device              Class
                            From PS    To Device
--------- ------ ---------- ---------- ---------- ------------------- -----

Gi2/41    auto   off        0.0        0.0        n/a                 n/a
Gi2/42    auto   off        0.0        0.0        n/a                 n/a
Gi2/43    auto   off        0.0        0.0        n/a                 n/a
Gi2/44    auto   off        0.0        0.0        n/a                 n/a
Gi2/45    auto   off        0.0        0.0        n/a                 n/a
Gi2/46    auto   off        0.0        0.0        n/a                 n/a
Gi2/47    auto   off        0.0        0.0        n/a                 n/a
Gi2/48    auto   off        0.0        0.0        n/a                 n/a
--------- ------ ---------- ---------- ---------- ------------------- -----

Totals:          12   on    138.2      123.0
Switch#
```

CHAPTER 12

Configuring the Catalyst 4500 Series Switch with Cisco Network Assistant

This chapter describes how to install Network Assistant on the workstation and configure the Catalyst 4500 (or 4900) series switch to communicate with Network Assistant. (Heretofore, the term *Catalyst 4500 series switch* will be used to refer to both switch types.) It also describes how to create communities and clusters. These are two technologies used by Network Assistant to manage a group of network devices, including the Catalyst 4500 series switch.

Network Assistant is a free network management tool that enables you to configure and manage Catalyst 4500 series switches using a Graphical User Interface (GUI). Network Assistant works in both secure and unsecure environments. Network Assistant manages standalone devices or groups of devices or switches (in communities or clusters) from anywhere in your intranet. Using Network Assistant, you can perform multiple configuration tasks without having to remember commands.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/go/NetworkAssistant.

Configuring and Using the Network Assistant

This chapter contains these topics:

- Network Assistant-Related Features and Their Defaults, page 12-2
- Overview of the CLI Commands, page 12-2
- Configuring Your Switch for Network Assistant, page 12-3
- Managing a Network using Community, page 12-5
- Converting a Cluster into a Community, page 12-9
- Managing a Network using Cluster, page 12-10
- Configuring Network Assistant in Community or Cluster Mode, page 12-13

Note The Network Assistant is not bundled with an online software image on Cisco.com. You can download the Network Assistant at: http://www.cisco.com/go/NetworkAssistant

Note For information on software and hardware requirements, installing Network Assistant, launching Network Assistant, and connecting Network Assistant to a device,, refer to *Getting Started with Cisco Network Assistant*, available at the URL: http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cna/v2_0/gsg/index.htm

Network Assistant-Related Features and Their Defaults

Table 1 lists the Network Assistant-related configuration parameters on a Catalyst 4500 series switch.

Table 1 Network Assistant-Related Configuration on a Catalyst 4500 Series Switch

Feature	Default Value	Recommended Value
Authentication	Disabled	Optional
IP address	Depends on community or discovery option[1]	User selectable **Note** Community is *not* supported on Supervisor Engine 6-E.
IP HTTP port number	80	Optional[2]
IP HTTPS port number	443	Optional[3]
IP HTTP server	Disabled	Enabled[4]
Cluster run	Disabled	Enabled[5]

[1] You need to set an IP address in each switch for community device discovery and for the cluster commander.

[2] Port number on the Network Assistant and the Catalyst 4500 series switch must match.

[3] You can only change this value for a cluster of devices. Port number on the Network Assistant and on the Catalyst 4500 series switch must match. Value can be changed to any non-default number above 1024.

[4] Required for Network Assistant to access the device.

[5] Enabled only if you want to manage a cluster of devices.

Overview of the CLI Commands

Table 2 is an overview of the Network Assistant-related CLI commands.

Table 2 CLI Commands

Command	Functions
[no] cluster enable	Names the cluster.
[no] cluster run	Enables clustering. **Note** This command is used strictly for clustering.

Table 2 CLI Commands

Command	Functions
[no] ip http server	Configures the HTTP on a switch.
[no] ip http port *port_number*	Configures the HTTP port.
[no] ip domain-name *domain_name*	Configures the domain on the switch.
[no] ip http secure-server	Configures and enable HTTPS on a switch.
[no] ip http secure-port *port_number*	Configures the HTTPS port.
[no] ip http max-connections *connection_number*	Configures the maximum concurrent connections to the HTTP server.
[no] ip http timeout-policy idle *idle_time* **life** *life_time* **requests** *requests*	Configures the HTTPS port. A **idle** value of 180 seconds is recommended. A **life** value of 180 seconds is recommended. The recommended maximum number of **requests** allowed is 25.
line vty	Configures additional VTYs for use by CNA.
show version	Displays the Cisco IOS release.
show running-config	Displays the switch configuration.
vtp domain	Creates a VTP domain to manage VLANs.
vtp mode	Sets the behavior for VTP management of the VLANs.

Configuring Your Switch for Network Assistant

The following topics are discussed:

- (Minimum) Configuration Required to Access Catalyst 4500 Accessible from CNA, page 12-3
- (Additional) Configuration Required to use Community, page 12-4
- (Additional) Configuration Required to use Cluster, page 12-5

(Minimum) Configuration Required to Access Catalyst 4500 Accessible from CNA

If you use the default configuration, access the Catalyst 4500 series switch and enter the **ip http server** (for HTTP) or **ip http secure-server** (for HTTPS) global configuration command:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.

Step 2	`Switch(config)# ip http server`
	or
	`Switch(config)# ip domain-name domain_name`
		(**HTTP only**) Enables the HTTP server on the switch. By default, the HTTP server is disabled.
		Enables the domain name on the switch to configure HTTPS.

Step 3	`Switch(config)# ip http secure-server`	Enables the HTTPS server on the switch. By default, the HTTPS server is disabled.

	Command	Purpose
Step 4	`Switch(config)# ip http max-connections connection_number`	Configures the maximum concurrent connections to the HTTP server. A *connection_number* of 16 is recommended.
Step 5	`Switch(config)# ip http timeout-policy idle idle_time life life_time requests requests`	Configures the HTTPS port. The **idle** keyword specifies the maximum amount of time a connection can stay idle. A **idle** value of 180 seconds is recommended. The **life** keyword specifies the maximum amount of time a connection can stay open since it was established. A **life** value of 180 seconds is recommended. The **requests** keyword specifies the maximum amount of requests on a connection. The recommended maximum number of **requests** allowed is 25.
Step 6	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show running-config`	Verifies the configuration.

Note If you have enabled clustering, disable clustering before configuring a community (see Table 2).

(Additional) Configuration Required to use Community

Note Community is *not* supported on Supervisor Engine 6-E.

If you plan to use community, define an IP address on each switch:

	Command	Purpose
Step 1	`Switch# configuration terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface {vlan vlan_ID	{fastethernet	gigabitethernet} slot/interface	Port-channel number}`	Selects an interface.
Step 3	`Switch(config-if)# ip address ip_address address_mask`	(Optional) Assigns an IP address to the Catalyst 4500 series **Note** This step is mandatory if the switch is part of community or is a cluster command switch. This step is optional if the switch is a cluster member candidate.
Step 4	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show running-config`	Verifies the configuration.

(Additional) Configuration Required to use Cluster

If you plan to use clustering, enter the **cluster run** global configuration command on each device and enter the **ip address** interface configuration command on the cluster commander:

	Command	Purpose
Step 1	`Switch# configuration terminal`	Enters global configuration mode.
Step 2	`Switch(config)# cluster run`	Enables clustering. **Note** Enable clustering on all switches that are part of the potential cluster.
Step 3	`Switch(config)# cluster enable`	Names the cluster.
Step 4	`Switch(config)# interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface | Port-channel number}`	Selects an interface.
Step 5	`Switch(config-if)# ip address ip_address address_mask`	(Optional) Assigns an IP address to the Catalyst 4500 series switch cluster master. **Note** This step is mandatory if the switch is part of a community or is a cluster command switch. This step is optional if the switch is a cluster member candidate.
Step 6	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show running-config`	Verifies the configuration.

Managing a Network using Community

Note Community is *not* supported on Supervisor Engine 6-E.

This section describes how to use communities to manage *devices* (including Catalyst 4500 series switches, routers, access points, and PIX firewalls) using the Network Assistant application.

When you use communities to group the switches in your network, the only requirements are an HTTP server and that you configure an IP address on each switch.

The total number of devices in the community cannot exceed 20 total devices (including up to 4 Catalyst 4500 series switches (modular), 16 Catalyst 2900/3500 or Catalyst 4948/4948-10GE switches ((non-modular), 2 routers, and 2 PIX firewalls).

Note Access points have been eliminated from the device limits. There is no current limit for the number of access points that can be managed by CNA.

Note The **Add to Community** dialog display any number of devices, but only allows you to select 20 devices. If you try to add a 21st device, the dialog displays the 21st device and prompts you to select the unwanted device.

Note For complete procedures for using Network Assistant to configure switch communities, refer to *Getting Started with Cisco Network Assistant*, available at:

http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cna/v2_0/gsg/index.htm

For the CLI cluster commands, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/index.htm

This section describes the guidelines, requirements, and caveats that you should understand before you create a community. This section contains the following topics:

- Candidate and Member Characteristics, page 12-6
- Automatic Discovery of Candidates and Members, page 12-7
- Community Names, page 12-7
- Hostnames, page 12-7
- Passwords, page 12-8
- Access Modes in Network Assistant, page 12-8
- Community Information, page 12-8

Candidate and Member Characteristics

Candidates are network devices that have IP addresses but are not part of a community. Members are network devices that are currently part of a community.

To join a community, a candidate must meet these requirements:

- It has an IP address.
- Cisco Discovery Protocol (CDP) version 2 is enabled (the default) - if you want the device to be autodiscovered.
- It has HTTP (or HTTPS) enabled.

Note A cluster member can be added to a community, but the reverse is not possible.

Note If the cluster commander is added to a community, the other member devices of the cluster are not added automatically. The cluster members must be added to the community on an individual basis in order to be managed.

Automatic Discovery of Candidates and Members

Network Assistant forms a community using CDP to locate or discover all the available devices in the network. Beginning with the IP address for a starting device and the port numbers for HTTP (or HTTPS) protocols, Network Assistant uses CDP to compile a list of community candidates that neighbor the starting device. Network Assistant can discover candidate and member devices across multiple networks and VLANs as long as they have valid IP addresses.

Note By default, Network Assistant in community mode discovers up to four hops away.

See the "Candidate and Member Characteristics" section on page 12-6 for a list of requirements that network devices must meet in order to be discovered.

Note Do not disable CDP on candidates, members, or on any network devices that you might want Network Assistant to discover.

Note PIX firewalls do not support the CDP, so they are not automatically shown as neighbors in the Topology view. They are shown only after you add them to a community with the Create Community or Modify Community window. To see a PIX firewall link to another community member, you must add the link manually by selecting ADD Link in a Topology popup menu.

You can edit the list of discovered devices to fit your needs and add them to the community. As each device is added to the community, its neighbors are discovered and added to the list of candidate devices. If Network Assistant fails to discover a device you can add it manually through the IP management IP address.

Community Names

When you apply the community configuration information to the list of member devices, Network Assistant requests that you enter a name (or IP address) for the community. You need to assign a name to the community before you can manage it. Network Assistant saves the name to your PC.

The community name can consist of the characters 0-9, a-z and A-Z, with spaces allowed between the characters.

Note You can connect to a cluster only through an IP address. When you select a name it is always for the community.

Hostnames

You do not need to assign a hostname to a starting device or a community member. However, Cisco recommends it and Network Assistant does not assign one by default. If a discovered device does have a hostname, Network Assistant saves it to your PC as identifying information for that device along with its IP address, communication protocol, and designated protocol port.

Passwords

Although you do not need to assign a password to a device if it will become a community member, Cisco recommends that you do so.

Community members can have different passwords.

Communication Protocols

Network Assistant uses the HTTP (or HTTPS) protocols to communicate with network devices. It attempts communication with HTTP (or HTTPS) when using CDP to discover candidate devices.

Access Modes in Network Assistant

When Network Assistant is connected to a community or cluster, two access modes are available: read-write and read-only, depending on the password.

Community Information

Network Assistant saves all community configuration information and individual device information such as IP address, hostname, and communication protocol to your local PC. When Network Assistant connects to a community, it uses the locally saved data to rediscover the member devices.

If you attempt to use a different PC to manage an existing community, the member device information will not be available. You will need to create the community again and add the same member devices.

Adding Devices

There are three ways to add members to a community.

The first uses the Devices Found window on Network Assistant to add devices that you discovered to a new community:

a. In the Devices Found window, select candidate devices that you wish to add.

   To add more than one candidate, press **Ctrl** and make your choices, or press **Shift** and choose the first and last device in a range.

b. Click **Add**.

The second way uses the Modify Community window to add devices to an existing community:

a. Choose **Application > Communities** to open the Communities window.

b. In the Communities window, select the name of the community to which you would like to add a device, and click **Modify**.

c. To add a single device manually, enter the IP address for the desired device in the Modify Community window, and click **Add**.

d. To discover candidate devices, enter the IP address for the starting device, and click **Discover**.

e. Select a candidate device from the list, click **Add**, and click **OK**.

   To add more than one candidate, press **Ctrl** and make your choices, or press **Shift** and choose the first and last device in a range.

The third way to add a device uses the Topology view:

a. If the Topology view is not displayed, choose **View** window**> Topology** from the feature bar.

b. Right-click a candidate icon, and select **Add to Community**.

Candidates are cyan; members are green. To add more than one candidate, press **Ctrl** and left-click the candidates that you want to add.

When a community has 20 members, the **Add to Community** option is not available for that community. In this case, you must remove a member before adding a new one.

Note If you are logged into a community and you delete that community from some other CNA instance, then unless you close that community session, you can perform all the configurations through that session. After you close that session (and thereby delete the community), you will not be able to connect to that community.

Converting a Cluster into a Community

Note Community is *not* supported on Supervisor Engine 6-E.

The Cluster Conversion wizard helps you convert a cluster into a community. When you complete the conversion, you can immediately manage the device group as a community. The benefits of managing a community is that the communication with the devices in a community is more secure (through multiple passwords and HTTPS) than in a cluster. Moreover, device availability is greater, and the range of devices that can be members is broader.

Note The Cluster Conversion wizard does not alter your cluster definition. This means that you can still manage the devices as a cluster.

To launch the Cluster Conversion Wizard, follow these steps:

Step 1 Start Network Assistant and connect to an existing cluster through its commander IP address.

Step 2 In the feature bar, click **Configure** > **Cluster** > **Cluster Conversion Wizard**.

You will see the query "Do you want to convert this cluster to a community?”

Step 3 Select **Yes** to proceed or **No** if you want to manually bring up the Cluster Conversion Wizard.

If you select **Yes**, the Welcome screen appears, providing information about clusters, communities, and their benefits.

Next, a table appears listing the devices in the cluster starting with those that have no IP address and subnet mask. Be aware that all the devices in the cluster must have an IP address and subnet mask to be members of a community.

Note If a device has more than one interface with an IP address and subnet mask, you see more than one interface listed when you click in the cell. You can choose a different interface from the one originally shown.

Step 4 In the IP Address column, enter an IP address for each device that does not have one.

Step 5 In the Subnet Mask column, click in the cell for each device that does not have a subnet mask and select one.

Step 6 Enter a name for the community.

Step 7 Click **Finish** to begin the conversion.

When the conversion completes, Network Assistant restarts and automatically connects to the newly created community.

Note If you have enabled clustering, you should disable clustering before configuring a community (see Table 2).

Managing a Network using Cluster

This section describes how to use clustering to create and manage Catalyst 4500 series switches using the standalone Network Assistant application or the command-line interface (CLI).

You can use clustering to group the switches in your network. You must enter the cluster run command on each switch to be managed. The major advantage is that you can manage 16 devices with one IP address.

Note Clustering is the auto- discovering mechanism used in CNA 1.0.

Note For complete procedures for using Network Assistant to configure switch clusters, refer to *Getting Started with Cisco Network Assistant*, available at:

http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cna/v2_0/gsg/index.htm

For the CLI cluster commands, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/index.htm

This section contains the following topics:

- Understanding Switch Clusters, page 12-10
- Using the CLI to Manage Switch Clusters, page 12-12

Understanding Switch Clusters

These sections describe:

- Clustering Overview, page 12-11
- Cluster Command Switch Characteristics, page 12-11
- Candidate Switch and Cluster Member Switch Characteristics, page 12-12

Clustering Overview

A *switch cluster* is a set of up to 16 connected, cluster-capable Catalyst switches that are managed as a single entity. The switches in the cluster use the switch clustering technology so that you can configure and troubleshoot a group of different Catalyst 4500 series switch platforms through a single IP address.

Using switch clusters simplifies the management of multiple switches, regardless of their physical location and platform families.

Note By default, Network Assistant in clustering mode discovers up to seven hops away.

In a switch cluster, one switch must be the *cluster commander switch*, and up to 15 other switches can be *cluster member switches*. The total number of switches in a cluster cannot exceed 16 switches. The cluster command switch is the single point of access used to configure, manage, and monitor the cluster member switches. Cluster members can belong to only one cluster at a time.

Note Always choose a Catalyst 4500 or 4948 series switch as the cluster command switch.

Cluster Command Switch Characteristics

A cluster command switch must meet these requirements:

- It is using Cisco IOS Release 12.2(20)EWA or later.
- It has an IP address.
- It has Cisco Discovery Protocol (CDP) version 2 enabled (the default).
- It is using cluster-capable software and has clustering enabled.
- It has IP HTTP (or HTTPS) server enabled.

  Note On a Catalyst 4500 series switch, neither HTTP or HTTPS is enabled by default.

- It has 16 VTY lines.

  Note On a Catalyst 4500 series switch, the default is 4 lines. You configure the switch to set the value to 16.

- It is not a command or cluster member switch of another cluster.

Note If your switch cluster contains a Catalyst 4500 series switch, the cluster command switch must also be a Catalyst 4500 series switch.

Network Assistant and VTY

Network Assistant uses virtual terminal (VTY) lines to communicate with the cluster command device. Catalyst 4500 series switches have 5 VTY lines configured by default. Network Assistant can employ an additional 8 lines. Therefore, you should configure the maximum number of lines (or at least, 8 + 5 = 13) so that Network Assistant can communicate with the switch and not use VTY lines that might be needed for telnet.

You can configure the Catalyst 4500 series switch to support an appropriate number of VTY lines with the **line vty** configuration command. For example, the **line vty 6 15** command configures the switch to include 9 VTY lines.

Note If your existing VTY lines have non-default configurations, you might want to apply those configurations to the new VTY lines.

Candidate Switch and Cluster Member Switch Characteristics

Candidate switches are cluster-capable switches that are not part of a cluster. Cluster member switches are switches that are currently part of a switch cluster. Although not required, a candidate or cluster member switch can have its own IP address and password.

Note The hostname of a candidate should not be in the form [a-zA-Z0-9]-*n*, where *n* is 0-16. These names are reserved.

To join a cluster, a candidate switch must meet these requirements:

- It is running cluster-capable software and has clustering enabled.
- It has CDP version 2 enabled.
- It has HTTP server enabled.

  Note Even when HTTP is enabled on the commander switch, communication between the commander switch and member switch is still carried over HTTP. So, it is not secure.

- It has 16 VTY lines.
- It is not a command or cluster member switch of another cluster.
- It is connected to the cluster command switch through at least one common VLAN.

  It is recommended that you configure the Catalyst 4500 candidate and cluster member switches with an SVI on the VLAN connection to the cluster command switch.

Using the CLI to Manage Switch Clusters

You can configure cluster member switches from the CLI by first logging in to the cluster command switch. Enter the **rcommand** user EXEC command and the cluster member switch number to start a Telnet session (through a console or Telnet connection) and to access the cluster member switch CLI. The command mode changes and the Cisco IOS commands operate as usual. Enter the **exit** privileged EXEC command on the cluster member switch to return to the command-switch CLI.

This example shows how to log in to member-switch 3 from the command-switch CLI:

```
switch# rcommand 3
```

If you do not know the member-switch number, enter the **show cluster members** privileged EXEC command on the cluster command switch. For more information about the **rcommand** command and all other cluster commands, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

The Telnet session accesses the member-switch CLI at the same privilege level as on the cluster command switch. The Cisco IOS commands then operate as usual. For instructions on configuring the switch for a Telnet session, see the “Accessing the CLI Through Telnet” section on page 2-2.

Note CISCO-CLUSTER_MIB is not supported.

Configuring Network Assistant in Community or Cluster Mode

Note Community is *not* supported on Supervisor Engine 6-E.

This section provides a detailed explanation of the CLI used to configure Network Assistant to work in a community or cluster. Network Assistant communicates with a Catalyst 4500 series switch by sending Cisco IOS commands over an HTTP (or HTTPS) connection.

The following topics are discussed:

- Configuring Network Assistant in on a Networked Switch in Community Mode, page 12-13
- Configuring Network Assistant in a Networked Switch in Cluster Mode, page 12-17

Configuring Network Assistant in on a Networked Switch in Community Mode

To configure Network Assistant on a networked switch in community mode, follow these steps:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`enable password`** *`name`*	Enables password protection of configuration mode.
Step 3	`Switch(config)#` **`vtp domain`** *`name`*	Creates a VTP domain to manage VLAN.
Step 4	`Switch(config)#` **`vlan`** *`vlan_id`*	Creates a VLAN.
Step 5	`Switch(config-vlan)#` **`interface`** `{`**`vlan`** *`vlan_ID`* `|` `{`**`fastethernet`** `|` **`gigabitethernet`**`}` *`slot/interface`* `|` **`Port-channel`** *`number`*`}`	Selects the interface that will connect to your CNA-enabled PC.
Step 6	`Switch(config-if)#` **`switchport access vlan`** *`vlan_id`*	Enables the selected interface to be in the specified VLAN.
Step 7	`Switch(config-if)#` **`interface`** `{`**`vlan`** *`vlan_ID`* `|` *`slot/interface`* `|` **`Port-channel`** *`number`*`}`	Select the VLAN instance for configuration.
Step 8	`Switch(config-if)#` **`ip address`** *`ip_address`*	Assigns an IP address to the SVI.
Step 9	`Switch(config-if)#` **`no shutdown`**	Enables the interface.
Step 10	`Switch(config-if)#` **`ip http server`**	Starts the HTTP server so that Network Assistant can talk to the switch.
Step 11	`Switch(config)#` **`ip domain-name`** *`domain_name`*	Enables the domain name on the switch to configure HTTPS.
Step 12	`Switch(config)#` **`ip http secure-server`**	Enables the HTTPS server on the switch. By default, the HTTPS server is disabled.
Step 13	`Switch(config)#` **`ip http max-connections`** *`connection_number`*	Configures the maximum concurrent connections to the HTTP server. A *connection_number* of 16 is recommended.

	Command	Purpose
Step 14	`Switch(config)# ip http timeout-policy idle idle_time life life_time requests requests`	Configures the HTTPS port. The **idle** keyword specifies the maximum amount of time a connection can stay idle. A **idle** value of 180 seconds is recommended. The **life** keyword specifies the maximum amount of time a connection can stay open since it was established. A **life** value of 180 seconds is recommended. The **requests** keyword specifies the maximum number of requests on a connection. A **requests** value of 25 recommended.
Step 15	`Switch(config-if)# ip http secure-server`	(Optionally) Enables the switch to accept HTTPS connections from Network Assistant.
Step 16	`Switch(config)# ip route a.b.c`	Establishes the route to the default router, usually supplied by the local Internet Provider. **Note** This line represents the only difference between the configuration for a standalone and a networked switch.
Step 17	`Switch(config)# line con 0`	Select the console port to perform the configuration.
Step 18	`Switch(config-line)# exec-timeout x y`	Configures an automatic session logout if no keyboard input or output is displayed on the terminal.
Step 19	`Switch(config-line)# password password`	Specifies a password for the console port.
Step 20	`Switch(config-line)# login`	Allows login to the console port.
Step 21	`Switch(config-line)# line vty x y`	Creates additional VTY lines for CNA to access the switch.
Step 22	`Switch(config-line)# password password`	Specifies a password for the switch.
Step 23	`Switch(config-line)# login`	Allows login to the switch.
Step 24	`Switch(config-line)# line vty x y`	Creates additional VTY lines for CNA to access the switch.
Step 25	`Switch(config-line)# password password`	Specifies a password for the switch.
Step 26	`Switch(config-line)# login`	Allows login to the switch.
Step 27	`Switch(config-line)# end`	Returns to privileged EXEC mode.
Step 28	`Switch# show running-config`	Verifies the configuration.

This example shows how to configure Network Assistant on a networked switch in community mode:

```
Switch# configure terminal
Switch(config)# vtp domain cnadoc
Changing VTP domain name from cisco to cnadoc
Switch(config)# vlan 2
Switch(config-vlan)# exit
Switch(config)# interface GigabitEthernet 2/1
Switch(config-if)# switchport access vlan 2
Switch(config-if)# exit
Switch(config)# interface vlan 2
Switch(config-if)# ip address 123.123.123.1 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# ip http server
Switch(config)# ip domain-name cisco.com
Switch(config)# ip http secure-server
Switch(config)# ip http max-connections 16
Switch(config)# ip http timeout-policy idle 180 life 180 requests 25
Switch(config)# ip route 0.0.0.0 0.0.0.0 123.123.123.2
Switch(config)# line con 0
Switch(config-line)# exec-timeout 0 0
Switch(config-line)# password keepout
Switch(config-line)# login
Switch(config-line)# line vty 5 15
Switch(config-line)# password keepout
Switch(config-line)# login
Switch(config-line)# line vty 5 15
Switch(config-line)# end
Switch# show running-config
Building configuration...

Current configuration : 1426 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service compress-config
!
hostname Switch
!
boot-start-marker
boot-end-marker
!
enable password cna
!
no aaa new-model
ip subnet-zero
ip domain-name cisco.com
!
vtp domain cnadoc
vtp mode transparent
!
crypto pki trustpoint TP-self-signed-913087
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-913087
 revocation-check none
 rsakeypair TP-self-signed-913087
!!
crypto pki certificate chain TP-self-signed-913087
 certificate self-signed 01
  3082028E 308201F7 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  52312B30 29060355 04031322 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 39313330 38373123 30210609 2A864886 F70D0109 02161456
  61646572 2D343531 302E6369 73636F2E 636F6D30 1E170D30 36303432 30323332
  3435305A 170D3230 30313031 30303030 30305A30 52312B30 29060355 04031322
  494F532D 53656C66 2D536967 6E65642D 43657274 69666963 6174652D 39313330
  38373123 30210609 2A864886 F70D0109 02161456 61646572 2D343531 302E6369
  73636F2E 636F6D30 819F300D 06092A86 4886F70D 01010105 0003818D 00308189
  02818100 F2C86FEA 49C37856 D1FA7CB2 9AFF748C DD443295 F6EC900A E83CDA8E
  FF8F9367 0A1E7A20 C0D3919F 0BAC2113 5EE37525 94CF24CF 7B313C01 BF177A73
  494B1096 B4D24729 E087B39C E44ED9F3 FCCD04BB 4AD3C6BF 66E0902D E234D08F
  E6F6C001 BAC80854 D4668160 9299FC73 C14A33F3 51A17BF5 8C0BEA07 3AC03D84
  889F2661 02030100 01A37430 72300F06 03551D13 0101FF04 05300301 01FF301F
  0603551D 11041830 16821456 61646572 2D343531 302E6369 73636F2E 636F6D30
  1F060355 1D230418 30168014 BB013B0D 00391D79 B628F2B3 74FC62B4 077AD908
  301D0603 551D0E04 160414BB 013B0D00 391D79B6 28F2B374 FC62B407 7AD90830
  0D06092A 864886F7 0D010104 05000381 81002963 26762EFA C52BA4B3 6E641A9D
  742CE404 E45FECB1 B5BD2E74 6F682476 A7C3DAA5 94393AE3 AA103B6E 5974F81B
  09DF16AE 7F9AE67C 5CB3D5B1 B945A5F3 36A8CC8C 8F142364 F849344D 5AE36410
  51182EB9 24A9330B 3583E1A3 79151470 D304C157 3417E240 52BE2A91 FC7BBEDE
  562BEDAD E6C46D9A F7FF3148 4CE9CEE1 5B17
  quit
!
!
!
power redundancy-mode redundant
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 2
!
interface GigabitEthernet1/1
 switchport access vlan 2
!
interface GigabitEthernet1/2
!
interface GigabitEthernet1/3
!
interface GigabitEthernet1/4
!
interface GigabitEthernet1/5
!
interface GigabitEthernet1/6
!
interface GigabitEthernet1/7
!
interface GigabitEthernet1/8
!
interface GigabitEthernet1/9
!
interface GigabitEthernet1/10
!
interface GigabitEthernet1/11
!
interface GigabitEthernet1/12
!
interface GigabitEthernet1/13
!
interface GigabitEthernet1/14
!
interface GigabitEthernet1/15
!
interface GigabitEthernet1/16
!
interface GigabitEthernet1/17
!
interface GigabitEthernet1/18
!
```

```
interface GigabitEthernet1/19
!
interface GigabitEthernet1/20
!
interface Vlan1
 no ip address
!
interface Vlan2
 ip address 123.123.123.1 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 123.123.123.2
ip http server
ip http secure-server
ip http max-connections 16
ip http timeout-policy idle 180 life 180 requests 25
!
line con 0
 password cna
 login
 stopbits 1
line vty 0 4
 password cna
 login
line vty 5 15
 password cna
 login
!
!
end

Switch#
```

Configuring Network Assistant in a Networked Switch in Cluster Mode

To configure Network Assistant on a networked switch in cluster mode, perform this task on the switch:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`enable password`** *`name`*	Enables password protection of configuration mode.
Step 3	`Switch(config)#` **`vtp domain`** *`name`*	Creates a VTP domain to manage VLANs and names.
Step 4	`Switch(config)#` **`cluster run`**	Launches the cluster on the cluster commander.
Step 5	`Switch(config)#` **`cluster enable`** *`cluster_name`*	Makes the switch the cluster commander.
Step 6	`Switch(config)#` **`vlan`** *`vlan_id`*	Creates a VLAN.
Step 7	`Switch(config-vlan)#` **`interface`** {**`vlan`** *`vlan_ID`* | {**`fastethernet`** | **`gigabitethernet`**} *`slot/interface`* | **`Port-channel`** *`number`*}	Selects the interface that will connect to your CNA-enabled PC.
Step 8	`Switch(config-if)#` **`switchport access vlan`** *`vlan_id`*	Enables the physical port to be in the specified VLAN.
Step 9	`Switch(config-if)#` **`interface`** {**`vlan`** *`vlan_ID`* | *`slot/interface`* | **`Port-channel`** *`number`*}	Select the VLAN instance for configuration.
Step 10	`Switch(config-if)#` **`ip address`** *`ip_address`*	Assigns an IP address to the SVI.
Step 11	`Switch(config-if)#` **`no shut`**	Enables the interface.

	Command	Purpose
Step 12	`Switch(config-if)# ip http server`	Starts the HTTP server so that Network Assistant can talk to the switch.
Step 13	`Switch(config)# ip http secure-server`	(Optionally) Enables the switch to accept HTTPS connections from Network Assistant.
Step 14	`Switch(config)# ip route a.b.c`	Establishes the route to the default router, usually supplied by the local Internet Provider. **Note** This line represents the only difference between the configuration for a standalone and a networked switch.
Step 15	`Switch(config)# line con 0`	Select the console port to perform the configuration.
Step 16	`Switch(config-line)# exec-timeout x y`	Configures an automatic session logout if no keyboard input or output is displayed on the terminal.
Step 17	`Switch(config-line)# password password`	Specifies a password for the console port.
Step 18	`Switch(config-line)# login`	Allows login to the console port.
Step 19	`Switch(config-line)# line vty x y`	Creates additional VTY lines for CNA to access the switch.
Step 20	`Switch(config-line)# password password`	Specifies a password for the switch.
Step 21	`Switch(config-line)# login`	Allows login to the switch.
Step 22	`Switch(config-line)# line vty x y`	Creates additional VTY lines for CNA to access the switch.
Step 23	`Switch(config-line)# password password`	Specifies a password for the switch.
Step 24	`Switch(config-line)# login`	Allows login to the switch.
Step 25	`Switch(config-line)# end`	Returns to privileged EXEC mode.
Step 26	`Switch# show running-config| include http`	Verifies that the HTTP server is enabled.

This example shows how to configure Network Assistant on a networked switch in cluster mode:

```
Switch# configure terminal
Switch(config)# vtp domain cnadoc
Switch(config)# cluster run
Switch(config)# cluster enable cnadoc
Switch(config)# vlan 10
Switch(config-vlan)# interface GigabitEthernet 2/1
Switch(config-if)# switchport access vlan 10
Switch(config-if)# interface vlan10
Switch(config-if)# ip address aa.bb.cc.dd
Switch(config-if)# no shut
Switch(config-if)# ip http server
Switch(config-if)# ip http secure-server
Switch(config)# ip route 0.0.0.0 0.0.0.0 123.123.123.2
Switch(config)# line con 0
Switch(config-line)# exec-timeout 0 0
Switch(config-line)# password keepout
Switch(config-line)# login
Switch(config-line)# line vty 5 15
Switch(config-line)# password keepout
Switch(config-line)# login
Switch(config-line)# line vty 5 15
Switch(config-line)# end
Switch# show running-config
Building configuration...
```

```
Current configuration : 1469 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service compress-config
!
hostname Switch
!
boot-start-marker
boot-end-marker
!
enable password cna
!
no aaa new-model
ip subnet-zero
!
vtp domain cnadoc
vtp mode transparent
cluster run
cluster enable cnadoccluster 0
!
!
!
!
!
power redundancy-mode redundant
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 2
!
interface GigabitEthernet1/1
 switchport access vlan 2
!
interface GigabitEthernet1/2
!
interface GigabitEthernet1/3
!
interface GigabitEthernet1/4
!
interface GigabitEthernet1/5
!
interface GigabitEthernet1/6
!
interface GigabitEthernet1/7
!
interface GigabitEthernet1/8
!
interface GigabitEthernet1/9
!
interface GigabitEthernet1/10
!
interface GigabitEthernet1/11
!
```

```
interface GigabitEthernet1/12
!
interface GigabitEthernet1/13
!
interface GigabitEthernet1/14
!
interface GigabitEthernet1/15
!
interface GigabitEthernet1/16
!
interface GigabitEthernet1/17
!
interface GigabitEthernet1/18
!
interface GigabitEthernet1/19
!
interface GigabitEthernet1/20
!
interface Vlan1
 no ip address
!
interface Vlan2
 ip address 123.123.123.1 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 123.123.123.2
ip http server
no ip http secure-server
!
!
!
line con 0

Switch#
```

CHAPTER 13

Configuring VLANs, VTP, and VMPS

This chapter describes VLANs on Catalyst 4500 series switches. It also describes how to enable the VLAN Trunking Protocol (VTP) and to configure the Catalyst 4500 series switch as a VMPS client.

This chapter includes the following major sections:

- VLANs, page 13-1
- VLAN Trunking Protocol, page 13-7
- VLAN Membership Policy Server, page 13-16

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

VLANs

This section includes the following major subsections:

- Overview of VLANs, page 13-1
- VLAN Configuration Guidelines and Restrictions, page 13-3
- VLAN Default Configuration, page 13-4
- Configuring VLANs, page 13-4

Overview of VLANs

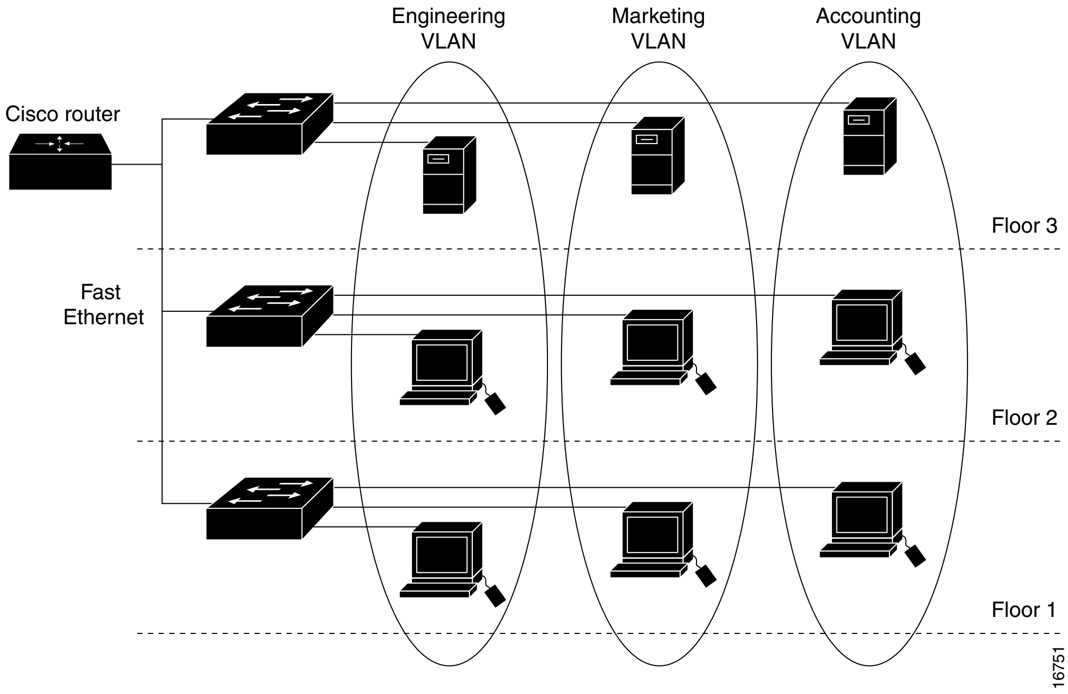
A VLAN is a group of devices on one or more LANs that are configured to communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible.

VLANs define broadcast domains in a Layer 2 network. A broadcast domain is the set of all devices that will receive broadcast frames originating from any device within the set. Broadcast domains are typically bounded by routers because routers do not forward broadcast frames. Layer 2 switches create broadcast domains based on the configuration of the switch. Switches are multiport bridges that allow you to create multiple broadcast domains. Each broadcast domain is like a distinct virtual bridge within a switch.

You can define one or many virtual bridges within a switch. Each virtual bridge you create in the switch defines a new broadcast domain (VLAN). Traffic cannot pass directly to another VLAN (between broadcast domains) within the switch or between two switches. To interconnect two different VLANs, you must use routers or Layer 3 switches. See the “Overview of Layer 3 Interfaces” section on page 26-1 for information on inter-VLAN routing on Catalyst 4500 series switches.

Figure 13-1 shows an example of three VLANs that create logically defined networks.

Figure 13-1 Sample VLANs



VLANs are often associated with IP subnetworks. For example, all of the end stations in a particular IP subnet belong to the same VLAN. Traffic between VLANs must be routed. You must assign LAN interface VLAN membership on an interface-by-interface basis (this is known as interface-based or static VLAN membership).

You can set the following parameters when you create a VLAN in the management domain:

- VLAN number
- VLAN name
- VLAN type
- VLAN state (active or suspended)
- Maximum transmission unit (MTU) for the VLAN
- Security Association Identifier (SAID)
- VLAN number to use when translating from one VLAN type to another

Note When the software translates from one VLAN type to another, it requires a different VLAN number for each media type.

VLAN Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when creating and modifying VLANs in your network:

- Before creating a VLAN, put the Catalyst 4500 series switch in VTP server mode or VTP transparent mode. If the Catalyst 4500 series switch is a VTP server, you must define a VTP domain. For information on configuring VTP, see the section VLAN Trunking Protocol, page 13-7.
- The Cisco IOS **end** command is not supported in VLAN database mode.
- You cannot use **Ctrl-Z** to exit VLAN database mode.

VLAN Ranges

Note You must enable the extended system ID to use 4094 VLANs. See the "Understanding the Bridge ID" section on page 17-2.

With Cisco IOS Release 12.2(25)EWA and later, Catalyst 4500 series switches support 4096 VLANs in compliance with the IEEE 802.1Q standard. These VLANs are organized into three ranges: reserved, normal, and extended.

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 13-1 describes the uses for VLAN ranges.

Table 13-1 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	Used 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. When configuring extended-range VLANs, note the following: • Layer 3 ports and some software features require internal VLANs. Internal VLANs are allocated from 1006 and up. You cannot use a VLAN that has been allocated for such use. To display the VLANs used internally, enter the **show vlan internal usage** command. • Switches running Catalyst product family software 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 product family software. • You must enable the extended system ID to use extended range VLANs. See the "Enabling the Extended System ID" section on page 17-8.	No

Configurable Normal-Range VLAN Parameters

Note Ethernet VLANs 1 and 1006 through 4094 use only default values.

You can configure the following parameters for VLANs 2 through 1001:

- VLAN name
- VLAN type
- VLAN state (active or suspended)
- SAID
- STP type for VLANs

VLAN Default Configuration

Table 13-2 shows the default VLAN configuration values.

Table 13-2 Ethernet VLAN Defaults and Ranges

Parameter	Default	Valid Values
VLAN ID	1	1–4094
VLAN name	VLAN*x*, where *x* is a number assigned by the software.	No range
802.10 SAID	100,001	1–4,294,967,294
MTU size	1500	1500–18,190
Translational bridge 1	1002	0–1005
Translational bridge 2	1003	0–1005
VLAN state	active	active; suspend; shutdown

Note Catalyst 4500 series switches do not support Token Ring or FDDI media. The switch does not forward FDDI, FDDI-NET, TrCRF, or TrBRF traffic, but it does propagate the VLAN configuration via VTP. The software reserves parameters for these media types, but they are not truly supported.

Configuring VLANs

Note Before you configure VLANs, you must use VLAN Trunking Protocol (VTP) to maintain global VLAN configuration information for your network. For complete information on VTP, see the "VLAN Trunking Protocol" section on page 7.

Note VLANs support a number of parameters that are not discussed in detail in this section. For complete information, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

Note 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 the following sections and in the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

The following sections describe how to configure VLANs:

- Configuring VLANs in Global Configuration Mode, page 13-5
- Assigning a Layer 2 LAN Interface to a VLAN, page 13-7

Configuring VLANs in Global Configuration Mode

If the switch is in VTP server or transparent mode (see the "VLAN Trunking Protocol" section on page 13-7), you can configure VLANs in global and VLAN configuration modes. When you configure VLANs in global and config-vlan configuration modes, the VLAN configuration is saved in the **vlan.dat** files, not the **running-config** or **startup-config** 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, the **show running-config** and **show startup-config** commands 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 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 contents of the file by entering the **show running-config** command.

User-configured VLANs have unique IDs from 1 to 4094. To create a VLAN, enter the **vlan** command with an unused ID. To verify whether a particular ID is in use, enter the **show vlan id** *ID* command**.** To modify a VLAN, enter the **vlan** command for an existing VLAN.

See the "VLAN Default Configuration" section on page 13-4 for the list of default parameters that are assigned when you create a VLAN. If you do not use the **media** keyword when specifying the VLAN type, the VLAN is an Ethernet VLAN.

To create a VLAN, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.

Step 2	`Switch(config)# vlan vlan_ID`
	`Switch(config-vlan)#`
		Adds an Ethernet VLAN.
		Note You cannot delete the default VLANs for these media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005.
		When you delete a VLAN, any LAN interfaces configured as access ports assigned to that VLAN become inactive. They remain associated with the VLAN (and thus inactive) until you assign them to a new VLAN.
		You can use the **no** keyword to delete a VLAN.
		When the prompt reads `Switch(config-vlan)#,` you are in vlan-configuration mode. If you wish to change any of the parameters for the newly created VLAN, use this mode.

Step 3	`Switch(config-vlan)# end`	Returns to enable mode from vlan-configuration mode.
Step 4	`Switch# show vlan [id	name] vlan_name`	Verifies the VLAN configuration.

When you create or modify an Ethernet VLAN, note the following:

- Because Layer 3 ports and some software features require internal VLANs allocated from 1006 and up, configure extended-range VLANs starting with 4094 and work downward.
- 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.

This example shows how to create an Ethernet VLAN in global configuration mode and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 3
Switch(config-vlan)# end
Switch# 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
------- --------- ----------------- -------------------------------------------
Switch#
```

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 interfaces to the VLAN.

Note Make sure you assign LAN interfaces to a VLAN of the proper type. Assign Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces to Ethernet-type VLANs.

To assign one or more LAN interfaces to a VLAN, complete the procedures in the "Configuring Ethernet Interfaces for Layer 2 Switching" section on page 15-5.

VLAN Trunking Protocol

This section describes the VLAN Trunking Protocol (VTP) on the Catalyst 4500 series switches.

This section includes the following major subsections:

- Overview of VTP, page 13-7
- VTP Configuration Guidelines and Restrictions, page 13-11
- VTP Default Configuration, page 13-11
- Configuring VTP, page 13-12

Overview of VTP

VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs within a VTP domain. A VTP domain (also called a VLAN management domain) is made up of one or more network devices that share the same VTP domain name and that are interconnected with trunks. VTP minimizes misconfigurations and configuration inconsistencies that can result in a number of problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations.

Before you create VLANs, you must decide whether you want 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. For details on configuring VLANs, see VLANs, page 13-1

These sections describe how VTP works:

- Understanding the VTP Domain, page 13-8
- Understanding VTP Modes, page 13-8
- Understanding VTP Advertisements, page 13-8
- Understanding VTP Version 2, page 13-9
- Understanding VTP Pruning, page 13-9

Understanding the VTP Domain

A VTP 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 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 4500 series switch is in VTP transparent 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. You cannot create or modify VLANs on a VTP server until the management domain name is specified or learned.

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 Inter-Switch Link (ISL) and IEEE 802.1Q trunk connections.

VTP maps VLANs dynamically across multiple LAN types with unique names and internal index associations. Mapping eliminates unnecessary device administration for network administrators.

Understanding VTP Modes

You can configure a Catalyst 4500 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.
- 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 on their trunking LAN interfaces. VTP transparent is the default mode.

Note Catalyst 4500 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 interface 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 Catalyst 4500 series switches do not support Token Ring or FDDI media. The switch does not forward FDDI, FDDI-Net, Token Ring Concentrator Relay Function [TrCRF], or Token Ring Bridge Relay Function [TrBRF] traffic, but it does propagate the VLAN configuration via VTP.

VTP version 2 supports the following features, which are not supported in version 1:

- Token Ring support—VTP version 2 supports Token Ring LAN switching and VLANs (TrBRF and TrCRF).
- 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, and 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 either support VTP pruning or, on devices that do not support VTP pruning, you must manually configure the VLANs allowed on trunks.

Figure 13-2 shows a switched network without VTP pruning enabled. Interface 1 on Switch 1 and Interface 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 interfaces in the Red VLAN.

You can enable pruning globally on the Catalyst 4500 series switch (see the "Enabling VTP Pruning" section on page 13-12).

Figure 13-2 Flooding Traffic without VTP Pruning

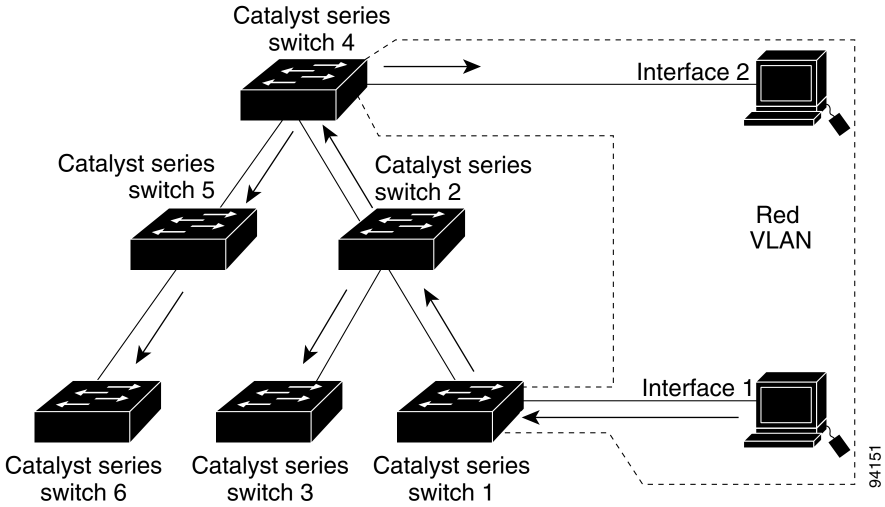
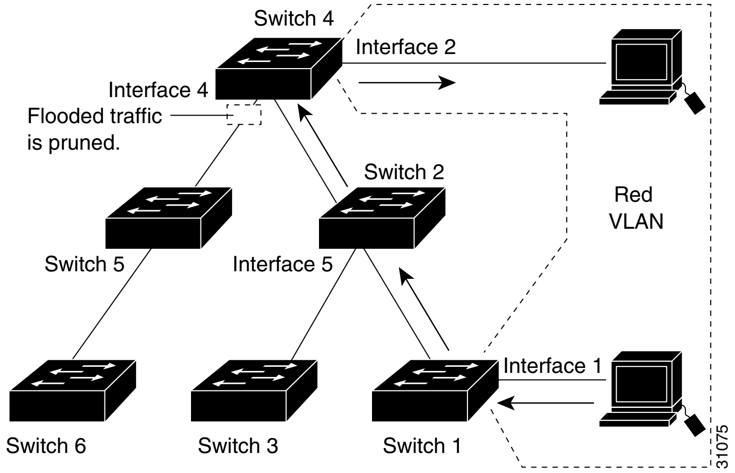


Figure 13-3 shows the same switched network with VTP pruning enabled. The broadcast traffic from Switch 1 is not forwarded to Switches 3, 5, and 6 because traffic for the Red VLAN has been pruned on the links indicated (Interface 5 on Switch 2 and Interface 4 on Switch 4).

Figure 13-3 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 eligible for pruning. VTP pruning does not prune traffic from pruning-ineligible VLANs. VLAN 1 is always ineligible for pruning; traffic from VLAN 1 cannot be pruned.

To configure VTP pruning on a trunking LAN interface, use the **switchport trunk pruning vlan** command. VTP pruning operates when a LAN interface is trunking. You can set VLAN pruning eligibility regardless of whether VTP pruning is enabled or disabled for the VTP domain, whether any given VLAN exists, and regardless of whether the LAN interface is currently trunking.

VTP Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when implementing VTP in your network:

- 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 VTP is 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 if VTP version 2 is disabled on the VTP version 2-capable network device (VTP version 2 is disabled by default).
- Do not enable VTP version 2 on a network device unless all of the network devices in the same VTP domain are version 2-capable. When you enable VTP version 2 on a server, all of the version 2-capable network devices in the domain enable VTP version 2.
- Enabling or disabling VTP pruning on a VTP server enables or disables VTP pruning for the entire management domain.
- Configuring VLANs as eligible for pruning on a Catalyst 4500 series switch affects pruning eligibility for those VLANs on that switch only, not on all network devices in the VTP domain.

VTP Default Configuration

Table 13-3 shows the default VTP configuration.

Table 13-3 VTP Default Configuration

Feature	Default Value
VTP domain name	Null
VTP mode	Transparent
VTP version 2 enable state	Version 2 is disabled
VTP password	None
VTP pruning	Disabled

The default VTP mode for newly manufactured Catalyst 4500 supervisor engines, Catalyst 4900 series switches, and the Cisco ME 4924-10GE switche is transparent. Deleting vlan.dat or issuing the **erase cat4000_flash:** command, and resetting the switch will change the VTP mode to server.

Configuring VTP

The following sections describe how to configure VTP:

- Configuring VTP Global Parameters, page 13-12
- Configuring the Switch as a VTP Server, page 13-13
- Configuring the Switch as a VTP Client, page 13-14
- Disabling VTP (VTP Transparent Mode), page 13-15
- Displaying VTP Statistics, page 13-15

Configuring VTP Global Parameters

The following sections describe configuring the VTP global parameters:

- Configuring a VTP Password, page 13-12
- Enabling VTP Pruning, page 13-12
- Enabling VTP Version 2, page 13-13

Configuring a VTP Password

To configure the VTP password, perform this task:

Command	Purpose
`Switch# [no] vtp password` *`password_string`*	Sets a password for the VTP domain. The password can be from 8 to 64 characters. Uses the **no** keyword to remove the password.

This example shows how to configure a VTP password:

```
Switch# vtp password WATER
Setting device VLAN database password to WATER.
Switch#show vtp password
VTP Password:WATER
Switch#
```

Enabling VTP Pruning

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

	Command	Purpose
Step 1	`Switch# [no] vtp pruning`	Enables VTP pruning in the management domain. Use the **no** keyword to disable VTP pruning in the management domain.
Step 2	`Switch# show vtp status`	Verifies the configuration.

This example shows how to enable VTP pruning in the management domain:

```
Switch# vtp pruning
Pruning switched ON
```

This example shows how to verify the configuration:

```
Switch# show vtp status | include Pruning
VTP Pruning Mode                : Enabled
Switch#
```

Enabling VTP Version 2

By default, VTP version 2 is disabled on VTP version 2-capable network devices. When you enable VTP version 2 on a server, 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.

To enable VTP version 2, perform this task:

	Command	Purpose
Step 1	`Switch# [no] vtp version {1 | 2}`	Enables VTP version 2. Use the **no** keyword to revert to the default.
Step 2	`Switch# show vtp status`	Verifies the configuration.

This example shows how to enable VTP version 2:

```
Switch# vtp version 2
V2 mode enabled.
Switch#
```

This example shows how to verify the configuration:

```
Switch# show vtp status | include V2
VTP V2 Mode                     : Enabled
Switch#
```

Configuring the Switch as a VTP Server

To configure the Catalyst 4500 series switch as a VTP server, perform this task:

	Command	Purpose
Step 1	`Switch# configuration terminal`	Enters configuration mode.
Step 2	`Switch(config)# vtp mode server`	Configures the switch as a VTP server.
Step 3	`Switch(config)# vtp domain domain_name`	Defines the VTP domain name, which can be up to 32 characters long.
Step 4	`Switch(config)# end`	Exits VLAN configuration mode.
Step 5	`Switch# show vtp status`	Verifies the configuration.

This example shows how to configure the switch as a VTP server:

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

This example shows how to verify the configuration:

```
Switch# show vtp status
VTP Version                     : running VTP1 (VTP2 capable)
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)
Switch#
```

Configuring the Switch as a VTP Client

To configure the Catalyst 4500 series switch as a VTP client, perform this task:

	Command	Purpose
Step 1	`Switch# configuration terminal`	Enters configuration mode.
Step 2	`Switch(config)# [no] vtp mode client`	Configure the switch as a VTP client. Use the **no** keyword to return to enable server mode.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show vtp status`	Verifies the configuration.

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

```
Switch# configuration terminal
Switch(config)# vtp mode client
Setting device to VTP CLIENT mode.
Switch(config)# exit
Switch#
```

This example shows how to verify the configuration:

```
Switch# show vtp status
VTP Version                     : running VTP1 (VTP2 capable)
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
Switch#
```

Disabling VTP (VTP Transparent Mode)

To disable VTP on the Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configuration terminal`**	Enters configuration mode.
Step 2	`Switch(config)# [`**`no`**`]` **`vtp mode transparent`**	Disables VTP on the switch. Use the **no** keyword to enable server mode.
Step 3	`Switch(config)#` **`end`**	Exits configuration mode.
Step 4	`Switch#` **`show vtp status`**	Verifies the configuration.

This example shows how to disable VTP on the switch:

```
Switch# configuration terminal
Switch(config)# vtp transparent
Setting device to VTP mode.
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# 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
Switch#
```

Displaying VTP Statistics

To display VTP statistics, including VTP advertisements sent and received and VTP errors, perform this task:

Command	Purpose
`Switch#` **`show vtp counters`**	Displays VTP statistics.

This example shows how to display VTP statistics:

```
Switch# show vtp counters
VTP statistics:
Summary advertisements received    : 7
Subset advertisements received     : 5
```

```
Request advertisements received    : 0
Summary advertisements transmitted : 997
Subset advertisements transmitted  : 13
Request advertisements transmitted : 3
Number of config revision errors   : 0
Number of config digest errors     : 0
Number of V1 summary errors        : 0

VTP pruning statistics:

Trunk            Join Transmitted Join Received    Summary advts received from
                                                   non-pruning-capable device
---------------- ---------------- ---------------- ---------------------------
Fa5/8               43071            42766            5
```

VLAN Membership Policy Server

This section describes how to configure dynamic port VLAN membership through the VLAN Membership Policy Server (VMPS).

This section includes the following subsections:

- Overview of VMPS, page 13-16
- Overview of VMPS Clients, page 13-19
- Dynamic Port VLAN Membership Configuration Example, page 13-25
- VMPS Database Configuration File Example, page 13-28

Overview of VMPS

These subsections describe what a VMPS server does and how it operates:

- Understanding the VMPS Server, page 13-16
- Security Modes for VMPS Server, page 13-17
- Fallback VLAN, page 13-18
- Illegal VMPS Client Requests, page 13-19

Understanding the VMPS Server

A VLAN Membership Policy Server (VMPS) provides a centralized server for selecting the VLAN for a port dynamically based on the MAC address of the device connected to the port. When the host moves from a port on one switch in the network to a port on another switch in the network, that switch dynamically assigns the new port to the proper VLAN for that host.

A Catalyst 4500 series switch running Cisco IOS software does not support the functionality of a VMPS. It can only function as a VLAN Query Protocol (VQP) client, which communicates with a VMPS through the VQP. For VMPS functionality, you need to use a Catalyst 4500 series switch (or Catalyst 6500 series switch) running Catalyst operating system (OS) software.

VMPS uses a UDP port to listen to VQP requests from clients, so, it is not necessary for VMPS clients to know if the VMPS resides on a local or remote device on the network. Upon receiving a valid request from a VMPS client, a VMPS server searches its database for an entry of a MAC-address to VLAN mapping.

In response to a request, the VMPS takes one of the following actions:

- If the assigned VLAN is restricted to a group of ports, the VMPS verifies the requesting port against this group and responds as follows:
  - If the VLAN is allowed on the port, the VMPS sends the VLAN name to the client in response.
  - If the VLAN is not allowed on the port and the VMPS is not in secure mode, the VMPS sends an “access-denied” response.
  - If the VLAN is not allowed on the port and the VMPS is in secure mode, the VMPS sends a “port-shutdown” response.
- If the VLAN in the database does not match the current VLAN on the port and there are active hosts on the port, the VMPS sends an “access-denied” (open), a “fallback VLAN name” (open with fallback VLAN configured), a “port-shutdown” (secure), or a “new VLAN name” (multiple) response, depending on the secure mode setting of the VMPS.

  If the switch receives an “access-denied” response from the VMPS, the switch continues to block traffic from the MAC address to or from the port. The switch continues to monitor the packets directed to the port and sends a query to the VMPS when it identifies a new address. If the switch receives a “port-shutdown” response from the VMPS, the switch disables the port. The port must be manually re-enabled by using the CLI, Cisco Visual Switch Manager (CVSM), or SNMP.

  You can also use an explicit entry in the configuration table to deny access to specific MAC addresses for security reasons. If you enter the **none** keyword for the VLAN name, the VMPS sends an “access-denied” or “port-shutdown” response.

For more information on a Catalyst 6500 series switch VMPS running Catalyst operating system software, refer to the
“Configuring Dynamic Port VLAN Membership with VMPS” chapter at the URL:
http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_3/confg_gd/vmps.htm

Security Modes for VMPS Server

VMPS operates in three different modes. The way a VMPS server responds to illegal requests depends on the mode in which the VMPS is configured:

- Open Mode, page 13-17
- Secure Mode, page 13-18
- Multiple Mode, page 13-18

Open Mode

If no VLAN is assigned to this port, VMPS verifies the requesting MAC address against this port:

- If the VLAN associated with this MAC address is allowed on the port, the VLAN name is returned to the client.
- If the VLAN associated with this MAC address is not allowed on the port, the host receives an “access denied” response.

If a VLAN is already assigned to this port, VMPS verifies the requesting MAC address against this port:

- If the VLAN associated with this MAC address in the database does not match the current VLAN assigned on the port, and a fallback VLAN name is configured, VMPS sends the fallback VLAN name to the client.
- If a VLAN associated with this MAC address in the database does not match the current VLAN assigned on the port, and a fallback VLAN name is not configured, the host receives an "access denied" response.

Secure Mode

If no VLAN is assigned to this port, VMPS verifies the requesting MAC address against this port:

- If the VLAN associated with this MAC address is allowed on the port, the VLAN name is returned to the client.
- If the VLAN associated with this MAC address is not allowed on the port, the port is shut down.

If a VLAN is already assigned to this port, VMPS verifies the requesting MAC address against this port:

- If a VLAN associated with this MAC address in the database does not match the current VLAN assigned on the port, the port is shutdown, even if a fallback VLAN name is configured.

Multiple Mode

Multiple hosts (MAC addresses) can be active on a dynamic port if they are all in the same VLAN. If the link fails on a dynamic port, the port returns to the unassigned state. Any hosts that come online through the port are checked again with VMPS before the port is assigned to a VLAN.

If multiple hosts connected to a dynamic port belong to different VLANs, the VLAN matching the MAC address in the last request is returned to the client provided that multiple mode is configured on the VMPS server.

Note Although Catalyst 4500 series and Catalyst 6500 series switches running Catalyst operating system software support VMPS in all three operation modes, the User Registration Tool (URT) supports open mode only.

Fallback VLAN

You can configure a fallback VLAN name on a VMPS server.

If no VLAN has been assigned to this port, VMPS compares the requesting MAC address to this port:

- If you connect a device with a MAC address that is not in the database, the VMPS sends the fallback VLAN name to the client.
- If you do not configure a fallback VLAN name and the MAC address does not exist in the database, the VMPS sends an "access-denied" response.

If a VLAN is already assigned to this port, VMPS compares the requesting MAC address to this port:

- If the VMPS is in secure mode, it sends a "port-shutdown" response, whether or not a fallback VLAN has been configured on the server.

Illegal VMPS Client Requests

Two examples of illegal VMPS client requests are as follows:

- When a MAC-address mapping is not present in the VMPS database and "no fall back" VLAN is configured on the VMPS.
- When a port is already assigned a VLAN (and the VMPS mode is not "multiple") but a second VMPS client request is received on the VMPS for a different MAC-address.

Overview of VMPS Clients

The following subsections describe how to configure a switch as a VMPS client and configure its ports for dynamic VLAN membership.

The following topics are included:

- Understanding Dynamic VLAN Membership, page 13-19
- Default VMPS Client Configuration, page 13-20
- Configuring a Switch as a VMPS Client, page 13-20
- Administering and Monitoring the VMPS, page 13-23
- Troubleshooting Dynamic Port VLAN Membership, page 13-24

Understanding Dynamic VLAN Membership

When a port is configured as "dynamic," it receives VLAN information based on the MAC-address that is on the port. The VLAN is not statically assigned to the port; it is dynamically acquired from the VMPS based on the MAC-address on the port.

A dynamic port can belong to one VLAN only. When the link becomes active, the switch does not forward traffic to or from this port until the port is assigned to a VLAN. The source MAC address from the first packet of a new host on the dynamic port is sent to the VMPS as part of the VQP request, which attempts to match the MAC address to a VLAN in the VMPS database. If there is a match, the VMPS sends the VLAN number for that port. If there is no match, the VMPS either denies the request or shuts down the port (depending on the VMPS security mode setting). See the "Overview of VMPS" section on page 13-16 for a complete description of possible VMPS responses.

Multiple hosts (MAC addresses) can be active on a dynamic port if all are in the same VLAN. If the link goes down on a dynamic port, the port returns to the unassigned state and does not belong to a VLAN. Any hosts that come online through the port are checked again with the VMPS before the port is assigned to a VLAN.

For this behavior to work, the client device must be able to reach the VMPS. A VMPS client sends VQP requests as UDP packets, trying a certain number of times before giving up. For details on how to set the retry interval, refer to section "Configuring the Retry Interval" on page 23.

The VMPS client also periodically reconfirms the VLAN membership. For details on how to set the reconfirm frequency, refer to section "Administering and Monitoring the VMPS" on page 23.

A maximum of 50 hosts are supported on a given port at any given time. Once this maximum is exceeded, the port is shut down, irrespective of the operating mode of the VMPS server.

Note The VMPS shuts down a dynamic port if more than 50 hosts are active on that port.

Default VMPS Client Configuration

Table 13-4 shows the default VMPS and dynamic port configuration on client switches.

Table 13-4 Default VMPS Client and Dynamic Port Configuration

Feature	Default Configuration
VMPS domain server	None
VMPS reconfirm interval	60 minutes
VMPS server retry count	3
Dynamic ports	None configured

Configuring a Switch as a VMPS Client

This section contains the following topics:

- Configuring the IP Address of the VMPS Server, page 13-20
- Configuring Dynamic Access Ports on a VMPS Client, page 13-21
- Reconfirming VLAN Memberships, page 13-22
- Configuring Reconfirmation Interval, page 13-22
- Reconfirming VLAN Memberships, page 13-22

Configuring the IP Address of the VMPS Server

To configure a Catalyst 4500 series switch as a VMPS client, you must enter the IP address or hostname of the switch acting as the VMPS.

To define the primary and secondary VMPS on a Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# vmps server {ipaddress | hostname} primary`	Specifies the IP address or hostname of the switch acting as the primary VMPS server.
Step 3	`Switch(config)# vmps server {ipaddress | hostname}`	Specifies the IP address or hostname of the switch acting as a secondary VMPS server.
Step 4	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show vmps`	Verifies the VMPS server entry.

This example shows how to define the primary and secondary VMPS devices:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vmps server 172.20.128.179 primary
Switch(config)# vmps server 172.20.128.178
Switch(config)# end
```

Note You can configure up to four VMPS servers using this CLI on the VMPS client.

```
Switch# show vmps
VQP Client Status:
--------------------
VMPS VQP Version:   1
Reconfirm Interval: 60 min
Server Retry Count: 3
VMPS domain server: 172.20.128.179 (primary, current)
                    172.20.128.178

Reconfirmation status
---------------------
VMPS Action:         No Dynamic Port
```

Configuring Dynamic Access Ports on a VMPS Client

To configure a dynamic access port on a VMPS client switch, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface** *interface*	Enters interface configuration mode and specifies the port to be configured.
Step 3	Switch(config-if)# **switchport mode access**	Sets the port to access mode.
Step 4	Switch(config-if)# **switchport access vlan dynamic**	Configures the port as eligible for dynamic VLAN access.
Step 5	Switch(config-if)# **end**	Returns to privileged EXEC mode.
Step 6	Switch# **show interface** *interface* **switchport**	Verifies the entry.

This example shows how to configure a dynamic access port and to verify the entry:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa1/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan dynamic
Switch(config-if)# end

Switch# show interface fa1/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative mode: dynamic auto
Operational Mode: dynamic access
Administrative Trunking Encapsulation: isl
Operational Trunking Encapsulation: isl
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: NONE
Pruning VLANs Enabled: NONE
```

Voice Ports

If a VVID (voice VLAN ID) is configured on a dynamic access port, the port can belong to both an access VLAN and a voice VLAN. Consequently, an access port configured for connecting an IP phone can have separate VLANs for the following:

- Data traffic to and from the PC that is connected to the switch through the access port of the IP phone (access VLAN)
- Voice traffic to and from the IP phone (voice VLAN)

Reconfirming VLAN Memberships

To confirm the dynamic port VLAN membership assignments that the switch has received from the VMPS, perform this task:

	Command	Purpose
Step 1	`Switch# vmps reconfirm`	Reconfirms dynamic port VLAN membership.
Step 2	`Switch# show vmps`	Verifies the dynamic VLAN reconfirmation status.

Configuring Reconfirmation Interval

VMPS clients periodically reconfirm the VLAN membership information received from the VMPS. You can set the number of minutes the VMPS client waits before reconfirming the VLAN-to-MAC-address assignments.

To configure the reconfirmation interval, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# vmps reconfirm` *minutes*	Specifies the number of minutes between reconfirmations of the dynamic VLAN membership.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show vmps`	Verifies the dynamic VLAN reconfirmation status.

This example shows how to change the reconfirmation interval to 60 minutes and verify the change:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vmps reconfirm 60
Switch(config)# end
Switch# show vmps
VQP Client Status:
--------------------
VMPS VQP Version:   1
Reconfirm Interval: 60 min
Server Retry Count: 10
VMPS domain server: 172.20.130.50 (primary, current)

Reconfirmation status
---------------------
VMPS Action:         No Host
```

Configuring the Retry Interval

You can set the number of times that the VMPS client attempts to contact the VMPS before querying the next server.

To configure the retry interval, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# vmps retry count`	Specifies the retry count for the VPQ queries. Default is 3. Range is from 1 to 10.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show vmps`	Verifies the retry count.

This example shows how to change the retry count to 5 and to verify the change:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vmps retry 5
Switch(config)# end

Switch# show vmps
VQP Client Status:
--------------------
VMPS VQP Version:   1
Reconfirm Interval: 60 min
Server Retry Count: 5
VMPS domain server: 172.20.130.50 (primary, current)

Reconfirmation status
---------------------
VMPS Action:         No Host
```

Administering and Monitoring the VMPS

You can display the following information about the VMPS with the **show vmps** command:

VQP Version — The version of VQP used to communicate with the VMPS. The switch queries the VMPS using VQP Version 1.

Reconfirm Interval — The number of minutes the switch waits before reconfirming the VLAN-to-MAC-address assignments.

Server Retry Count — The number of times VQP resends a query to the VMPS. If no response is received after this many tries, the switch starts to query the secondary VMPS.

VMPS domain server — The IP address of the configured VLAN membership policy servers. The switch currently sends queries to the one marked "current." The one marked "primary" is the primary server.

VMPS Action — The result of the most-recent reconfirmation attempt. This action can occur automatically when the reconfirmation interval expired, or you can force it by entering the **vmps reconfirm** command or its CVSM or SNMP equivalent.

The following example shows how to display VMPS information:

```
Switch# show vmps
VQP Client Status:
--------------------
VMPS VQP Version:   1
Reconfirm Interval: 60 min
Server Retry Count: 3
VMPS domain server:

Reconfirmation status
---------------------
VMPS Action:         other

The following example shows how to display VMPS statistics:
Switch# show vmps statistics
VMPS Client Statistics
----------------------
VQP  Queries:              0
VQP  Responses:            0
VMPS Changes:              0
VQP  Shutdowns:            0
VQP  Denied:               0
VQP  Wrong Domain:         0
VQP  Wrong Version:        0
VQP  Insufficient Resource: 0
```

Note Refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* for details on VMPS statistics.

Troubleshooting Dynamic Port VLAN Membership

VMPS errdisables a dynamic port under the following conditions:

- The VMPS is in secure mode, and it will not allow the host to connect to the port. The VMPS errdisables the port to prevent the host from connecting to the network.
- More than 50 active hosts reside on a dynamic port.

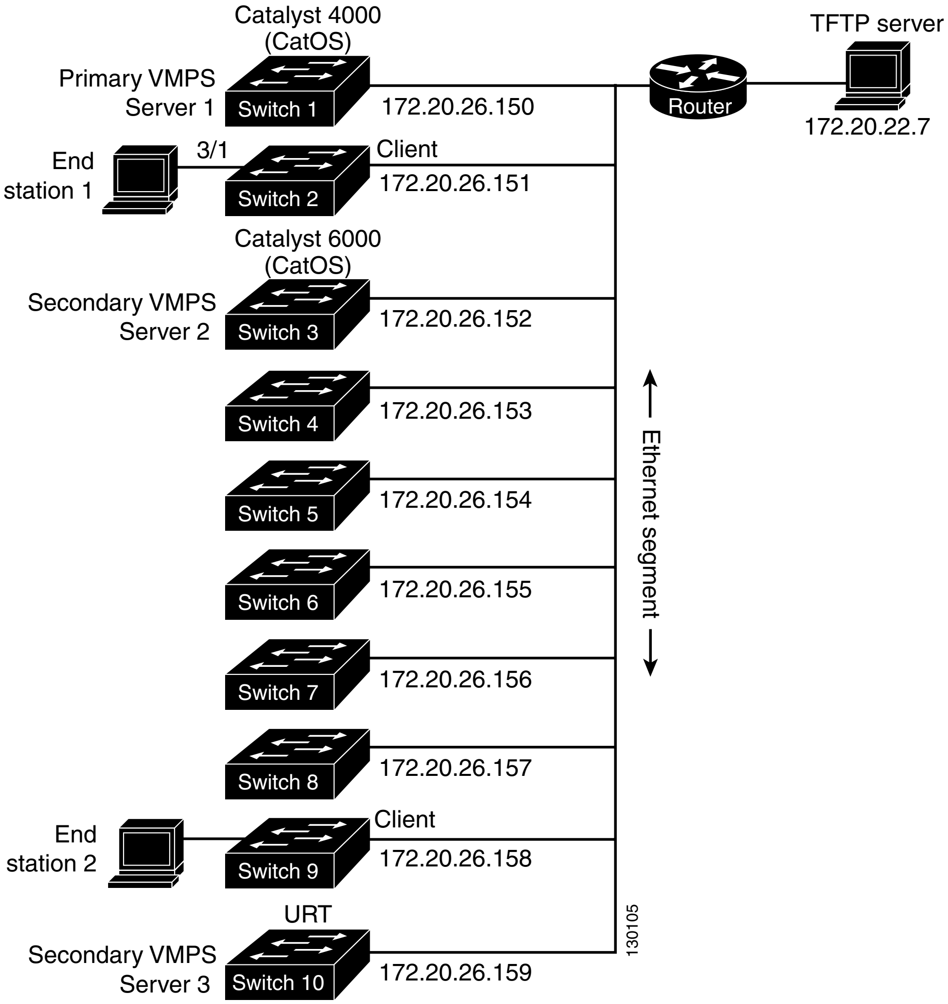
For information on how to display the status of interfaces in error-disabled state, refer to Chapter 7, "Checking Port Status and Connectivity." To recover an errdisabled port, use the **errdisable recovery cause vmps** global configuration command.

Dynamic Port VLAN Membership Configuration Example

Figure 13-4 on page 13-25 shows a network with a VMPS servers and VMPS client switches with dynamic ports. In this example, these assumptions apply:

- The VMPS server and the VMPS client are separate switches.
- The Catalyst 4000 family Switch 1 (running CatOS) is the primary VMPS server.
- The Catalyst 6000 family Switch 3 (running CatOS) and the URT are secondary VMPS servers.
- End stations are connected to these clients:
  - Catalyst 4500 series XL Switch 2 (running Catalyst IOS)
  - Catalyst 4500 series XL Switch 9 (running Catalyst IOS)
- The database configuration file is called Bldg-G.db and is stored on the TFTP server with the IP address 172.20.22.7.

Figure 13-4 Dynamic Port VLAN Membership Configuration



Two topologies are possible. Figure 13-5 illustrates a topology with one end station attached directly to a Catalyst 4500 series switch operating as a VMPS client. Figure 13-6 illustrates a topology with an end station attached to a Cisco IP Phone, which is attached to a Catalyst 4500 series switch.

Figure 13-5 Dynamic Port VLAN Membership Configuration

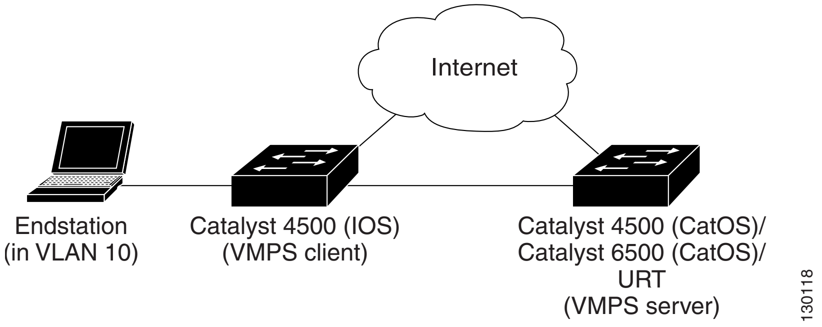
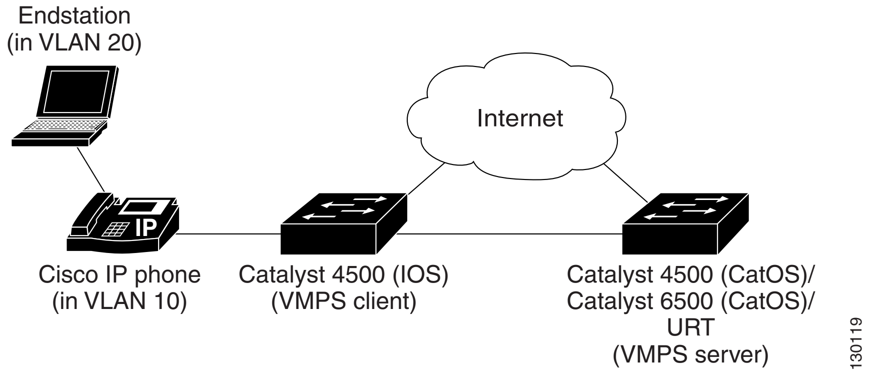


Figure 13-6 Dynamic Port VLAN Membership Configuration



In the following procedure, the Catalyst 4000 and Catalyst 6000 series switches (running CatOS) are the VMPS servers. Use this procedure to configure the Catalyst 4500 series switch clients in the network:

Step 1 Configure the VMPS server addresses on Switch 2, the client switch.

a. Starting from privileged EXEC mode, enter global configuration mode:

```
switch# configuration terminal
```

b. Enter the primary VMPS server IP address:

```
switch(config)# vmps server 172.20.26.150 primary
```

c. Enter the secondary VMPS server IP addresses:

```
switch(config)# vmps server 172.20.26.152
```

d. To verify your entry of the VMPS IP addresses, return to privileged EXEC mode:

```
switch#(config) exit
```

e. Display VMPS information configured for the switch:

```
switch# show vmps
VQP Client Status:
--------------------
VMPS VQP Version:   1
Reconfirm Interval: 60 min
Server Retry Count: 3
VMPS domain server: 172.20.26.152
                    172.20.26.150 (primary, current
```

Step 2 Configure port Fa0/1 on Switch 2 as a dynamic port.

a. Return to global configuration mode:

```
switch# configure terminal
```

b. Enter interface configuration mode:

```
switch(config)# interface fa2/1
```

c. Configure the VLAN membership mode for static-access ports:

```
switch(config-if)# switchport mode access
```

d. Assign the port dynamic VLAN membership:

```
switch(config-if)# switchport access vlan dynamic
```

e. Return to privileged EXEC mode:

```
switch(config-if)# exit
switch#
```

Step 3 Connect End Station 2 on port Fa2/1. When End Station 2 sends a packet, Switch 2 sends a query to the primary VMPS server, Switch 1. Switch 1 responds with the VLAN ID for port Fa2/1. If spanning-tree PortFast mode is enabled on Fa2/1, port Fa2/1 connects immediately and begins forwarding.

Step 4 Set the VMPS reconfirmation period to 60 minutes. The reconfirmation period is the number of minutes the switch waits before reconfirming the VLAN to MAC address assignments.

```
switch# config terminal
switch(config)# vmps reconfirm 60
```

Step 5 Confirm the entry from privileged EXEC mode:

```
switch# show vmps
VQP Client Status:
--------------------
VMPS VQP Version:   1
Reconfirm Interval: 60 min
Server Retry Count: 3
VMPS domain server:

Reconfirmation status
---------------------
VMPS Action:         No Dynamic Port
```

Step 6 Repeat Steps 1 and 2 to configure the VMPS server addresses, and assign dynamic ports on each VMPS client switch.

VMPS Database Configuration File Example

This example shows a sample VMPS database configuration file as it appears on a VMPS server. A VMPS database configuration file is an ASCII text file that is stored on a TFTP server accessible to the switch that functions as the VMPS server.

```
!vmps domain <domain-name>
! The VMPS domain must be defined.
!vmps mode { open | secure }
! The default mode is open.
!vmps fallback <vlan-name>
!vmps no-domain-req { allow | deny }
!
! The default value is allow.
vmps domain WBU
vmps mode open
vmps fallback default
vmps no-domain-req deny
!
!
!MAC Addresses
!
vmps-mac-addrs
!
! address <addr> vlan-name <vlan_name>
!
address 0012.2233.4455 vlan-name hardware
address 0000.6509.a080 vlan-name hardware
address aabb.ccdd.eeff vlan-name Green
address 1223.5678.9abc vlan-name ExecStaff
address fedc.ba98.7654 vlan-name --NONE--
address fedc.ba23.1245 vlan-name Purple
!
!Port Groups
!
!vmps-port-group <group-name>
! device <device-id> { port <port-name> | all-ports }
!
vmps-port-group WiringCloset1
 device 198.92.30.32 port Fa1/3
 device 172.20.26.141 port Fa1/4
vmps-port-group “Executive Row”
 device 198.4.254.222 port es5%Fa0/1
 device 198.4.254.222 port es5%Fa0/2
 device 198.4.254.223 all-ports
!
!VLAN groups
!
!vmps-vlan-group <group-name>
! vlan-name <vlan-name>
!
vmps-vlan-group Engineering
vlan-name hardware
vlan-name software
!
!VLAN port Policies
!
!vmps-port-policies {vlan-name <vlan_name> | vlan-group <group-name> }
! { port-group <group-name> | device <device-id> port <port-name> }
!
```

```
vmps-port-policies vlan-group Engineering
 port-group WiringCloset1
vmps-port-policies vlan-name Green
 device 198.92.30.32 port Fa0/9
vmps-port-policies vlan-name Purple
 device 198.4.254.22 port Fa0/10
 port-group “Executive Row”
```

CHAPTER 14

Configuring IP Unnumbered Interface

Note IP UnControl Plane Policing is *not* supported on Supervisor Engine 6-E.

This chapter discusses the IP Unnumbered Interface feature, which allows you to enable IP processing on an interface without assigning an explicit IP address.

This chapter contains these sections:

- Overview of IP Unnumbered Support, page 14-2
- Configuring IP Unnumbered Interface Support with DHCP Server, page 14-4
- Configuring IP Unnumbered Interface Support with Connected Host Polling, page 14-6
- Displaying IP Unnumbered Interface Settings, page 14-7
- Troubleshooting IP Unnumbered, page 14-8

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Related Documents

Related Topic	Document Title
DHCP and other IP addressing configuration tasks	"IP Addressing and Services" section of the *Cisco IOS IP Addressing Services Configuration Guide,* Release 12.4
DHCP and other IP addressing commands	*Cisco IOS IP Addressing Services Command Reference*, Release 12.4 T
VLAN configuration tasks	"Virtual LANs" chapter of the *Cisco IOS LAN Switching Configuration Guide,* Release 12.4
VLAN configuration commands	*Cisco IOS LAN Switching Command Reference,* Release 12.4 T

Overview of IP Unnumbered Support

Before you configure VLANs and LAN interfaces with IP unnumbered interfaces, you should understand the following concepts:

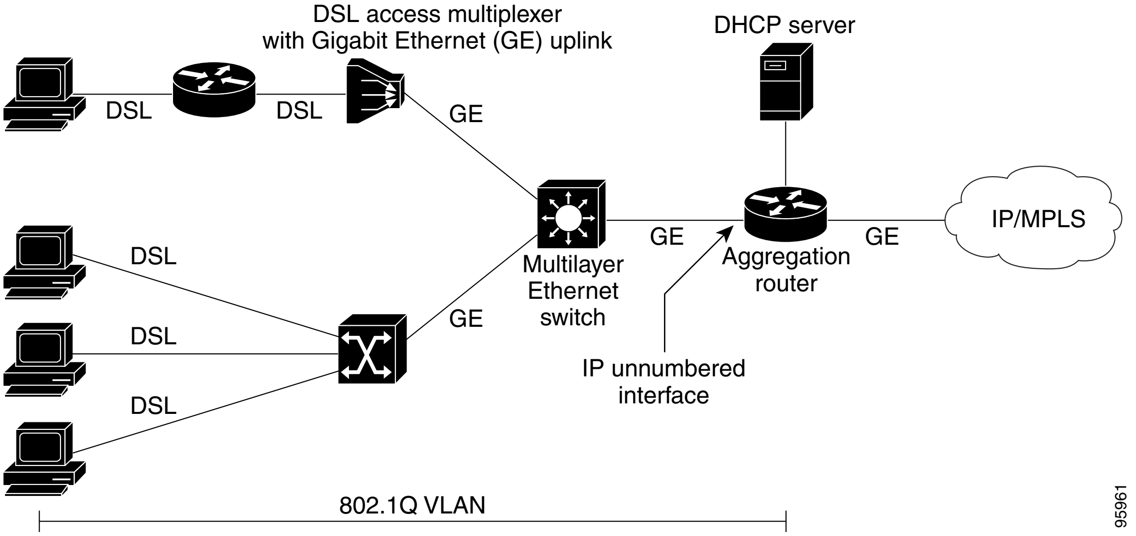
- IP Unnumbered Interface Support with DHCP Server and Relay Agent, page 14-2
- IP Unnumbered with Connected Host Polling, page 14-3

IP Unnumbered Interface Support with DHCP Server and Relay Agent

The IP unnumbered interface configuration allows you to enable IP processing on an interface without assigning it an explicit IP address. The IP unnumbered interface can “borrow” the IP address from another interface that is already configured on the Catalyst 4500 series switch, thereby conserving network and address space. When employed with the DHCP server/relay agent, this feature allows a host address assigned by the DHCP server to be learned dynamically at the DHCP relay agent.

Figure 1 shows a sample network topology implementing the IP Unnumbered Interface feature. In this topology, IP routes are dynamically established by the aggregation switch when the DHCP server assigns IP addresses to the hosts.

Figure 1 Sample Network Topology Using the VLANs over IP Unnumbered Interfaces Feature



DHCP Option 82

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items are also called *options*. Option 82 is organized as a single DHCP option that contains information known by the relay agent.

The IP Unnumbered Interface feature communicates information to the DHCP server using a suboption of the DHCP relay agent information option called *agent remote ID*. The information sent in the agent remote ID includes an IP address identifying the relay agent and information about the interface and the connection over which the DHCP request entered. The DHCP server can use this information to make IP address assignments and security policy decisions.

Figure 2 shows the agent remote ID suboption format that is used with the IP Unnumbered Interfaces feature.

Figure 2 Format of the Agent Remote ID Suboption

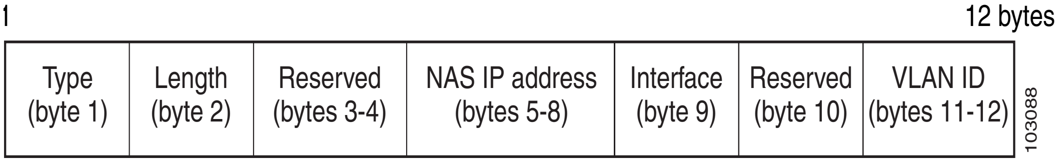


Table 1 describes the agent remote ID suboption fields displayed in Figure 2.

Table 1 Agent Remote ID Suboption Field Descriptions

Field	Description
Type	Format type. The value 2 specifies the format for use with this feature. (1 byte)
Length	Length of the Agent Remote ID suboption, not including the type and length fields. (1 byte)
Reserved	Reserved. (2 bytes)
NAS IP Address	IP address of the interface specified by the **ip unnumbered** command. (4 bytes)
Interface	Physical interface. This field has the following format: slot (4 bits) | module (1 bit) | port (3 bits). For example, if the interface name is interface ethernet 2/1/1, the slot is 2, the module is 1, and the port is 1. (1 byte)
Reserved	Reserved. (1 byte)
VLAN ID	VLAN identifier for the Ethernet interface. (2 bytes)

IP Unnumbered with Connected Host Polling

Note This feature option is applicable to LAN and VLAN interfaces only.

In some cases, the host IP address is assigned statically. The IP Unnumbered Interfaces feature can learn the static host IP address dynamically.

Configuring IP Unnumbered Interface Support with DHCP Server

Note DHCP must be configured and operational.

This section contains the following procedures:

- Configuring IP Unnumbered Interface Support on LAN and VLAN Interfaces, page 14-4
- Configuring IP Unnumbered Interface Support on a Range of Ethernet VLANs, page 14-5

Configuring IP Unnumbered Interface Support on LAN and VLAN Interfaces

To configure IP unnumbered interface support on a single LAN or VLAN interface, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** [**fastethernet** | **gigabitethernet** | **tengigabitethernet** | **vlan** *vlan*} **port-channel** | **loopback**]
4. **ip unnumbered** *type number*

DETAILED STEPS

	Command	Purpose
Step 1	`Switch# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# configure terminal`	Enters global configuration mode.
Step 3	`Switch(config)# interface [fastethernet	gigabitethernet	tengigabitethernet	vlan vlan	port-channel	loopback]`	Enters interface configuration mode and the interface to be configured as a tunnel port.
Step 4	`Switch(config-if)# ip unnumbered type number`	Enables IP processing on an interface without assigning an explicit IP address to the interface. The *type* and *number* arguments specify another interface on which the switch has an assigned IP address. The interface specified cannot be another unnumbered interface.
Step 5	`Switch(config-if)# exit`	Returns to global configuration mode.
Step 6	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show running-config`	Verifies that IP unnumbered support has been configured correctly.

In the following example, Ethernet VLAN 10 is configured as an IP unnumbered interfaces:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface vlan 10
Switch(config-if)# ip unnumbered Lookback 0
```

Configuring IP Unnumbered Interface Support on a Range of Ethernet VLANs

To configure IP unnumbered interface support on a range of Ethernet VLAN interfaces, perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface range** {{**fastethernet** | **gigabitethernet** | **vlan** *vlan*} *slot*/*interface* {**fastethernet** | **gigabitethernet** | **vlan** *vlan*} *slot*/*interface* **macro** *macro-name*}
4. **ip unnumbered** *type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	`Switch# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# configure terminal`	Enters global configuration mode.
Step 3	`Switch(config)# interface range {{fastethernet | gigabitethernet | vlan vlan} slot/interface {fastethernet | gigabitethernet | vlan vlan} slot/interface | macro macro-name}`	Executes commands on multiple interfaces at the same time. A hyphen must be entered with a space on either side to separate the range information.
Step 4	`Switch(config-if)# ip unnumbered type number`	Enables IP processing on an interface without assigning an explicit IP address to the interface. The *type* and *number* arguments specify another interface on which the switch has an assigned IP address. The specified interface cannot be another unnumbered interface.
Step 5	`Switch(config-if)# exit`	Returns to global configuration mode.
Step 6	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show running-config`	Verifies that IP unnumbered support has been configured correctly.

In the following example, Vlan in the range from 1 to 10 are configured as IP unnumbered interfaces, sharing ip address of fastethernet 3/1:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface range vlan 1 - 10
Switch(config-if)# ip unnumbered fastethernet 3/1
Switch(config-if)# exit
Switch(config)# end
```

Configuring IP Unnumbered Interface Support with Connected Host Polling

To configure IP unnumbered interface support with connected host polling, perform this task:

	Command	Purpose
Step 1	`Switch# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Switch# configure terminal`	Enters global configuration mode.
Step 3	`Switch(config)# interface vlan vlan-id`	Enters interface configuration mode and the interface to be configured as a tunnel port.
Step 4	`Switch(config-if)# ip unnumbered type number poll`	Enables IP processing and connected host polling on an interface without assigning an explicit IP address to the interface *type* and *number* specify another interface on which the switch has an assigned IP address. The interface specified cannot be another unnumbered interface. The *type* argument can have the values: *loopback*, *fastethernet*, *gigabitethernet*, *svi*, and *portchannel*.
Step 5	`Switch(config-if)# exit`	Returns to global configuration mode.
Step 6	`Switch(config)# ip arp poll queue <10-10000>`	Configures the global backlog queue of host addresses to be discovered. Default for the queue size is 1000
Step 7	`Switch(config)# ip arp poll rate <10-10000>`	Configures the maximum number of arp requests sent over unnumbered interfaces. Default number of arp requests is 1000 packet per second
Step 8	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 9	`Switch# show running-config`	Verifies that IP unnumbered support has been configured correctly.

The following example shows how to enable IP processing and connected host polling on Fast Ethernet interface 6/2. It also shows how to set the global backlog queue to 2000 and the maximum number of arp requests to 500:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastEthernet 6/2
Switch(config-if)# no switchport
Switch(config-if)# ip unnumbered loopback 0 poll
Warning: dynamic routing protocols will not work on non-point-to-point interfaces with IP
unnumbered configured.
```

```
Switch(config-if)# exit
Switch(config)# ip arp poll queue 2000
Switch(config)# ip arp poll rate 500
Switch(config)# end
```

Displaying IP Unnumbered Interface Settings

Use the **show ip interface [type number] unnumbered [detail]** command to display status of an unnumbered interface with connected host polling for the switch.

To display status of an unnumbered interface, perform one or more of these tasks:

Command	Purpose
`Switch# show ip interface [type number] unnumbered [detail]`	Displays the status of unnumbered interface with connected host polling for the Catalyst 4500 series switch.

The following example shows how to display the status of unnumbered interface with connected host polling:

```
Switch# show ip interface loopback 0 unnumbered detail
Number of unnumbered interfaces with polling: 1
Number of IP addresses processed for polling: 2
10.1.1.7
10.1.1.8
Number of IP addresses in queue for polling: 2(high water mark: 3)
10.1.1.17
10.1.1.18
```

To display key statistic for the backlog of unnumbered interface with connected host polling for the switch, use the **show ip arp poll** command.

Command	Purpose
`Switch# show ip arp poll [detail]`	display key statistic for the backlog of unnumbered interface with connected host polling for the switch

The following example shows how to display key statistic for the backlog of unnumbered interface with connected host polling:

```
Switch# show ip arp poll
Number of IP addresses processed for polling: 439
Number of IP addresses in queue for polling: 3 (high water mark: 0, max: 1000)
Number of requests dropped:
  Queue was full: 0
  Request was throttled by incomplete ARP: 0
  Duplicate request was found in queue: 0
```

To clear the key statistic for the backlog of unnumbered interface, use the **clear ip arp poll statistic** command, as follows:

```
Switch# clear ip arp poll statistic
Switch# show ip arp poll
Number of IP addresses processed for polling: 0
Number of IP addresses in queue for polling: 0 (high water mark: 0, max: 1000)
Number of requests dropped:
  Queue was full: 0
  Request was throttled by incomplete ARP: 0
  Duplicate request was found in queue: 0
```

Troubleshooting IP Unnumbered

To understand how to debug connect host polling, see the IOS documentation of the **debug arp** command on cisco.com.

When an IP unnumbered interface shares the IP address of a loopback interface whose prefix is advertised in an OSPF network, you must modify the loopback interface as a point to point interface. Otherwise, only the loopback interface host route will be advertised to an OSPF neighbor.

```
Switch(config)# int loopback 0
Switch(config-if)# ip address
Switch(config-if)# ip address 10.1.0.1 255.255.0.0
Switch(config-if)# ip ospf network point-to-point
Switch(config-if)# end
```

CHAPTER 15

Configuring Layer 2 Ethernet Interfaces

This chapter describes how to use the command-line interface (CLI) to configure Fast Ethernet and Gigabit Ethernet interfaces for Layer 2 switching on Catalyst 4500 series switches. It also provides guidelines, procedures, and configuration examples. The configuration tasks in this chapter apply to Fast Ethernet and Gigabit Ethernet interfaces on any module, including the uplink ports on the supervisor engine.

This chapter includes the following major sections:

- Overview of Layer 2 Ethernet Switching, page 15-1
- Default Layer 2 Ethernet Interface Configuration, page 15-4
- Layer 2 Interface Configuration Guidelines and Restrictions, page 15-5
- Configuring Ethernet Interfaces for Layer 2 Switching, page 15-5

Note To configure Layer 3 interfaces, see Chapter 26, "Configuring Layer 3 Interfaces."

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of Layer 2 Ethernet Switching

The following sections describe how Layer 2 Ethernet switching works on Catalyst 4500 series switches:

- Understanding Layer 2 Ethernet Switching, page 15-1
- Understanding VLAN Trunks, page 15-3
- Layer 2 Interface Modes, page 15-4

Understanding Layer 2 Ethernet Switching

Catalyst 4500 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 successive packets.

Note With release 12.1(13)EW, the Catalyst 4500 series switches can handle packets of 1600 bytes, rather than treat them as "oversized" and discard them. This size is larger than the usual IEEE Ethernet Maximum Transmission Unit (MTU) (1518 bytes) and 802.1q MTU (1522 bytes). The ability to handle larger packets is required to support two nested 802.1q headers and Multiprotocol Label Switching (MPLS) on a network.

The Catalyst 4500 series solves congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own 10-, 100-, or 1000-Mbps segment. Because each Ethernet interface on the switch represents a separate Ethernet segment, servers in a properly configured switched environment achieve full access to the bandwidth.

Because collisions are a major bottleneck 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 devices can transmit and receive at the same time. When packets can flow in both directions simultaneously, effective Ethernet bandwidth doubles to 20 Mbps for 10-Mbps interfaces and to 200 Mbps for Fast Ethernet interfaces. Gigabit Ethernet interfaces on the Catalyst 4500 series switch are full-duplex mode only, providing 2-Gbps effective bandwidth.

Switching Frames Between Segments

Each Ethernet interface on a Catalyst 4500 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 devices 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 treats each interface as an individual segment. When stations on different interfaces need to communicate, the switch forwards frames from one interface to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between interfaces efficiently, the switch maintains an address table. When a frame enters the switch, it associates the MAC address of the sending station with the interface on which it was received.

Building the MAC Address Table

The Catalyst 4500 series builds the MAC 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 MAC address table, it floods the frame to all interfaces of the same VLAN except the interface that received the frame. When the destination device replies, the switch adds its relevant source address and interface ID to the address table. The switch then forwards subsequent frames to a single interface without flooding to all interfaces.

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

A trunk is a point-to-point link between one or more Ethernet switch interfaces and another networking device such as a router or a switch. 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 interfaces:

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

Note Supervisor Engine 6-E does *not* support ISL trunking. so the **switchport trunk encapsulute** command is not supported.

Note The blocking Gigabit ports on the WS-X4418-GB and WS-X4412-2GB-T modules do not support ISL. Ports 3 to 18 are blocking Gigabit ports on the WS-X4418-GB module. Ports 1to 12 are blocking Gigabit ports on the WS-X4412-2GB-T module.

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

You can configure a trunk on a single Ethernet interface or on an EtherChannel bundle. For more information about EtherChannel, see Chapter 19, “Configuring EtherChannel.”

Ethernet trunk interfaces support different trunking modes (see Table 15-2). You can specify whether the trunk uses ISL encapsulation, 802.1Q encapsulation, or if the encapsulation type is autonegotiated.

To autonegotiate trunking, make sure your interfaces are in the same VTP domain. Use the **trunk** or **nonegotiate** keywords to force interfaces in different domains to trunk. For more information on VTP domains, see

Trunk negotiation is managed by the Dynamic Trunking Protocol (DTP). DTP supports autonegotiation of both ISL and 802.1Q trunks.

Encapsulation Types

Table 15-1 lists the Ethernet trunk encapsulation types.

Table 15-1 Ethernet Trunk Encapsulation Types

Encapsulation Type	Encapsulation Command	Purpose
ISL	**switchport trunk encapsulation isl**	Specifies ISL encapsulation on the trunk link.
802.1Q	**switchport trunk encapsulation dot1q**	Specifies 802.1Q encapsulation on the trunk link.
Negotiate	**switchport trunk encapsulation negotiate**	Specifies that the interface negotiate with the neighboring interface to become an ISL (preferred) or 802.1Q trunk, depending on the configuration and capabilities of the neighboring interface.

The trunking mode, the trunk encapsulation type, and the hardware capabilities of the two connected interfaces determine whether a link becomes an ISL or 802.1Q trunk.

Layer 2 Interface Modes

Table 15-2 lists the Layer 2 interface modes and describes how they function on Ethernet interfaces.

Table 15-2 Layer 2 Interface Modes

Mode	Purpose
switchport mode access	Puts the interface into permanent nontrunking mode and negotiates to convert the link into a nontrunking link. The interface becomes a nontrunk interface even if the neighboring interface does not change.
switchport mode dynamic desirable	Makes the interface actively attempt to convert the link to a trunking link. The interface becomes a trunk interface if the neighboring interface is set to **trunk**, **desirable**, or **auto** mode.
switchport mode dynamic auto	Makes the interface convert the link to a trunking link if the neighboring interface is set to **trunk** or **desirable** mode. This is the default mode for all Ethernet interfaces.
switchport mode trunk	Puts the interface into permanent trunking mode and negotiates to convert the link into a trunking link. The interface becomes a trunk interface even if the neighboring interface does not change.
switchport nonegotiate	Puts the interface into permanent trunking mode but prevents the interface from generating DTP frames. You must configure the neighboring interface manually as a trunk interface to establish a trunking link.

Note DTP is a point-to-point protocol. However, some internetworking devices might forward DTP frames improperly. To avoid this problem, ensure that interfaces 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 interface to become a trunk without generating DTP frames.

Default Layer 2 Ethernet Interface Configuration

Table 15-3 shows the Layer 2 Ethernet interface default configuration.

Table 15-3 Layer 2 Ethernet Interface Default Configuration

Feature	Default Value
Interface mode	switchport mode dynamic auto
Trunk encapsulation	switchport trunk encapsulation negotiate
Allowed VLAN range	VLANs 1–1005
VLAN range eligible for pruning	VLANs 2–1001
Default VLAN (for access ports)	VLAN 1

Table 15-3 Layer 2 Ethernet Interface Default Configuration (continued)

Feature	Default Value
Native VLAN (for 802.1Q only trunks)	VLAN 1
STP[1]	Enabled for all VLANs
STP port priority	128
STP port cost	• 100 for 10-Mbps Ethernet LAN ports • 19 for 10/100-Mbps Fast Ethernet ports • 19 for 100-Mbps Fast Ethernet ports • 4 for 1000-Mbps Gigabit Ethernet ports • 2 for 10,000-Mbps 10-Gigabit Ethernet LAN ports

[1] STP = Spanning Tree Protocol

Layer 2 Interface Configuration Guidelines and Restrictions

Keep the following guidelines and restrictions in mind when you configure Layer 2 interfaces:

- In a network of Cisco switches connected through 802.1Q trunks, the switches maintain one instance of spanning tree for each VLAN allowed on the trunks. Non-Cisco 802.1Q switches maintain only one instance of spanning tree for all VLANs allowed on the trunks.

  When you connect a Cisco switch to a non-Cisco device through an 802.1Q trunk, the Cisco switch combines the spanning tree instance of the native VLAN of the trunk with the spanning tree instance of the non-Cisco 802.1Q switch. However, spanning tree information for each VLAN is maintained by Cisco switches separated by a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud separating the Cisco switches is treated as a single trunk link between the switches.
- Make sure the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the VLAN on one end of the trunk is different from the VLAN on the other end, spanning tree loops might result.
- Disabling spanning tree on any VLAN of an 802.1Q trunk can cause spanning tree loops.

Configuring Ethernet Interfaces for Layer 2 Switching

The following sections describe how to configure Layer 2 switching on a Catalyst 4500 series switch:

- Configuring an Ethernet Interface as a Layer 2 Trunk, page 15-6
- Configuring an Interface as a Layer 2 Access Port, page 15-8
- Clearing Layer 2 Configuration, page 15-9

Configuring an Ethernet Interface as a Layer 2 Trunk

Note The default for Layer 2 interfaces is **switchport mode dynamic auto**. If the neighboring interface supports trunking and is configured to trunk mode or dynamic desirable mode, the link becomes a Layer 2 trunk. By default, trunks negotiate encapsulation. If the neighboring interface supports ISL and 802.1Q encapsulation and both interfaces are set to negotiate the encapsulation type, the trunk uses ISL encapsulation.

Note Supervisor Engine 6-E does *not* support ISL trunking.

To configure an interface as a Layer 2 trunk, perform this task:

	Command	Purpose
Step 1	Switch(config)# **interface** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*	Specifies the interface to configure.
Step 2	Switch(config-if)# **shutdown**	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	Switch(config-if)# **switchport trunk encapsulation** {**isl** | **dot1q** | **negotiate**}	(Optional) Specifies the encapsulation. **Note** You must enter this command with either the **isl** or **dot1q** keyword to support the **switchport mode trunk** command, which is not supported by the default mode (**negotiate**). **Note** Supervisor Engine 6-E does *not* support ISL trunking.
Step 4	Switch(config-if)# **switchport mode** {**dynamic** {**auto** | **desirable**} | **trunk**}	Configures the interface as a Layer 2 trunk. (Required only if the interface is a Layer 2 access port or to specify the trunking mode.)
Step 5	Switch(config-if)# **switchport access vlan** *vlan_num*	(Optional) Specifies the access VLAN, which is used if the interface stops trunking. The access VLAN is not used as the native VLAN. **Note** The *vlan_num* parameter is either a single VLAN number from 1 to 1005 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.
Step 6	Switch(config-if)# **switchport trunk native vlan** *vlan_num*	For 802.1Q trunks, specifies the native VLAN. **Note** If you do not set the native VLAN, the default is used (VLAN 1).
Step 7	Switch(config-if)# **switchport trunk allowed vlan** {**add** | **except** | **all** | **remove**} *vlan_num*[,*vlan_num*[,*vlan_num*[,....]]	(Optional) Configures the list of VLANs allowed on the trunk. All VLANs are allowed by default. You cannot remove any of the default VLANs from a trunk.

	Command	Purpose
Step 8	`Switch(config-if)#` **`switchport trunk pruning vlan`** `{`**`add`** `|` **`except`** `|` **`none`** `|` **`remove`**`}` *`vlan_num`*`[,`*`vlan_num`*`[,`*`vlan_num`*`[,....]]`	(Optional) Configures the list of VLANs allowed to be pruned from the trunk (see the "VLAN Trunking Protocol" section on page 13-7). The default list of VLANs allowed to be pruned contains all VLANs, except for VLAN 1.
Step 9	`Switch(config-if)#` **`no shutdown`**	Activates the interface. (Required only if you shut down the interface.)
Step 10	`Switch(config-if)#` **`end`**	Exits interface configuration mode.
Step 11	`Switch#` **`show running-config interface`** `{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`*	Displays the running configuration of the interface.
Step 12	`Switch#` **`show interfaces`** `[`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`]` *`slot/port`* **`switchport`**	Displays the switch port configuration of the interface.
Step 13	`Switch#` **`show interfaces`** `[{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`*`]` **`trunk`**	Displays the trunk configuration of the interface.

This example shows how to configure the Fast Ethernet interface 5/8 as an 802.1Q trunk. This example assumes that the neighbor interface is configured to support 802.1Q trunking and that the native VLAN defaults to VLAN 1:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/8
Switch(config-if)# shutdown
Switch(config-if)# switchport mode dynamic desirable
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# no shutdown
Switch(config-if)# end
Switch# exit
```

This example shows how to verify the running configuration:

```
Switch# show running-config interface fastethernet 5/8
Building configuration...
Current configuration:
!
interface FastEthernet5/8
 switchport mode dynamic desirable
 switchport trunk encapsulation dot1q
end
```

This example shows how to verify the switch port configuration:

```
Switch# 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: 2-1001
```

This example shows how to verify the trunk configuration:

```
Switch# show interfaces fastethernet 5/8 trunk

Port      Mode         Encapsulation  Status        Native vlan
Fa5/8     desirable    n-802.1q       trunking      1

Port      Vlans allowed on trunk
Fa5/8 1-1005

Port      Vlans allowed and active in management domain
Fa5/8 1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-8
02,850,917,999,1002-1005

Port      Vlans in spanning tree forwarding state and not pruned
Fa5/8 1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-8
02,850,917,999,1002-1005

Switch#
```

Configuring an Interface as a Layer 2 Access Port

Note If you assign an interface to a VLAN that does not exist, the interface is not operational until you create the VLAN in the VLAN database (see the "Configuring VLANs in Global Configuration Mode" section on page 13-5).

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

	Command	Purpose
Step 1	`Switch(config)#` **`interface`** `{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`*	Specifies the interface to configure.
Step 2	`Switch(config-if)#` **`shutdown`**	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	`Switch(config-if)#` **`switchport`**	Configures the interface for Layer 2 switching: • You must enter the **switchport** command once without any keywords to configure the interface as a Layer 2 port before you can enter additional **switchport** commands with keywords. • Required only if you previously entered the **no switchport** command for the interface.
Step 4	`Switch(config-if)#` **`switchport mode access`**	Configures the interface as a Layer 2 access port.
Step 5	`Switch(config-if)#` **`switchport access vlan`** *`vlan_num`*	Place the interface in a VLAN.
Step 6	`Switch(config-if)#` **`no shutdown`**	Activates the interface. (Required only if you had shut down the interface.)
Step 7	`Switch(config-if)#` **`end`**	Exits interface configuration mode.

	Command	Purpose
Step 8	Switch# **show running-config interface** {**fastethernet** | **gigabitethernet**} *slot/port*	Displays the running configuration of the interface.
Step 9	Switch# **show interfaces** [{**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*] **switchport**	Displays the switch port configuration of the interface.

This example shows how to configure the Fast Ethernet interface 5/6 as an access port in VLAN 200:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/6
Switch(config-if)# shutdown
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 200
Switch(config-if)# no shutdown
Switch(config-if)# end
Switch# exit
```

This example shows how to verify the running configuration:

```
Switch# show running-config interface fastethernet 5/6
Building configuration...
!
Current configuration :33 bytes
interface FastEthernet 5/6
 switchport access vlan 200
 switchport mode access
end
```

This example shows how to verify the switch port configuration:

```
Switch# show running-config interface fastethernet 5/6 switchport
Name:Fa5/6
Switchport:Enabled
Administrative Mode:dynamic auto
Operational Mode:static access
Administrative Trunking Encapsulation:negotiate
Operational Trunking Encapsulation:native
Negotiation of Trunking:On
Access Mode VLAN:1 (default)
Trunking Native Mode VLAN:1 (default)
Administrative private-vlan host-association:none
Administrative private-vlan mapping:none
Operational private-vlan:none
Trunking VLANs Enabled:ALL
Pruning VLANs Enabled:2-1001
Switch#
```

Clearing Layer 2 Configuration

To clear the Layer 2 configuration on an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# **default interface** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*	Specifies the interface to clear.
Step 2	Switch(config-if)# **end**	Exits interface configuration mode.

	Command	Purpose
Step 3	`Switch# show running-config interface {fastethernet | gigabitethernet | tengigabitethernet} slot/port`	Displays the running configuration of the interface.
Step 4	`Switch# show interfaces [{fastethernet | gigabitethernet | tengigabitethernet} slot/port] switchport`	Displays the switch port configuration of the interface.

This example shows how to clear the Layer 2 configuration on the Fast Ethernet interface 5/6:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# default interface fastethernet 5/6
Switch(config)# end
Switch# exit
```

This example shows how to verify that the Layer 2 configuration was cleared:

```
Switch# show running-config interface fastethernet 5/6
Building configuration...
Current configuration:
!
interface FastEthernet5/6
end
```

This example shows how to verify the switch port configuration:

```
Switch# show interfaces fastethernet 5/6 switchport
Name: Fa5/6
Switchport: Enabled
Switch#
```

CHAPTER 16

Configuring SmartPort Macros

This chapter describes how to configure and apply SmartPort macros on your switch.

This chapter consists of these sections:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Understanding SmartPort Macros

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

Each SmartPort macro is a set of CLI commands that you define. SmartPort macro sets do not contain new CLI commands; Each SmartPort macro is a group of existing CLI commands.

When you apply a SmartPort macro on an interface, the CLI commands contained within the macro are configured on the interface. When the macro is applied to an interface, the existing interface configurations are not lost. The new commands are added to interface and are saved in the running configuration file.

There are Cisco-default Smartports macros embedded in the switch software (see Table 16-1). You can display these macros and the commands they contain by using the **show parser macro** user EXEC command.

Table 16-1 Cisco-Default Smartports Macros

Macro Name[1]	Description
cisco-global	Use this global configuration macro to enable rapid PVST+, loop guard, and dynamic port error recovery for link state failures.
cisco-desktop	Use this interface configuration macro for increased network security and reliability when connecting a desktop device, such as a PC, to a switch port.

Table 16-1 Cisco-Default Smartports Macros (continued)

Macro Name[1]	Description
cisco-phone	Use this interface configuration macro when connecting a desktop device such as a PC with a Cisco IP Phone to a switch port. This macro is an extension of the **cisco-desktop** macro and provides the same security and resiliency features, but with the addition of dedicated voice VLANs to ensure proper treatment of delay-sensitive voice traffic.
cisco-switch	Use this interface configuration macro when connecting an access switch and a distribution switch or between access switches connected using GigaStack modules or GBICs.
cisco-router	Use this interface configuration macro when connecting the switch and a WAN router.

[1] Cisco-default Smartports macros vary depending on the software version running on your switch.

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

http://www.cisco.com/go/smartports

Configuring Smart-Port Macros

You can create a new SmartPort macro or use an existing macro as a template to create a new macro that is specific to your application. After you create the macro, you can apply it to an interface or a range of interfaces.

This section includes information about these topics:

- Passing Parameters Through the Macro, page 16-2
- Default SmartPort Macro Configuration, page 16-3
- SmartPort Macro Configuration Guidelines, page 16-6
- Creating Smartports Macros, page 16-7
- Applying Smartports Macros, page 16-8

Passing Parameters Through the Macro

Some commands might not be sufficiently generic for all the interfaces; for example, VLAN ID for Layer 2 interfaces and the IP address for Layer 3 interface. Retaining such commands in macro definitions requires that you change the value of such parameters (like VLAN ID or IP address) before applying the macro to different interfaces. Alternatively, it requires that you create different macros for each possible value of its parameters.

The macro infrastructure can be enhanced to support accepting parameters while applying a macro. The parameters are passed as *keyword-value* pairs.

The CLI limits the number of keyword-value pairs to a maximum of three, where the first parameter must be the keyword, the second is its corresponding value, and the third parameter would be the keyword for the second keyword-value pair. Here is an example of how to pass parameters to a command-macro:

```
Switch(config)# macro name parameter-test
Enter macro commands one per line. End with the character '@'.
switchport mode access
switchport access vlan $VLANID
switchport port-security
switchport port-security maximum $MAXHOST
```

If the above macro is applied to some interface without parameters, the invalid commands fail. Instead, you should apply the macro with appropriate keyword-value pair parameters, as follows:

```
Switch(config-if)# macro apply parameter-test $VLANID 1 $MAXHOST 5
```

The above command applies the macro after replacing $VLANID with 1 and $MAXHOST with 5. Be aware that you can specify any string in the macro as a keyword.

Macro Parameter Help

It is often difficult to remember the macro keywords while applying a macro to an interface or switch. Macros can contain the definitions for mandatory keywords. If you apply a macro without those keyword values, the commands are considered invalid and they will fail.

You can enhance the macro infrastructure to provide help on keywords defined in macros. While creating a macro, you can specify a help string (as a comment) to list the mandatory keywords for that macro.

The following example illustrates how to specify the help string for the keywords:

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

Help string can be anywhere in the macro. The following example illustrates an alternate way to specify the help string:

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

Default SmartPort Macro Configuration

This section illustrates the default configurations for the four supported macros. These macros can only be viewed and applied; they cannot be modified by the user.

- cisco-global, page 16-4
- cisco-desktop, page 16-4
- cisco-phone, page 16-4
- cisco-router, page 16-5
- cisco-switch, page 16-5

cisco-global

This is the example for the cisco-global macro:

```
# Enable dynamic port error recovery for link state failures.
errdisable recovery cause link-flap
errdisable recovery interval 60

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

# Enable aggressive mode UDLD on all fiber uplinks
udld aggressive

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

cisco-desktop

This is the example for the cisco-desktop macro:

```
# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
switchport mode access
# Enable port security limiting port to a single
# MAC address -- that of desktop
switchport port-security
# Ensure port-security age is greater than one minute
# and use inactivity timer
# “Port-security maximum 1” is the default and will not
# Show up in the config
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable
```

cisco-phone

This is the example for the cisco-phone macro:

```
# VoIP enabled interface - Enable data VLAN
# and voice VLAN (VVID)
# Recommended value for access vlan (AVID) should not be 1\
switchport access vlan $AVID
switchport mode access
# Update the Voice VLAN (VVID) value which should be
# different from data VLAN
# Recommended value for voice vlan (VVID) should not be 1
switchport voice vlan $VVID
# Enable port security limiting port to a 2 MAC
# addressess -- One for desktop and two for phone
switchport port-security
switchport port-security maximum 2
```

```
# Ensure port-security age is greater than one minute
# and use inactivity timer
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
# Enable auto-qos to extend trust to attached Cisco phone
auto qos voip cisco-phone
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable@
```

cisco-router

This is the example for the cisco-router macro:

```
# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE
# Hardcode trunk and disable negotiation to
# speed up convergence
# Hardcode speed and duplex to router
switchport mode trunk
switchport nonegotiate
speed 100
duplex full
# Configure qos to trust this interface
auto qos voip trust
qos trust dscp
# Ensure fast access to the network when enabling the interface.
# Ensure that switch devices cannot become active on the interface.
spanning-tree portfast
spanning-tree bpduguard enable
```

cisco-switch

This is the example for the cisco-switch macro:

```
# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE
# Hardcode trunk and disable negotiation to
# speed up convergence
switchport mode trunk
switchport nonegotiate
# Configure qos to trust this interface
auto qos voip trust
# 802.1w defines the link as pt-pt for rapid convergence
spanning-tree link-type point-to-point
```

SmartPort Macro Configuration Guidelines

Follow these guidelines when configuring macros on your switch:

- If a command fails when you apply a macro, either due to a syntax error or to a configuration error, the macro continues to apply the remaining commands to the interface.
- **cisco-global** needs to be applied at the global configuration mode. Cisco recommends that you apply this macro before any other interface level macro.
- Specific keywords are required when you apply the system-defined macros (**cisco-desktop**, **cisco-phone**, **cisco-switch**, and **cisco-router**) on an interface.
- When using the **cisco-phone** macro to apply port security, the port security maximum is 2 (**switchport port-security maximum 2**).
- At most, 3 keyword-value pairs are allowed per system-defined macro.
- When creating a macro, do not use the **exit** or **end** commands or change the command mode by using **interface** *interface-id*. This could cause commands that follow **exit**, **end**, or **interface** *interface-id* to execute in a different command mode.
- When creating a macro, ensure that all CLI commands are in the same configuration mode.
- When creating a macro that requires the assignment of unique values, use the **parameter** *value* keywords to designate values specific to the interface. Keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. Any full match of a keyword, even if it is part of a larger string, is considered a match and is replaced by the corresponding value.
- Macro names are case sensitive. For example, the commands **macro name Sample-Macro** and **macro name sample-macro** result in two separate macros.
- Some macros might contain keywords that require a parameter value. You can use the **macro global apply** *macro-name* **?** global configuration command or the **macro apply** *macro-name* **?** interface configuration command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.
- When a macro is applied globally to a switch or to a switch interface, all existing configuration on the interface is retained. This is helpful when applying an incremental configuration.
- If you modify a macro definition by adding or deleting commands, the changes are not reflected on the interface where the original macro was applied. You need to reapply the updated macro on the interface to apply the new or changed commands.
- You can use the **macro global trace** *macro-name* global configuration command or the **macro trace** *macro-name* interface configuration command to apply and debug a macro to find any syntax or configuration errors. If a command fails because of a syntax error or a configuration error, the macro continues to apply the remaining commands.
- Some CLI commands are specific to certain interface types. If a macro is applied to an interface that does not accept the configuration, the macro will fail the syntax check or the configuration check, and the switch will return an error message.
- Applying a macro to an interface range is the same as applying a macro to a single interface. When you use an interface range, the macro is applied sequentially to each interface within the range. If a macro command fails on one interface, it is still applied to the remaining interfaces.

- When you apply a macro to a switch or a switch interface, the macro name is automatically added to the macro description of the switch or interface. You can display the applied commands and macro names by using the **show parser macro description** user EXEC command.
- The user-configurable macro has a buffer that can take commands and comments up to 3000 characters. Each newline would take 2 characters and empty lines are counted as is.

There are Cisco-default Smartports macros embedded in the switch software (see Table 16-1). You can display these macros and the commands they contain by using the **show parser macro** user EXEC command.

Follow these guidelines when you apply a Cisco-default Smartports macro on an interface:

- Display all macros on the switch by using the **show parser macro** user EXEC command. Display the contents of a specific macro by using the **show parser macro** *macro-name* user EXEC command.
- Keywords that begin with **$** mean that a unique parameter value is required. Append the Cisco-default macro with the required values by using the **parameter** *value* keywords.

  The Cisco-default macros use the **$** character to help identify required keywords. There is no restriction on using the **$** character to define keywords when you create a macro.

Creating Smartports Macros

To create a Smartports macro, follow these steps:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**macro name** *macro-name*	Creates a macro definition, and enter a macro name. A macro definition can contain up to 3000 characters. Enter the macro commands with one command per line. Use the @ character to end the macro. Use the # character at the beginning of a line to enter comment text within the macro. Macro names are case sensitive. For example, the commands **macro name Sample-Macro** and **macro name sample-macro** result in two separate macros. We recommend that you do not use the **exit** or **end** commands or change the command mode by using **interface** *interface-id* in a macro. This could cause any commands following **exit**, **end**, or **interface** *interface-id* to execute in a different command mode. For best results, all commands in a macro should be in the same configuration mode. **Note** The **no** form of the **macro name** global configuration command only deletes the macro definition. It does not affect the configuration of those interfaces on which the macro is already applied.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show parser macro name** *macro-name*	Verifies that the macro was created.

Applying Smartports Macros

To apply a Smartports macro, follow these steps:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**macro global** {**apply** | **trace**} *macro-name* [**parameter** {*value*}] [**parameter** {*value*}] [**parameter** {*value*}]	Applies each individual command defined in the macro to the switch by entering **macro global apply** *macro-name*. Specify **macro global trace** *macro-name* to apply and debug a macro to find any syntax or configuration errors. (Optional) Specify unique parameter values that are specific to the switch. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. Some macros might contain keywords that require a parameter value. You can use the **macro global apply** *macro-name* **?** command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.
Step 3	**macro global description** *text*	(Optional) Enters a description about the macro that is applied to the switch.
Step 4	**interface** *interface-id*	(Optional) Enters interface configuration mode, and specify the interface on which to apply the macro.
Step 5	**default interface** *interface-id*	(Optional) Clears all configuration from the specified interface.

Step 6	**macro** {**apply** | **trace**} *macro-name* [**parameter** {*value*}] [**parameter** {*value*}] [**parameter** {*value*}]	Applies each individual command defined in the macro to the interface by entering **macro apply** *macro-name*. Specify **macro trace** *macro-name* to apply and debug a macro to find any syntax or configuration errors.
		(Optional) Specify unique parameter values that are specific to the interface. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value.
		Some macros might contain keywords that require a parameter value. You can use the **macro apply** *macro-name* **?** command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.
		For example, here is how you apply this command:
		`Switch(config-if)# macro apply cisco-phone ?`
		`  WORD  Keyword to replace with a value e.g. $AVID, $VVID`
		`  <cr>`

Step 7	**macro description** *text*	(Optional) Enters a description about the macro that is applied to the interface.
Step 8	**end**	Returns to privileged EXEC mode.

	Command	Purpose
Step 9	**show parser macro description** [**interface** *interface-id*]	Verifies that the macro is applied to the interface.
Step 10	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

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

The **no** form of the **macro name** global configuration command deletes only the macro definition. It does not affect the configuration of those interfaces on which the macro is already applied. You can delete a macro-applied configuration on an interface by entering the **default interface** *interface-id* interface configuration command. Alternatively, you can create an *anti-macro* for an existing macro that contains the **no** form of all the corresponding commands in the original macro. Then, apply the anti-macro to the interface.

The following sections illustrate how to apply and display the attachments on each of the supported macros:

- cisco-global, page 16-9
- cisco-desktop, page 16-10
- cisco-phone, page 16-10
- cisco-switch, page 16-11
- cisco-router, page 16-12

cisco-global

This example shows how to use the system-defined macro **cisco-global**:

```
Switch(config)# macro global apply cisco-global
Changing VTP domain name from gsg-switch to [smartports]
Setting device to VTP TRANSPARENT mode.
Switch(config)# end
Switch# show parser macro name cisco-global
Macro name : cisco-global
Macro type : default global
# Enable dynamic port error recovery for link state failures.
errdisable recovery cause link-flap
errdisable recovery interval 60

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

# Enable aggressive mode UDLD on all fiber uplinks udld aggressive

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

cisco-desktop

This example shows how to use the system-defined macro **cisco-desktop** to assign a value of 35 to the access VLAN of the Fast Ethernet interface 2/9.

Note This macro requires the **$AVID** keyword, which is the access VLAN of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-desktop $AVID 35
Switch(config-if)# end
Switch# show parser macro name cisco-desktop
Macro name : cisco-desktop
Macro type : customizable

# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID [access_vlan_id]
switchport mode access
# Enable port security limiting port to a single
# MAC address -- that of desktop
switchport port-security
# Ensure port-security age is greater than one minute
# and use inactivity timer
# “Port-security maximum 1” is the default and will not
# Show up in the config
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable
Switch# show parser macro description
Interface    Macro Description
--------------------------------------------------------------
Fa2/9        cisco-desktop
--------------------------------------------------------------
```

cisco-phone

This example shows how to use the system-defined macro **cisco-phone** to assign a value of 35 to the access VLAN and 56 to the voice VLAN on the Fast Ethernet interface 2/9.

Note This macro requires the **$AVID** and **$VVID** keywords, which are the access and voice VLANs of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-phone
Switch(config-if)# macro description cisco-phone $AVID 35 $VVID 56
Switch(config-if)# end
Switch# show parser macro name cisco-phone
Macro name : cisco-phone
Macro type : customizable

# VoIP enabled interface - Enable data VLAN
# and voice VLAN (VVID)
# Recommended value for access vlan (AVID) should not be 1\
switchport access vlan $AVID [access_vlan_id]
switchport mode access
# Update the Voice VLAN (VVID) value which should be
# different from data VLAN
# Recommended value for voice vlan (VVID) should not be 1
switchport voice vlan $VVID [voice_vlan_id]
# Enable port security limiting port to a 2 MAC
# addressess -- One for desktop and one for phone
switchport port-security
switchport port-security maximum 2
# Ensure port-security age is greater than one minute
# and use inactivity timer
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
# Enable auto-qos to extend trust to attached Cisco phone
auto qos voip cisco-phone
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable@

Switch# show parser macro description
Interface    Macro Description
--------------------------------------------------------------
Fa2/9        cisco-phone
--------------------------------------------------------------
```

cisco-switch

This example shows how to use the system-defined macro **cisco-switch** to assign a value of 38 to the native VLAN on the Fast Ethernet interface 2/9.

Note This macro requires the **$NVID** keyword, which is the native VLANs of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-switch
Switch(config-if)# macro description cisco-switch $NVID 38
Switch(config-if)# end
Switch# show parser macro name cisco-switch
Macro name : cisco-switch
Macro type : customizable

# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID [native_vlan_id]
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE [vlan_range]
# Hardcode trunk and disable negotiation to
# speed up convergence
switchport mode trunk
switchport nonegotiate
# Configure qos to trust this interface
auto qos voip trust
# 802.1w defines the link as pt-pt for rapid convergence
spanning-tree link-type point-to-point
```

```
Switch# show parser macro description
Interface    Macro Description
--------------------------------------------------------------
Fa2/9        cisco-switch
--------------------------------------------------------------
```

cisco-router

This example shows how to use the system-defined macro **cisco-router** to assign a value of 451 to the native VLAN on the Fast Ethernet interface 2/9.

Note This macro requires the **$NVID** keyword, which is the native VLANs of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-router
Switch(config-if)# macro description cisco-router $NVID 45I
Switch(config-if)# end
Switch# show parser macro name cisco-router
Macro name : cisco-router
Macro type : customizable

# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID [native_vlan_id]
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE [vlan_range]
# Hardcode trunk and disable negotiation to
# speed up convergence
# Hardcode speed and duplex to router
switchport mode trunk
switchport nonegotiate
speed 100
duplex full
# Configure qos to trust this interface
auto qos voip trust
qos trust dscp
# Ensure fast access to the network when enabling the interface.
# Ensure that switch devices cannot become active on the interface.
spanning-tree portfast
spanning-tree bpduguard enable

Switch# show parser macro description
Interface    Macro Description
--------------------------------------------------------------
Fa2/9        cisco-router
--------------------------------------------------------------
```

Displaying SmartPort Macros

To display the SmartPort macros, use one or more of the privileged EXEC commands in Table 16-2.

Table 16-2 Commands for Displaying SmartPort Macros

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

CHAPTER 17

Configuring STP and MST

This chapter describes how to configure the Spanning Tree Protocol (STP) on a Catalyst 4500 series switch. This chapter also describes how to configure the IEEE 802.1s Multiple Spanning Tree (MST) protocol on the Catalyst 4500 series switch. MST is a new IEEE standard derived from Cisco's proprietary Multi-Instance Spanning-Tree Protocol (MISTP) implementation. With MST, you can map a single spanning-tree instance to several VLANs.

This chapter provides guidelines, procedures, and configuration examples. It includes the following major sections:

- Overview of STP, page 17-1
- Default STP Configuration, page 17-6
- Configuring STP, page 17-7
- Overview of MST, page 17-21
- MST Configuration Restrictions and Guidelines, page 17-29
- Configuring MST, page 17-29

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

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of STP

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. A loop-free subset of a network topology is called a spanning tree. The operation of a spanning tree is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

A Catalyst 4500 series switch use STP (the IEEE 802.1D bridge protocol) on all VLANs. By default, a single spanning tree runs on each configured VLAN (provided you do not manually disable the spanning tree). You can enable and disable a spanning tree 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 spanning tree algorithm calculates the best loop-free path throughout a switched Layer 2 network. Switches send and receive spanning tree frames at regular intervals. The switches 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 switches might learn end station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network.

A spanning tree defines a tree with a root switch and a loop-free path from the root to all switches in the Layer 2 network. A spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning tree algorithm recalculates the spanning tree topology and activates the standby path.

When two ports on a switch are part of a loop, the spanning tree 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 spanning tree port priority value represents the location of an interface in the network topology and how well located it is to pass traffic. The spanning tree 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.

Bridge Priority Value

The bridge priority value determines whether a given redundant link will be given priority and considered part of a given span in a spanning tree. Preference is given to lower values, and if you want to manually configure a preference, assign a lower bridge priority value to a link than to its redundant possibility. With Cisco IOS releases prior to 12.1(12c)EW, the bridge priority is a 16-bit value (see Table 17-1).With Cisco IOS Release 12.1(12c)EW and later releases, the bridge priority is a 4-bit value when the extended system ID is enabled (see Table 17-2). See the “Configuring the Bridge Priority of a VLAN” section on page 17-16.

Extended System ID

Extended system IDs are VLAN IDs between 1025 and 4096. Cisco IOS Releases 12.1(12c)EW and later releases support a 12-bit extended system ID field as part of the bridge ID (see Table 17-2). 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 17-8.

Table 17-1 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 17-2 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	VLAN ID											

STP MAC Address Allocation

A Catalyst 4500 series switch chassis has either 64 or 1024 MAC addresses available to support software features like STP. Enter the **show module** command to view the MAC address range on your chassis.

Cisco IOS Release 12.1(12c)EW and later releases support chassis with 64 or 1024 MAC addresses. 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.

Bridge Protocol Data Units

The following elements determine the stable active spanning tree topology of a switched network:

- The unique bridge ID (bridge priority and MAC address) associated with each VLAN on each switch
- The spanning tree path cost (or bridge priority value) to the root bridge
- The port identifier (port priority and MAC address) associated with each Layer 2 interface

Bridge protocol data units (BPDUs) contain information about the transmitting bridge and its ports, including the bridge and MAC addresses, bridge priority, port priority, and path cost. The system computes the spanning tree topology by transmitting BPDUs among connecting switches, and in one direction from the root switch. Each configuration BPDU contains at least the following:

- The unique bridge ID of the switch that the transmitting switch believes to be the root switch
- The spanning tree path cost to the root
- The bridge ID of the transmitting bridge
- The age of the message
- The identifier of the transmitting port
- Values for the *hello*, *forward delay*, and *max-age* protocol timers

When a switch transmits a BPDU frame, all switches connected to the LAN on which the frame is transmitted receive the BPDU. When a switch 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 switch is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each switch based on the path cost.
- A designated bridge for each LAN segment is selected. This is the switch 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 switch with the highest bridge priority (the lowest numerical priority value) is elected as the root bridge. If all switches are configured with the default priority value (32,768), the switch with the lowest MAC address in the VLAN becomes the root bridge.

The spanning tree root bridge is the logical center of the spanning tree topology in a switched network. All paths that are not required to reach the root bridge from anywhere in the switched network are placed in spanning tree blocking mode.

A spanning tree uses the information provided by BPDUs to elect the root bridge and root port for the switched network, as well as the root port and designated port for each switched segment.

STP Timers

Table 17-3 describes the STP timers that affect the performance of the entire spanning tree.

Table 17-3 Spanning Tree Protocol Timers

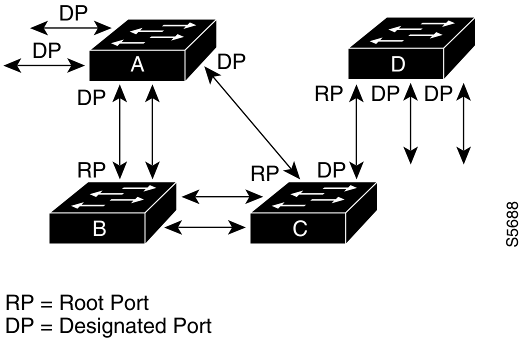
Variable	Description
hello_time	Determines how often the switch broadcasts hello messages to other switches.
forward_time	Determines how long each of the listening and learning states will last before the port begins forwarding.
max_age	Determines the amount of time that protocol information received on a port is stored by the switch.

Creating the STP Topology

The goal of the spanning tree algorithm is to make the most direct link the root 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 optimal according to link speed. 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.

In Figure 17-1, Switch A is elected as the root bridge. (This could happen if the bridge priority of all the switches is set to the default value [32,768] and Switch A has the lowest MAC address.) However, due to traffic patterns, the number of forwarding ports, or link types, Switch A might not be the ideal root bridge. By increasing the STP port priority (lowering the numerical value) of the ideal switch so that it becomes the root bridge, you force a spanning tree recalculation to form a new spanning tree topology with the ideal switch as the root.

Figure 17-1 Spanning Tree Topology



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 spanning tree port priority on the fiber-optic port to a higher priority (lower numerical value) than the priority set for the root port, the fiber-optic port becomes the new root port.

STP Port States

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 interface 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 under the old topology.

Each Layer 2 interface on a switch that uses spanning tree exists in one of the following five states:

- Blocking—In this state, the Layer 2 interface does not participate in frame forwarding.
- Listening—This state is the first transitional state after the blocking state when spanning tree determines that the Layer 2 interface should participate in frame forwarding.
- Learning—In this state, the Layer 2 interface prepares to participate in frame forwarding.
- Forwarding—In this state, the Layer 2 interface forwards frames.
- Disabled—In this state, the Layer 2 interface does not participate in spanning tree and does not forward frames.

MAC Address Allocation

The supervisor engine has a pool of 1024 MAC addresses that are used as the bridge IDs for the VLAN spanning trees. You can use the **show module** command to view the MAC address range (allocation range for the supervisor) that the spanning tree uses for the algorithm.

MAC addresses for the Catalyst 4506 switch are allocated sequentially, with the first MAC address in the range assigned to VLAN 1, the second MAC address in the range assigned to VLAN 2, and so forth. For example, if the MAC address range is 00-e0-1e-9b-2e-00 to 00-e0-1e-9b-31-ff, the VLAN 1 bridge ID is 00-e0-1e-9b-2e-00, the VLAN 2 bridge ID is 00-e0-1e-9b-2e-01, the VLAN 3 bridge ID is 00-e0-1e-9b-2e-02, and so on. On other Catalyst 4500 series platforms, all VLANS map to the same MAC address rather than mapping to separate MAC addresses.

STP and IEEE 802.1Q Trunks

802.1Q VLAN trunks impose some limitations on the spanning tree strategy for a network. In a network of Cisco switches connected through 802.1Q trunks, the switches maintain one instance of spanning tree for each VLAN allowed on the trunks. However, non-Cisco 802.1Q switches maintain only one instance of spanning tree for all VLANs allowed on the trunks.

When you connect a Cisco switch to a non-Cisco device (that supports 802.1Q) through an 802.1Q trunk, the Cisco switch combines the spanning tree instance of the 802.1Q native VLAN of the trunk with the spanning tree instance of the non-Cisco 802.1Q switch. However, all per-VLAN spanning tree information is maintained by Cisco switches separated by a network of non-Cisco 802.1Q switches. The non-Cisco 802.1Q network separating the Cisco switches is treated as a single trunk link between the switches.

Note For more information on 802.1Q trunks, see Chapter 15, "Configuring Layer 2 Ethernet Interfaces."

Per-VLAN Rapid Spanning Tree

Per-VLAN Rapid Spanning Tree (PVRST+) is the same as PVST+, although PVRST+ utilizes a rapid STP based on IEEE 802.1w rather than 802.1D to provide faster convergence. PVRST+ uses roughly the same configuration as PVST+ and needs only minimal configuration. In PVRST+, dynamic CAM entries are flushed immediately on a per-port basis when any topology change is made. UplinkFast and BackboneFast are enabled but not active in this mode, since the functionality is built into the Rapid STP. PVRST+ provides for rapid recovery of connectivity following the failure of a bridge, bridge port, or LAN.

For enabling information, see "Enabling Per-VLAN Rapid Spanning Tree" on page 20.

Default STP Configuration

Table 17-4 shows the default spanning tree configuration.

Table 17-4 Spanning Tree Default Configuration Values

Feature	Default Value
Enable state	Spanning tree enabled for all VLANs
Bridge priority value	32,768
Spanning tree port priority value (configurable on a per-interface basis—used on interfaces configured as Layer 2 access ports)	128
Spanning tree port cost (configurable on a per-interface basis—used on interfaces configured as Layer 2 access ports)	• 10-Gigabit Ethernet: 2 • Gigabit Ethernet: 4 • Fast Ethernet: 19
Spanning tree VLAN port priority value (configurable on a per-VLAN basis—used on interfaces configured as Layer 2 trunk ports)	128

Table 17-4 Spanning Tree Default Configuration Values (continued)

Feature	Default Value
Spanning tree VLAN port cost (configurable on a per-VLAN basis—used on interfaces configured as Layer 2 trunk ports)	• 10-Gigabit Ethernet: 2 • Gigabit Ethernet: 4 • Fast Ethernet: 19
Hello time	2 sec
Forward delay time	15 sec
Maximum aging time	20 sec

Configuring STP

The following sections describe how to configure spanning tree on VLANs:

- Enabling STP, page 17-7
- Enabling the Extended System ID, page 17-8
- Configuring the Root Bridge, page 17-9
- Configuring a Secondary Root Switch, page 17-12
- Configuring STP Port Priority, page 17-13
- Configuring STP Port Cost, page 17-15
- Configuring the Bridge Priority of a VLAN, page 17-16
- Configuring the Hello Time, page 17-17
- Configuring the Maximum Aging Time for a VLAN, page 17-18
- Configuring the Forward-Delay Time for a VLAN, page 17-18
- Disabling Spanning Tree Protocol, page 17-19
- Enabling Per-VLAN Rapid Spanning Tree, page 17-20

Note The spanning tree commands described in this chapter can be configured on any interface except those configured with the **no switchport** command.

Enabling STP

Note By default, spanning tree is enabled on all the VLANs.

You can enable a spanning tree on a per-VLAN basis. The switch maintains a separate instance of spanning tree for each VLAN (except on VLANs on which you have disabled a spanning tree).

To enable a spanning tree on a per-VLAN basis, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`spanning-tree vlan`** *`vlan_ID`*	Enables spanning tree for VLAN *vlan_id*. The *vlan_ID* value can range from 1 to 4094.
Step 3	`Switch(config)#` **`end`**	Exits configuration mode.
Step 4	`Switch#` **`show spanning-tree vlan`** *`vlan_ID`*	Verifies that spanning tree is enabled.

This example shows how to enable a spanning tree on VLAN 200:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200
Switch(config)# end
Switch#
```

Note Because spanning tree is enabled by default, issuing a **show running** command to view the resulting configuration will not display the command you entered to enable spanning tree.

This example shows how to verify that spanning tree is enabled on VLAN 200:

```
Switch# show spanning-tree vlan 200

 VLAN200 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 0050.3e8d.6401
  Configured hello time 2, max age 20, forward delay 15
  Current root has priority 16384, address 0060.704c.7000
  Root port is 264 (FastEthernet5/8), cost of root path is 38
  Topology change flag not set, detected flag not set
  Number of topology changes 0 last change occurred 01:53:48 ago
  Times:  hold 1, topology change 24, notification 2
          hello 2, max age 14, forward delay 10
  Timers: hello 0, topology change 0, notification 0

 Port 264 (FastEthernet5/8) of VLAN200 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 129.9.
   Designated root has priority 16384, address 0060.704c.7000
   Designated bridge has priority 32768, address 00e0.4fac.b000
   Designated port id is 128.2, designated path cost 19
   Timers: message age 3, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 3, received 3417

Switch#
```

Enabling the Extended System ID

Note The extended system ID is enabled permanently on chassis that support 64 MAC addresses.

You can use the **spanning-tree extend system-id** command to enable the extended system ID on chassis that support 1024 MAC addresses. See the "Understanding the Bridge ID" section on page 17-2.

To enable the extended system ID, perform this task:

	Command	Purpose
Step 1	`Switch(config)# spanning-tree extend system-id`	Enables the extended 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 "Table 17-4Spanning Tree Default Configuration Values" section on page 17-6).
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree vlan vlan_ID`	Verifies the configuration.

Note When you enable or disable the extended system ID, the bridge IDs of all active STP instances are updated, which might change the spanning tree topology.

This example shows how to enable the extended system ID:

```
Switch# configure terminal
Switch(config)# spanning-tree extend system-id
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

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

Configuring the Root Bridge

A Catalyst 4000 family switch maintains an instance of spanning tree for each active VLAN configured on the switch. A bridge ID, consisting of the bridge priority and the bridge MAC address, is associated with each instance. For each VLAN, the switch with the lowest bridge ID will become the root bridge for that VLAN. Whenever the bridge priority changes, the bridge ID also changes. This results in the recomputation of the root bridge for the VLAN.

To configure a switch to become the root bridge for the specified VLAN, use the **spanning-tree vlan** *vlan-ID* **root** command to modify the bridge priority from the default value (32,768) to a significantly lower value. The bridge priority for the specified VLAN is set to 8192 if this value will cause the switch to become the root for the VLAN. If any bridge for the VLAN has a priority lower than 8192, the switch sets the priority to 1 less than the lowest bridge priority.

For example, assume that all the switches in the network have the bridge priority for VLAN 100 set to the default value of 32,768. Entering the **spanning-tree vlan 100 root primary** command on a switch will set the bridge priority for VLAN 100 to 8192, causing this switch to become the root bridge for VLAN 100.

Note The root switch for each instance of spanning tree should be a backbone or distribution switch. Do not configure an access switch as the spanning tree primary root.

Use the **diameter** keyword to specify the Layer 2 network diameter (the maximum number of bridge hops between any two end stations in the network). When you specify the network diameter, a switch automatically picks an optimal hello time, forward delay time, and maximum age time for a network of that diameter. This can significantly reduce the spanning tree convergence time.

Use the **hello-time** keyword to override the automatically calculated hello time.

Note We recommend that you avoid manually configuring the hello time, forward delay time, and maximum age time after configuring the switch as the root bridge.

To configure a switch as the root switch, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree vlan vlan_ID root primary [diameter hops [hello-time seconds]]`	Configures a switch as the root switch. You can use the **no** keyword to restore the defaults.
Step 2	`Switch(config)# end`	Exits configuration mode.

This example shows how to configure a switch as the root bridge for VLAN 10, with a network diameter of 4:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 10 root primary diameter 4
Switch(config)# end
Switch#
```

This example shows how the configuration changes when a switch becomes a spanning tree root. This is the configuration before the switch becomes the root for VLAN 1:

```
Switch#show spanning-tree vlan 1

VLAN1 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 0030.94fc.0a00
  Configured hello time 2, max age 20, forward delay 15
  Current root has priority 32768, address 0001.6445.4400
  Root port is 323 (FastEthernet6/3), cost of root path is 19
  Topology change flag not set, detected flag not set
  Number of topology changes 2 last change occurred 00:02:19 ago
          from FastEthernet6/1
  Times: hold 1, topology change 35, notification 2
          hello 2, max age 20, forward delay 15
  Timers:hello 0, topology change 0, notification 0, aging 300

 Port 323 (FastEthernet6/3) of VLAN1 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 129.67.
   Designated root has priority 32768, address 0001.6445.4400
   Designated bridge has priority 32768, address 0001.6445.4400
   Designated port id is 129.67, designated path cost 0
   Timers:message age 2, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   BPDU:sent 3, received 91
 Port 324 (FastEthernet6/4) of VLAN1 is blocking
   Port path cost 19, Port priority 128, Port Identifier 129.68.
   Designated root has priority 32768, address 0001.6445.4400
   Designated bridge has priority 32768, address 0001.6445.4400
   Designated port id is 129.68, designated path cost 0
   Timers:message age 2, forward delay 0, hold 0
   Number of transitions to forwarding state:0
   BPDU:sent 1, received 89
```

Now, you can set the switch as the root:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 1 root primary
Switch(config)# spanning-tree vlan 1 root primary
 VLAN 1 bridge priority set to 8192
 VLAN 1 bridge max aging time unchanged at 20
 VLAN 1 bridge hello time unchanged at 2
 VLAN 1 bridge forward delay unchanged at 15
Switch(config)# end
```

This is the configuration after the switch becomes the root:

```
Switch# show spanning-tree vlan 1

VLAN1 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 8192, address 0030.94fc.0a00
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag set, detected flag set
  Number of topology changes 3 last change occurred 00:00:09 ago
  Times: hold 1, topology change 35, notification 2
          hello 2, max age 20, forward delay 15
  Timers:hello 0, topology change 25, notification 0, aging 15

Port 323 (FastEthernet6/3) of VLAN1 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 129.67.
   Designated root has priority 8192, address 0030.94fc.0a00
   Designated bridge has priority 8192, address 0030.94fc.0a00
   Designated port id is 129.67, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   BPDU:sent 9, received 105

 Port 324 (FastEthernet6/4) of VLAN1 is listening
   Port path cost 19, Port priority 128, Port Identifier 129.68.
   Designated root has priority 8192, address 0030.94fc.0a00
   Designated bridge has priority 8192, address 0030.94fc.0a00
   Designated port id is 129.68, designated path cost 0
   Timers:message age 0, forward delay 5, hold 0
   Number of transitions to forwarding state:0
   BPDU:sent 6, received 102

Switch#
```

Note Because the bridge priority is now set at 8192, this switch becomes the root of the spanning tree.

Configuring a Secondary Root Switch

When you configure a switch as the secondary root, the spanning tree bridge priority is modified from the default value (32,768) to 16,384. This means that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other switches in the network use the default bridge priority of 32,768).

You can run this command on more than one switch to configure multiple backup root switches. Use the same network diameter and hello time values that you used when configuring the primary root switch.

Note We recommend that you avoid manually configuring the hello time, forward delay time, and maximum age time after configuring the switch as the root bridge.

To configure a switch as the secondary root switch, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree vlan vlan_ID root secondary [diameter hops [hello-time seconds]]`	Configures a switch as the secondary root switch. You can use the **no** keyword to restore the defaults.
Step 2	`Switch(config)# end`	Exits configuration mode.

This example shows how to configure the switch as the secondary root switch for VLAN 10, with a network diameter of 4:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 10 root secondary diameter 4
VLAN 10 bridge priority set to 16384
 VLAN 10 bridge max aging time set to 14
 VLAN 10 bridge hello time unchanged at 2
 VLAN 10 bridge forward delay set to 10
Switch(config)# end
Switch#
```

This example shows how to verify the configuration of VLAN 1:

```
Switch#sh spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     0003.6b10.e800
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768
             Address     0003.6b10.e800
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
Fa3/1            Desg FWD 19        128.129  P2p
Fa3/2            Desg FWD 19        128.130  P2p
Fa3/48           Desg FWD 19        128.176  Edge P2p

Switch#
```

Configuring STP Port Priority

In the event of a loop, a spanning tree considers port priority when selecting an interface to put into the forwarding state. You can assign higher priority values to interfaces that you want a spanning tree to select first and lower priority values to interfaces that you want a spanning tree to select last. If all interfaces have the same priority value, a spanning tree puts the interface with the lowest interface number in the forwarding state and blocks other interfaces. The possible priority range is 0 through 240, configurable in increments of 16 (the default is 128).

Note The Cisco IOS software uses the port priority value when the interface is configured as an access port and uses VLAN port priority values when the interface is configured as a trunk port.

To configure the spanning tree port priority of an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`interface {{fastethernet | gigabitethernet | tengigabitethernet}`** *slot/port*} | {**port-channel** *port_channel_number*}	Specifies an interface to configure.
Step 2	`Switch(config-if)#` [**no**] **spanning-tree port-priority** *port_priority*	Configures the port priority for an interface. The *port_priority* value can be from 0 to 240, in increments of 16. You can use the **no** keyword to restore the defaults.
Step 3	`Switch(config-if)#` [**no**] **spanning-tree vlan** *vlan_ID* **port-priority** *port_priority*	Configures the VLAN port priority for an interface. The *port_priority* value can be from 0 to 240, in increments of 16. You can use the **no** keyword to restore the defaults.
Step 4	`Switch(config-if)#` **end**	Exits configuration mode.
Step 5	`Switch#` **show spanning-tree interface** {{**fastethernet** | **gigabitethernet**} *slot/port*} | {**port-channel** *port_channel_number*} **show spanning-tree vlan** *vlan_ID*	Verifies the configuration.

This example shows how to configure the spanning tree port priority of a Fast Ethernet interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree port-priority 100
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration of a Fast Ethernet interface when it is configured as an access port:

```
Switch# show spanning-tree interface fastethernet 3/1

Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0001         Desg FWD 19        128.129  P2p
VLAN1002         Desg FWD 19        128.129  P2p
VLAN1003         Desg FWD 19        128.129  P2p
VLAN1004         Desg FWD 19        128.129  P2p
VLAN1005         Desg FWD 19        128.129  P2p
Switch#
```

This example shows how to display the details of the interface configuration when the interface is configured as an access port:

```
Switch# show spanning-tree interface fastethernet 3/1 detail
Port 129 (FastEthernet3/1) of VLAN0001 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.129.
   Designated root has priority 32768, address 0003.6b10.e800
   Designated bridge has priority 32768, address 0003.6b10.e800
   Designated port id is 128.129, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   Link type is point-to-point by default
   BPDU:sent 187, received 1

 Port 129 (FastEthernet3/1) of VLAN1002 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.129.
   Designated root has priority 32768, address 0003.6b10.ebe9
   Designated bridge has priority 32768, address 0003.6b10.ebe9
   Designated port id is 128.129, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   Link type is point-to-point by default
   BPDU:sent 94, received 2

 Port 129 (FastEthernet3/1) of VLAN1003 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.129.
   Designated root has priority 32768, address 0003.6b10.ebea
   Designated bridge has priority 32768, address 0003.6b10.ebea
   Designated port id is 128.129, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   Link type is point-to-point by default
   BPDU:sent 94, received 2

 Port 129 (FastEthernet3/1) of VLAN1004 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.129.
   Designated root has priority 32768, address 0003.6b10.ebeb
   Designated bridge has priority 32768, address 0003.6b10.ebeb
   Designated port id is 128.129, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   Link type is point-to-point by default
   BPDU:sent 95, received 2

 Port 129 (FastEthernet3/1) of VLAN1005 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.129.
   Designated root has priority 32768, address 0003.6b10.ebec
   Designated bridge has priority 32768, address 0003.6b10.ebec
   Designated port id is 128.129, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   Link type is point-to-point by default
   BPDU:sent 95, received 2
Switch#
```

Note The **show spanning-tree port-priority** command displays only information for ports with an active link. If there is no port with an active link, enter a **show running-config interface** command to verify the configuration.

This example shows how to configure the spanning tree VLAN port priority of a Fast Ethernet interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree vlan 200 port-priority 64
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration of VLAN 200 on the interface when it is configured as a trunk port:

```
Switch# show spanning-tree vlan 200
<...output truncated...>

Port 264 (FastEthernet5/8) of VLAN200 is forwarding
Port path cost 19, Port priority 64, Port Identifier 129.8.
   Designated root has priority 32768, address 0010.0d40.34c7
   Designated bridge has priority 32768, address 0010.0d40.34c7
   Designated port id is 128.1, designated path cost 0
   Timers: message age 2, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 0, received 13513

<...output truncated...>
Switch#
```

Configuring STP Port Cost

The default value for spanning tree port path cost is derived from the interface media speed. In the event of a loop, spanning tree considers port cost when selecting an interface to put into the forwarding state. You can assign lower cost values to interfaces that you want spanning tree to select first, and higher cost values to interfaces that you want spanning tree to select last. If all interfaces have the same cost value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks other interfaces. The possible cost range is 1 through 200,000,000 (the default is media-specific).

Spanning tree uses the port cost value when the interface is configured as an access port and uses VLAN port cost values when the interface is configured as a trunk port.

To configure the spanning tree port cost of an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# `**`interface`**` {{`**`fastethernet`**` | `**`gigabitethernet`**` | `**`tengigabitethernet`**`} `*`slot/port`*`} | {`**`port-channel`**` `*`port_channel_number`*`}`	Specifies an interface to configure.
Step 2	`Switch(config-if)# [`**`no`**`] `**`spanning-tree cost`**` `*`port_cost`*	Configures the port cost for an interface. The *port_cost* value can be from 1 to 200,000,000. You can use the **no** keyword to restore the defaults.
Step 3	`Switch(config-if)# [`**`no`**`] `**`spanning-tree vlan`**` `*`vlan_ID`*` `**`cost`**` `*`port_cost`*	Configures the VLAN port cost for an interface. The *port_cost* value can be from 1 to 200,000,000. You can use the **no** keyword to restore the defaults.
Step 4	`Switch(config-if)# `**`end`**	Exits configuration mode.

Step 5	`Switch# `**`show spanning-tree interface`**` {{`**`fastethernet`**` | `**`gigabitethernet`**`} `*`slot/port`*`} | {`**`port-channel`**` `*`port_channel_number`*`}`
	`show spanning-tree vlan`` `*`vlan_ID`*
		Verifies the configuration.

This example shows how to change the spanning tree port cost of a Fast Ethernet interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree cost 18
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration of the interface when it is configured as an access port:

```
Switch# show spanning-tree interface fastethernet 5/8
 Port 264 (FastEthernet5/8) of VLAN200 is forwarding
   Port path cost 18, Port priority 100, Port Identifier 129.8.
   Designated root has priority 32768, address 0010.0d40.34c7
   Designated bridge has priority 32768, address 0010.0d40.34c7
   Designated port id is 128.1, designated path cost 0
   Timers: message age 2, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 0, received 13513
Switch#
```

This example shows how to configure the spanning tree VLAN port cost of a Fast Ethernet interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree vlan 200 cost 17
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration of VLAN 200 on the interface when it is configured as a trunk port:

```
Switch# show spanning-tree vlan 200
<...output truncated...>
Port 264 (FastEthernet5/8) of VLAN200 is forwarding
Port path cost 17, Port priority 64, Port Identifier 129.8.
   Designated root has priority 32768, address 0010.0d40.34c7
   Designated bridge has priority 32768, address 0010.0d40.34c7
   Designated port id is 128.1, designated path cost 0
   Timers: message age 2, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 0, received 13513

<...output truncated...>
Switch#
```

Note The **show spanning-tree** command displays only information for ports with an active link (green light is on). If there is no port with an active link, you can issue a **show running-config** command to confirm the configuration.

Configuring the Bridge Priority of a VLAN

Note Exercise care when configuring the bridge priority of a VLAN. In most cases, 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 spanning tree bridge priority of a VLAN, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree vlan vlan_ID priority bridge_priority`	Configures the bridge priority of a VLAN. The *bridge_priority* value can be from 1 to 65,534. You can use the **no** keyword to restore the defaults.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree vlan vlan_ID bridge [brief]`	Verifies the configuration.

This example shows how to configure the bridge priority of VLAN 200 to 33,792:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200 priority 33792
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200 bridge brief
                                      Hello Max  Fwd
Vlan                   Bridge ID      Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          33792 0050.3e8d.64c8    2   20    15  ieee
Switch#
```

Configuring the Hello Time

Note Exercise care when configuring the hello time. In most cases, we recommend that you use the **spanning-tree vlan** *vlan_ID* **root primary** and the **spanning-tree vlan** *vlan_ID* **root secondary** commands to modify the hello time.

To configure the spanning tree hello time of a VLAN, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree vlan vlan_ID hello-time hello_time`	Configures the hello time of a VLAN. The *hello_time* value can be from 1 to 10 seconds. You can use the **no** keyword to restore the defaults.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree vlan vlan_ID bridge [brief]`	Verifies the configuration.

This example shows how to configure the hello time for VLAN 200 to 7 seconds:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200 hello-time 7
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200 bridge brief
                                      Hello Max  Fwd
Vlan                   Bridge ID      Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          49152 0050.3e8d.64c8    7   20    15  ieee
Switch#
```

Configuring the Maximum Aging Time for a VLAN

Note Exercise care when configuring aging time. In most cases, we recommend that you use the **spanning-tree vlan** *vlan_ID* **root primary** and the **spanning-tree vlan** *vlan_ID* **root secondary** commands to modify the maximum aging time.

To configure the spanning tree maximum aging time for a VLAN, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [`**`no`**`]` **`spanning-tree vlan`** *`vlan_ID`* **`max-age`** *`max_age`*	Configures the maximum aging time of a VLAN. The *max_age* value can be from 6 to 40 seconds. You can use the **no** keyword to restore the defaults.
Step 2	`Switch(config)#` **`end`**	Exits configuration mode.
Step 3	`Switch#` **`show spanning-tree vlan`** *`vlan_ID`* **`bridge`** `[`**`brief`**`]`	Verifies the configuration.

This example shows how to configure the maximum aging time for VLAN 200 to 36 seconds:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200 max-age 36
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200 bridge brief
                                      Hello Max  Fwd
Vlan                   Bridge ID      Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          49152 0050.3e8d.64c8    2   36    15  ieee
Switch#
```

Configuring the Forward-Delay Time for a VLAN

Note Exercise care when configuring forward-delay time. In most cases, we recommend that you use the **spanning-tree vlan** *vlan_ID* **root primary** and the **spanning-tree vlan** *vlan_ID* **root secondary** commands to modify the forward delay time.

To configure the spanning tree forward delay time for a VLAN, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree vlan vlan_ID forward-time forward_time`	Configures the forward time of a VLAN. The *forward_time* value can be from 4 to 30 seconds. You can use the **no** keyword to restore the defaults.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree vlan vlan_ID bridge [brief]`	Verifies the configuration.

This example shows how to configure the forward delay time for VLAN 200 to 21 seconds:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200 forward-time 21
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200 bridge brief
                                      Hello Max  Fwd
Vlan                   Bridge ID      Time  Age Delay  Protocol
---------------- -------------------- ---- ---- -----  --------
VLAN200          49152 0050.3e8d.64c8    2   20    21  ieee
Switch#
```

This example shows how to display spanning tree information for the bridge:

```
Switch# show spanning-tree bridge

                                                   Hello  Max  Fwd
Vlan                         Bridge ID              Time  Age  Dly  Protocol
---------------- --------------------------------- -----  ---  ---  --------
VLAN200               49152 0050.3e8d.64c8           2    20   15  ieee
VLAN202               49152 0050.3e8d.64c9           2    20   15  ieee
VLAN203               49152 0050.3e8d.64ca           2    20   15  ieee
VLAN204               49152 0050.3e8d.64cb           2    20   15  ieee
VLAN205               49152 0050.3e8d.64cc           2    20   15  ieee
VLAN206               49152 0050.3e8d.64cd           2    20   15  ieee
Switch#
```

Disabling Spanning Tree Protocol

To disable spanning tree on a per-VLAN basis, perform this task:

	Command	Purpose
Step 1	`Switch(config)# no spanning-tree vlan vlan_ID`	Disables spanning tree on a per-VLAN basis.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree vlan vlan_ID`	Verifies that spanning tree is disabled.

This example shows how to disable spanning tree on VLAN 200:

```
Switch# configure terminal
Switch(config)# no spanning-tree vlan 200
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200
Spanning tree instance for VLAN 200 does not exist.
Switch#
```

Enabling Per-VLAN Rapid Spanning Tree

Per-VLAN Rapid Spanning Tree (PVRST+) uses the existing PVST+ framework for configuration purposes and for interaction with other features. It also supports some of the PVST+ extensions.

To configure PVRST+, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spantree mode rapid-pvst`	Enables rapid-PVST+.
Step 2	`Switch(config)# interface interface/port`	Switches to interface configuration mode.
Step 3	`Switch(config)# spanning-tree link-type point-to-point`	Sets the link-type to point-to-point mode for the port.
Step 4	`Switch(config-if)# exit`	Exits interface mode.
Step 5	`Switch(config)# exit`	Exits configuration mode.
Step 6	`Switch(config-if)# clear spantree detected-protocols mod/port`	Detects any legacy bridges on the port
Step 7	`Switch# show spanning-tree summary totals`	Verifies the rapid-PVST+ configuration.

The following example shows how to configure Rapid-PVST+:

```
Switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# spanning-tree mode rapid-pvst
Switch(config)# int fa 6/4
Switch(config-if)# spanning-tree link-type point-to-point
Switch(config-if)# end
Switch(config)# end
Switch#
23:55:32:%SYS-5-CONFIG_I:Configured from console by console
Switch# clear spanning-tree detected-protocols
```

The following example shows how to verify the configuration:

```
Switch# show spanning-tree summary totals
Switch is in rapid-pvst mode
Root bridge for:VLAN0001
Extended system ID           is disabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
EtherChannel misconfig guard is enabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Pathcost method used         is short
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
1 vlan                       0         0        0          2          2
Switch#
```

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, use 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. Furthermore, when an MSTP switch receives a legacy BPDU, it can also detect the following:

- that a port is at the boundary of a region
- an MST BPDU (version 3) associated with a different region, or
- an RST BPDU (version 2).

The switch, however, does not automatically revert to the MSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether or not 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 on the entire switch (that is, to force renegotiation with neighboring switches), use the **clear spanning-tree detected-protocols** commands in privileged EXEC mode. To restart the protocol migration process on a specific interface, enter the **clear spanning-tree detected-protocols interface** command *in interface-id* privileged EXEC mode.

Overview of MST

The following sections describe how MST works on a Catalyst 4000 family switch:

- IEEE 802.1s MST, page 17-22
- IEEE 802.1w RSTP, page 17-23

- MST-to-SST Interoperability, page 17-24
- Common Spanning Tree, page 17-25
- MST Instances, page 17-26
- MST Configuration Parameters, page 17-26
- MST Regions, page 17-26
- Message Age and Hop Count, page 17-28
- MST-to-PVST+ Interoperability, page 17-28

IEEE 802.1s MST

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 Per VLAN Spanning Tree Plus (PVST+) and 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 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.

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, MSTP. MST has the following characteristics:

- MST runs a variant of spanning tree called Internal Spanning Tree (IST). IST augments 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 SST bridges as follows:
  - MST bridges run IST, which augments 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 the following: ISTs in each MST region, the CST that interconnects the MST regions, and the SST bridges. CIST is identical to an IST inside an MST region and identical to a 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 termed 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 and 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 bridge protocol data units (BPDUs). The original spanning trees computed by MSTP are called M-trees, which 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 part of 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), you must map a secondary VLAN to the same instance as the primary.

IEEE 802.1w RSTP

RSTP, specified in 802.1w, supersedes STP specified in 802.1D, but remains compatible with STP. You configure RSTP when you configure the MST feature. For more information, see the “Configuring MST” section on page 17-29.

RSTP provides the structure on which the MST operates, significantly reducing the time to reconfigure the active topology of a network when its physical topology or configuration parameters change. 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 the forwarding state through an explicit handshake between them. RSTP allows switch port configuration so the ports can transition to forwarding directly when the switch reinitializes.

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

In RSTP, the port roles are defined as follows:

- 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 mode 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.

The system assigns port roles 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 17-5 shows the STP port states and RSTP port states.

Table 17-5 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.

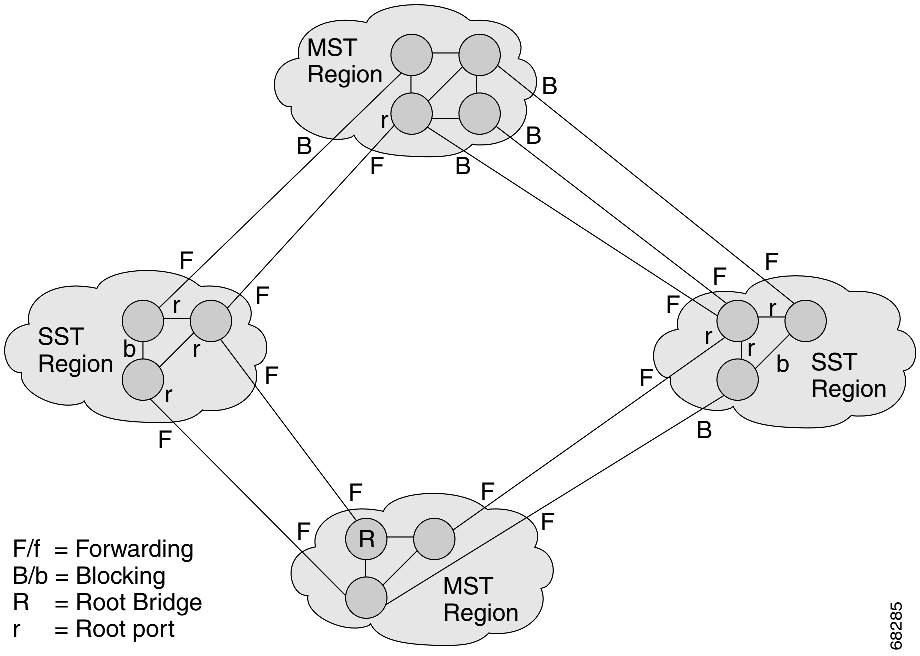
[2] IEEE 802.1w port state designation. Discarding is the same as blocking in MST.

In a stable topology, RSTP ensures that every root port and designated port transitions to the forwarding state while all alternate ports and backup ports are always in the discarding state.

MST-to-SST Interoperability

A virtual bridged LAN may contain interconnected regions of SST and MST bridges. Figure 17-2 shows this relationship.

Figure 17-2 Network with Interconnected SST and MST Regions



To STP running in the SST region, an MST region appears as a single SST or pseudobridge, which operates as follows:

- Although the values for root identifiers and root path costs match for all BPDUs in all pseudobridges, a pseudobridge differs 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 one second for each hop, the difference in the message age is measured in 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 might follow different paths within the MST regions established by MST.
- The system prevents looping by doing 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.

Common Spanning Tree

CST (802.1Q) is a single spanning tree for all the VLANs. In a Catalyst 4500 series switch running PVST+, the VLAN 1 spanning tree corresponds to CST. In a Catalyst 4500 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 has three parts, as follows:

- Name—A 32-character string (null padded) that identifies 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 either value is different, the MST BPDU is considered to be an SST BPDU.

MST Regions

These sections describe MST regions:

- MST Region Overview, page 17-26
- Boundary Ports, page 17-27
- IST Master, page 17-27
- Edge Ports, page 17-27
- Link Type, page 17-28

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 mechanism assigns a port role to the boundary and the same state as that 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. Enter the **show spanning-tree mst** command to display the information about the IST master, path cost, and remaining hops for the bridge.

Edge Ports

A port that is connected to a nonbridging device (for example, a host or a switch) is an edge port. A port that connects to a hub is also an edge port if the hub or any LAN that is connected to it does not have a bridge. An edge port can start forwarding as soon as the link is up.

MST requires that you configure each port connected to a host. To establish rapid connectivity after a failure, you need to block the non-edge 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 switches as edge ports while using MST.

To prevent a misconfiguration, the PortFast operation is turned off if the port receives a BPDU. You can display the configured and operational status of PortFast by using 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 switch. 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 mechanism that is very similar to the IP time-to live (TTL) mechanism. 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.

MST-to-PVST+ Interoperability

Keep these guidelines in mind when you configure MST switches (in the same region) to interact with PVST+ switches:

- Configure the root for all VLANs inside the MST region as shown in this example:

  ```
  Switch# 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 Vlans 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 renewable 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.

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 such a case, the topology changes are propagated only in the instance to which the VLAN is mapped. The topology change stays local to the first MST region, and the Cisco Access Manager (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.

MST Configuration Restrictions and Guidelines

Follow these restrictions and guidelines to avoid configuration problems:

- Do not disable spanning tree on any VLAN in any of the PVST bridges.
- Do no use PVST bridges as the root of CST.
- Do not connect switches with access links, because access links may partition a VLAN.
- Ensure that all PVST 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.
- Complete any MST configuration that incorporates a large number of either existing or new logical VLAN ports 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 MST

The following sections describe how to configure MST:

- Enabling MST, page 17-29
- Configuring MST Instance Parameters, page 17-31
- Configuring MST Instance Port Parameters, page 17-32
- Restarting Protocol Migration, page 17-33
- Displaying MST Configurations, page 17-33

Enabling MST

To enable and configure MST on a Catalyst 4500, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`spanning-tree mode mst`**	Enters MST mode.
Step 2	`Switch(config)#` **`spanning-tree mst configuration`**	Enters MST configuration submode. You can use the **no** keyword to clear the MST configuration.
Step 3	`Switch(config-mst)#` **`show current`**	Displays the current MST configuration.
Step 4	`Switch(config-mst)#` **`name`** *`name`*	Sets the MST region name.
Step 5	`Switch(config-mst)#` **`revision`** *`revision_number`*	Sets the MST configuration revision number.

	Command	Purpose
Step 6	`Switch(config-mst)#` **`instance`** *`instance_number`* **`vlan`** *`vlan_range`*	Maps the VLANs to an MST instance. If you do not specify the **vlan** keyword, you can use the **no** keyword to unmap all the VLANs that were mapped to an MST instance. If you specify the **vlan** keyword, you can use the **no** keyword to unmap a specified VLAN from an MST instance.
Step 7	`Switch(config-mst)#` **`show pending`**	Displays the new MST configuration to be applied.
Step 8	`Switch(config-mst)#` **`end`**	Applies the configuration and exit MST configuration submode.
Step 9	`Switch#` **`show spanning-tree mst configuration`**	Displays the current MST configuration.

This example show how to enable MST:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# spanning-tree mode mst

Switch(config)# spanning-tree mst configuration

Switch(config-mst)# show current
Current MST configuration
Name      []
Revision  0
Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         1-4094
-------------------------------------------------------------------------------
Switch(config-mst)# name cisco
Switch(config-mst)# revision 2
Switch(config-mst)# instance 1 vlan 1
Switch(config-mst)# instance 2 vlan 1-1000
Switch(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         1001-4094
2         1-1000
-------------------------------------------------------------------------------
Switch(config-mst)# no instance 2
Switch(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         1-4094
-------------------------------------------------------------------------------
Switch(config-mst)# instance 1 vlan 2000-3000
Switch(config-mst)# no instance 1 vlan 1500
Switch(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
-------------------------------------------------------------------------------
Switch(config-mst)# end
Switch(config)# no spanning-tree mst configuration
Switch(config)# end
Switch# show spanning-tree mst configuration
Name      []
Revision  0
Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         1-4094
-------------------------------------------------------------------------------
```

Configuring MST Instance Parameters

To configure MST instance parameters, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **spanning-tree mst** *X* **priority** *Y*	Configures the priority for an MST instance.
Step 2	`Switch(config)#` **spanning-tree mst** *X* **root** [**primary** | **secondary**]	Configures the bridge as root for an MST instance.
Step 3	`Switch(config)#` **Ctrl-Z**	Exits configuration mode.
Step 4	`Switch#` **show spanning-tree mst**	Verifies the configuration.

This example shows how to configure MST instance parameters:

```
Switch(config)# spanning-tree mst 1 priority ?
 <0-61440>  bridge priority in increments of 4096

Switch(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

Switch(config)# spanning-tree mst 1 priority 49152
Switch(config)#

Switch(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
Switch(config)# ^Z
Switch#

Switch# 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      240.196  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)

Switch#
```

Configuring MST Instance Port Parameters

To configure MST instance port parameters, perform this task:

	Command	Purpose
Step 1	`Switch(config-if)#` **`spanning-tree mst`** *`x`* **`cost`** *`y`*	Configures the MST instance port cost.
Step 2	`Switch(config-if)#` **`spanning-tree mst`** *`x`* **`port-priority`** *`y`*	Configures the MST instance port priority.
Step 3	`Switch(config-if)#` **`Ctrl-Z`**	Exits configuration mode.
Step 4	`Switch#` **`show spanning-tree mst`** *`x`* **`interface`** *`y`*	Verifies the configuration.

This example shows how to configure MST instance port parameters:

```
Switch(config)# interface fastethernet 4/4
Switch(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

Switch(config-if)# spanning-tree mst 1 cost 1234567

Switch(config-if)# spanning-tree mst 1 port-priority 240
Switch(config-if)# ^Z

Switch# 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

Switch#
```

Restarting Protocol Migration

RSTP and MST have built-in compatibility mechanisms that allow them to interact properly with other regions or other versions of IEEE spanning-tree. For example, an RSTP bridge connected to a legacy bridge can send 802.1D BPDUs on one of its ports. Similarly, when an MST bridge receives a legacy BPDU or an MST BPDU associated with a different region, it is also to detect that a port is at the boundary of a region.

Unfortunately, these mechanisms cannot always revert to the most efficient mode. For example, an RSTP bridge designated for a legacy 802.1D will stay in 802.1D mode even after the legacy bridge has been removed from the link. Similarly, an MST port still assumes that it is a boundary port when the bridge(s) to which it is connected have joined the same region. To force a Catalyst 4500 series switch to renegotiate with the neighbors (that is, to restart protocol migration), you must enter the **clear spanning-tree detected-protocols** command, as follows:

```
Switch# clear spanning-tree detected-protocols fastethernet 4/4
Switch#
```

Displaying MST Configurations

To display MST configurations, perform this task:

	Command	Purpose
Step 1	`Switch# show spanning-tree mst configuration`	Displays the active region configuration information.
Step 2	`Switch# show spanning-tree mst [detail]`	Displays detailed MST protocol information.
Step 3	`Switch# show spanning-tree mst instance-id [detail]`	Displays information about a specific MST instance.
Step 4	`Switch# show spanning-tree mst interface interface [detail]`	Displays information for a given port.
Step 5	`Switch# show spanning-tree mst instance-id interface interface [detail]`	Displays MST information for a given port and a given instance.
Step 6	`Switch# show spanning-tree vlan vlan_ID`	Displays VLAN information in MST mode.

The following examples show how to display spanning tree VLAN configurations in MST mode:

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance 1 vlan 1-10
Switch(config-mst)# name cisco
Switch(config-mst)# revision 1
Switch(config-mst)# Ctrl-D

Switch# show spanning-tree mst configuration
Name      [cisco]
Revision  1
Instance  Vlans mapped
--------  ---------------------------------------------------------------------
0         11-4094
1         1-10
-------------------------------------------------------------------------------
```

```
Switch# 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      240.196  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      240.196  P2p
Fa4/5            Desg FWD 200000    128.197  P2p
Fa4/48           Boun FWD 200000    128.240  P2p Bound(STP)


Switch# 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      240.196  P2p
Fa4/5            Desg FWD 200000    128.197  P2p
Fa4/48           Boun FWD 200000    128.240  P2p Bound(STP)

Switch# 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      240.196  11-4094
1        Back BLK 1000      240.196  1-10


Switch# 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      240.196  1-10

Switch# 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        240.196  priority    240  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

Switch# 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      240.196  P2p
Fa4/5            Desg FWD 200000    128.197  P2p

Switch# 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
Switch#
```

C H A P T E R 18

Configuring Optional STP Features

This chapter describes the Spanning Tree Protocol (STP) features supported on the Catalyst 4500 series switches. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- Overview of Root Guard, page 18-2
- Enabling Root Guard, page 18-2
- Overview of Loop Guard, page 18-3
- Enabling Loop Guard, page 18-4
- Overview of PortFast, page 18-6
- Enabling PortFast, page 18-6
- Overview of BPDU Guard, page 18-7
- Enabling BackboneFast, page 18-15
- Overview of PortFast BPDU Filtering, page 18-8
- Enabling BackboneFast, page 18-15
- Overview of UplinkFast, page 18-10
- Enabling UplinkFast, page 18-11
- Overview of BackboneFast, page 18-12
- Enabling BackboneFast, page 18-15

Note For information on configuring STP, see Chapter 17, "Configuring STP and MST."

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of Root Guard

Spanning Tree root guard forces an interface to become a designated port, to protect the current root status and prevent surrounding switches from becoming the root switch.

When you enable root guard on a per-port basis, it is automatically applied to all of the active VLANs to which that port belongs. When you disable root guard, it is disabled for the specified port and the port automatically goes into the listening state.

When a switch that has ports with root guard enabled detects a new root, the ports goes into root-inconsistent state. Then, when the switch no longer detects a new root, its ports automatically go into the listening state.

Enabling Root Guard

To enable root guard on a Layer 2 access port (to force it to become a designated port), perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {{fastethernet	gigabitethernet	tengigabitethernet} slot/port}`	Specifies an interface to configure.
Step 2	`Switch(config-if)# [no] spanning-tree guard root`	Enables root guard. You can use the **no** keyword to disable Root Guard.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show spanning-tree`	Verifies the configuration.

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

```
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree guard root
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show running-config interface fastethernet 5/8
Building configuration...

Current configuration: 67 bytes
!
interface FastEthernet5/8
 switchport mode access
 spanning-tree guard root
end

Switch#
```

This example shows how to determine whether any ports are in root inconsistent state:

```
Switch# show spanning-tree inconsistentports

Name                 Interface              Inconsistency
-------------------- ---------------------- ------------------
VLAN0001             FastEthernet3/1        Port Type Inconsistent
VLAN0001             FastEthernet3/2        Port Type Inconsistent
VLAN1002             FastEthernet3/1        Port Type Inconsistent
VLAN1002             FastEthernet3/2        Port Type Inconsistent
VLAN1003             FastEthernet3/1        Port Type Inconsistent
VLAN1003             FastEthernet3/2        Port Type Inconsistent
VLAN1004             FastEthernet3/1        Port Type Inconsistent
VLAN1004             FastEthernet3/2        Port Type Inconsistent
VLAN1005             FastEthernet3/1        Port Type Inconsistent
VLAN1005             FastEthernet3/2        Port Type Inconsistent

Number of inconsistent ports (segments) in the system :10
```

Overview of Loop Guard

Loop guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link. When enabled globally, 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 blocking state, assuming there is a physical link error on this port. The port recovers from this state as soon as it receives a BPDU.

You can enable loop guard on a per-port basis. When you enable loop guard, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable loop guard, it is disabled for the specified ports. Disabling loop guard moves all loop-inconsistent ports to the listening state.

If you enable loop guard on a channel and the first link becomes unidirectional, loop guard blocks the entire channel until the affected port is removed from the channel. Figure 18-1 shows loop guard in a triangular switch configuration.

Figure 18-1 Triangular Switch Configuration with Loop Guard

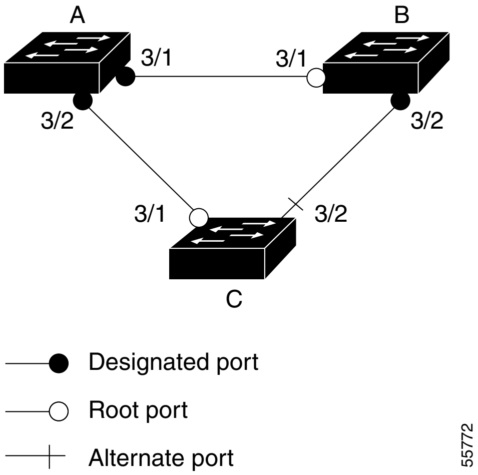


Figure 18-1 illustrates the following configuration:

- 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.

Follow these guidelines when using loop guard:

- Do not enable loop guard on PortFast-enabled or dynamic VLAN ports.
- Do not 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 does not work.
- Root guard forces a port to always be 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 is not able to detect it.

- Loop guard has no effect on a disabled spanning tree instance or a VLAN.

Enabling Loop Guard

You can enable loop guard globally or per port.

To enable loop guard globally on the switch, perform this task:

	Command	Purpose
Step 1	`Switch(config)# spanning-tree loopguard default`	Enables loop guard globally on the switch.

	Command	Purpose
Step 2	Switch(config)# **end**	Exits configuration mode.
Step 3	Switch# **show spanning tree interface 4/4 detail**	Verifies the configuration impact on a port.

This example shows how to enable loop guard globally:

```
Switch(config)# spanning-tree loopguard default
Switch(config)# Ctrl-Z
```

This example shows how to verify the previous configuration of port 4/4:

```
Switch# 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 an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# **interface** {*type slot/port*} | {**port-channel** *port_channel_number*}	Selects an interface to configure.
Step 2	Switch(config-if)# **spanning-tree guard loop**	Configures loop guard.
Step 3	Switch(config)# **end**	Exits configuration mode.
Step 4	Switch# **show spanning tree interface 4/4 detail**	Verifies the configuration impact on that port.

This example shows how to enable loop guard on port 4/4:

```
Switch(config)# interface fastEthernet 4/4
Switch(config-if)# spanning-tree guard loop
Switch(config-if)# ^Z
```

This example shows how to verify the configuration impact on port 4/4:

```
Switch# 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
Switch#
```

Overview of PortFast

Spanning Tree PortFast causes an interface configured as a Layer 2 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, rather than waiting for spanning tree to converge. Even if the interface receives a bridge protocol data unit (BPDU), spanning tree will not place the port into the blocking state but it will set the port's operating state to *non-port fast* even if the configured state remains *port fast* and starts participating in the Topology change.

Note Because the purpose of PortFast is to minimize the time that access ports must wait for spanning tree to converge, it is most effective when used on access ports. If you enable PortFast on a port connecting to another switch, you risk creating a spanning tree loop.

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 to force it to enter the forwarding state immediately, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`interface`** `{{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`*`}	{`**`port-channel`** *`port_channel_number`*`}`	Specifies an interface to configure.
Step 2	`Switch(config-if)# [`**`no`**`]` **`spanning-tree portfast`**	Enables PortFast on a Layer 2 access port connected to a single workstation or server. You can use the **no** keyword to disable PortFast.
Step 3	`Switch(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Switch#` **`show running interface`** `{{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`*`}	{`**`port-channel`** *`port_channel_number`*`}`	Verifies the configuration.

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

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

This example shows how to verify the configuration:

```
Switch# 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

Switch#
```

Overview of BPDU Guard

Spanning Tree BPDU guard shuts down PortFast-configured interfaces that receive BPDUs, rather than putting them into the spanning tree blocking state. In a valid configuration, PortFast-configured interfaces do not receive BPDUs. Reception of a BPDU by a PortFast-configured 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 interface back in service.

Note When the BPDU guard feature is enabled, spanning tree applies the BPDU guard feature to all PortFast-configured interfaces.

Enabling BPDU Guard

To enable BPDU guard to shut down PortFast-configured interfaces that receive BPDUs, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree portfast bpduguard`	Enables BPDU guard on all the switch’s PortFast-configured interfaces. You can use the **no** keyword to disable BPDU guard.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree summary totals`	Verifies the BPDU configuration.

This example shows how to enable BPDU guard:

```
Switch(config)# spanning-tree portfast bpduguard
Switch(config)# end
Switch#
```

This example shows how to verify the BPDU configuration:

```
Switch# show spanning-tree summary totals

Root bridge for: none.
PortFast BPDU Guard is enabled
Etherchannel misconfiguration guard is enabled
UplinkFast is disabled
BackboneFast is disabled
Default pathcost method used is short

Name                 Blocking Listening Learning Forwarding STP Active
-------------------- -------- --------- -------- ---------- ----------
            34 VLANs 0        0         0        36         36
Switch#
```

Overview of PortFast BPDU Filtering

Cisco IOS Release 12.2(25)EW and later support PortFast BPDU filtering, which 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 Explicitly configuring PortFast BPDU filtering on a port that is not connected to a host can result in bridging loops, because the port ignores any BPDU it receives and goes to the forwarding state.

When you enable PortFast BPDU filtering globally and set the port configuration as the default for PortFast BPDU filtering (see the "Enabling BackboneFast" section on page 18-15), PortFast enables or disables PortFast BPDU filtering.

If the port configuration is not set to default, then the PortFast configuration does not affect PortFast BPDU filtering. Table 18-1 lists all the possible PortFast BPDU filtering combinations. PortFast BPDU filtering allows access ports to move directly to the forwarding state as soon as the end hosts are connected.

Table 18-1 PortFast BPDU Filtering Port Configurations

Per-Port Configuration	Global Configuration	PortFast State	PortFast BPDU Filtering State
Default	Enable	Enable	Enable[1]
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

[1] The port transmits at least 10 BPDUs. If this port receives any BPDUs, then PortFast and PortFast BPDU filtering are disabled.

Enabling PortFast BPDU Filtering

To enable PortFast BPDU filtering globally, perform this task:

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

This example shows how to enable PortFast BPDU filtering on a port:

```
Switch(config)# spanning-tree portfast bpdufilter default
Switch(config)# Ctrl-Z
```

This example shows how to verify the BPDU configuration in PVST+ mode:

```
Switch# 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

Switch#
```

Note For PVST+ information, see Chapter 15, "Configuring Multiple Spanning Trees."

To enable PortFast BPDU filtering, perform this task:

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

This example shows how to enable PortFast BPDU filtering on port 4/4:

```
Switch(config)# interface fastethernet 4/4
Switch(config-if)# spanning-tree bpdufilter enable
Switch(config-if)# ^Z
```

This example shows how to verify that PortFast BPDU filtering is enabled:

```
Switch# show spanning-tree interface fastethernet 4/4

Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0010         Desg FWD 1000      160.196  Edge P2p
```

This example shows more detail on the port:

```
Switch# 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
Switch#
```

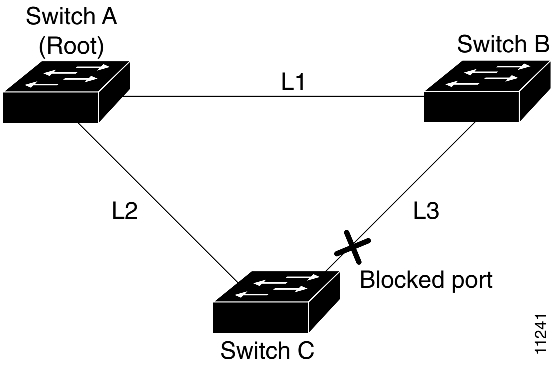
Overview of UplinkFast

Note UplinkFast is most useful in wiring-closet switches. This feature might not be useful for other types of applications.

Spanning Tree UplinkFast provides fast convergence after a direct link failure and uses uplink groups to achieve load balancing between redundant Layer 2 links. Convergence is the speed and ability of a group of internetworking devices running a specific routing protocol to agree on the topology of an internetwork after a change in that topology. An uplink group is a set of Layer 2 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.

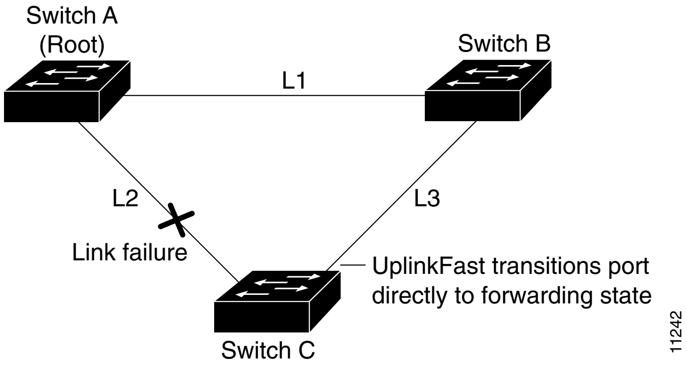
Figure 18-2 shows an example of a topology with no link failures. Switch A, the root switch, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 18-2 UplinkFast Before Direct Link Failure



If Switch C detects a link failure on the currently active link L2 on the root port (a direct link failure), UplinkFast unblocks the blocked port on Switch C and transitions it to the forwarding state without going through the listening and learning states, as shown in Figure 18-3. This switchover takes approximately one to five seconds.

Figure 18-3 UplinkFast After Direct Link Failure



Enabling UplinkFast

UplinkFast increases the bridge priority to 49,152 and adds 3000 to the spanning tree port cost of all interfaces on the switch, making it unlikely that the switch becomes the root switch. The *max_update_rate* value represents the number of multicast packets transmitted per second (the default is 150 packets per second [pps]).

UplinkFast cannot be enabled on VLANs that have been configured for bridge priority. To enable UplinkFast on a VLAN with bridge priority configured, restore the bridge priority on the VLAN to the default value by entering a **no spanning-tree vlan** *vlan_ID* **priority** command in global configuration mode.

Note When you enable UplinkFast, it affects all VLANs on the switch. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree uplinkfast [max-update-rate max_update_rate]`	Enables UplinkFast. You can use the **no** keyword to disable UplinkFast and restore the default rate, use the command
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree vlan vlan_ID`	Verifies that UplinkFast is enabled on that VLAN.

This example shows how to enable UplinkFast with a maximum update rate of 400 pps:

```
Switch(config)# spanning-tree uplinkfast max-update-rate 400
Switch(config)# exit
Switch#
```

This example shows how to verify which VLANS have UplinkFast enabled:

```
Switch# show spanning-tree uplinkfast
UplinkFast is enabled

Station update rate set to 150 packets/sec.

UplinkFast statistics
-----------------------
Number of transitions via uplinkFast (all VLANs)            :14
Number of proxy multicast addresses transmitted (all VLANs) :5308

Name                 Interface List
-------------------- ------------------------------------
VLAN1                Fa6/9(fwd), Gi5/7
VLAN2                Gi5/7(fwd)
VLAN3                Gi5/7(fwd)
VLAN4
VLAN5
VLAN6
VLAN7
VLAN8
VLAN10
VLAN15
VLAN1002             Gi5/7(fwd)
VLAN1003             Gi5/7(fwd)
VLAN1004             Gi5/7(fwd)
VLAN1005             Gi5/7(fwd)
Switch#
```

Overview of BackboneFast

BackboneFast is a complementary technology to UplinkFast. Whereas UplinkFast is designed to quickly respond to failures on links directly connected to leaf-node switches, it does not help with indirect failures in the backbone core. BackboneFast optimizes based on the Max Age setting. It allows the default convergence time for indirect failures to be reduced from 50 seconds to 30 seconds. However, it never eliminates forward delays and offers no assistance for direct failures.

Note BackboneFast should be enabled on every switch in your network.

Sometimes a switch receives a BPDU from a designated switch that identifies the root bridge and the designated bridge as the same switch. Because this shouldn't happen, the BPDU is considered inferior.

BPDUs are considered inferior when a link from the designated switch has lost its link to the root bridge. The designated switch transmits the BPDUs with the information that it is now the root bridge as well as the designated bridge. The receiving switch ignores the inferior BPDU for the time defined by the Max Age setting.

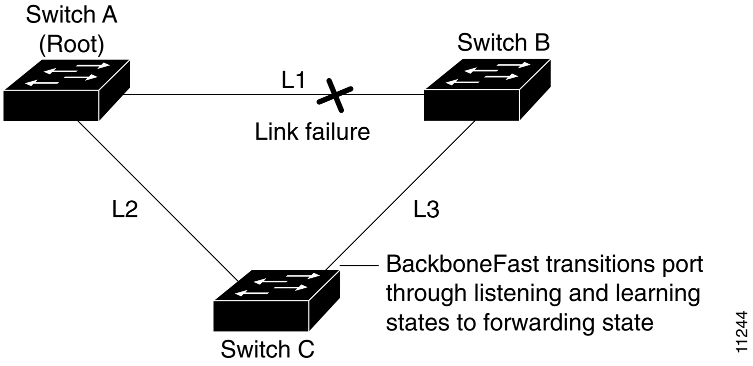
After receiving inferior BPDUs, the receiving switch tries to determine if there is an alternate path to the root bridge.

- If the port that the inferior BPDUs are received on is already in blocking mode, then the root port and other blocked ports on the switch become alternate paths to the root bridge.
- If the inferior BPDUs are received on a root port, then all presently blocking ports become the alternate paths to the root bridge. Also, if the inferior BPDUs are received on a root port and there are no other blocking ports on the switch, the receiving switch assumes that the link to the root bridge is down and the time defined by the Max Age setting expires, which turns the switch into the root switch.

If the switch finds an alternate path to the root bridge, it uses this new alternate path. This new path, and any other alternate paths, are used to send a Root Link Query (RLQ) BPDU. When BackboneFast is enabled, the RLQ BPDUs are sent out as soon as an inferior BPDU is received. This process can enable faster convergence in the event of a backbone link failure.

Figure 18-4 shows an example of a topology with no link failures. Switch A, the root switch, connects directly to Switch B over link L1 and to Switch C over link L2. In this example, because switch B has a lower priority than A but higher than C, switch B becomes the designated bridge for L3. Consequently, the Layer 2 interface on Switch C that connects directly to Switch B must be in the blocking state.

Figure 18-4 BackboneFast Before Indirect Link Failure


Switch A
(Root)
Switch B
L1
Link failure
L2
L3
BackboneFast transitions port through listening and learning states to forwarding state
Switch C
11244

Next, assume that L1 fails. Switch A and Switch B, the switches directly connected to this segment, instantly know that the link is down. The blocking interface on Switch C must enter the forwarding state for the network to recover by itself. However, because L1 is not directly connected to Switch C, Switch C does not start sending any BPDUs on L3 under the normal rules of STP until the time defined by the Max Age setting has expired.

In an STP environment without BackboneFast, if L1 should fail, 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 switch over L1, Switch B detects the failure and elects itself the root. Then Switch B begins sending configuration BDPUs to Switch C, listing itself as the root.

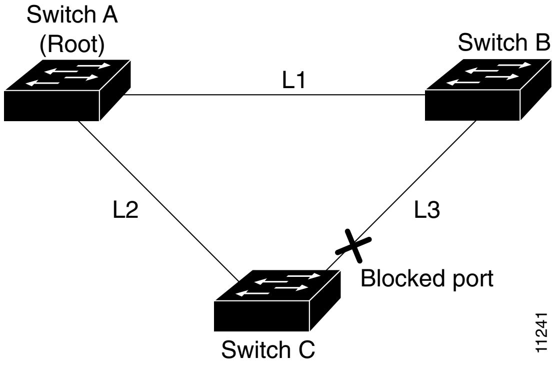
Here is what happens additionally when you use BackboneFast to eliminate the time defined by the Max Age setting (20-second) delay:

1. When Switch C receives the inferior configuration BPDUs from Switch B, Switch C infers that an indirect failure has occurred.
2. Switch C then sends out an RLQ.
3. Switch A receives the RLQ. Because Switch A is the root bridge, it replies with an RLQ response, listing itself as the root bridge.
4. When Switch C receives the RLQ response on its existing root port, it knows that it still has a stable connection to the root bridge. Because Switch C originated the RLQ request, it does not need to forward the RLQ response on to other switches.
5. BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the time defined by the Max Age setting for the port to expire.
6. BackboneFast transitions the Layer 2 interface on Switch C to the forwarding state, providing a path from Switch B to Switch A.

This switchover takes approximately 30 seconds, twice the Forward Delay time if the default forward delay time of 15 seconds is set.

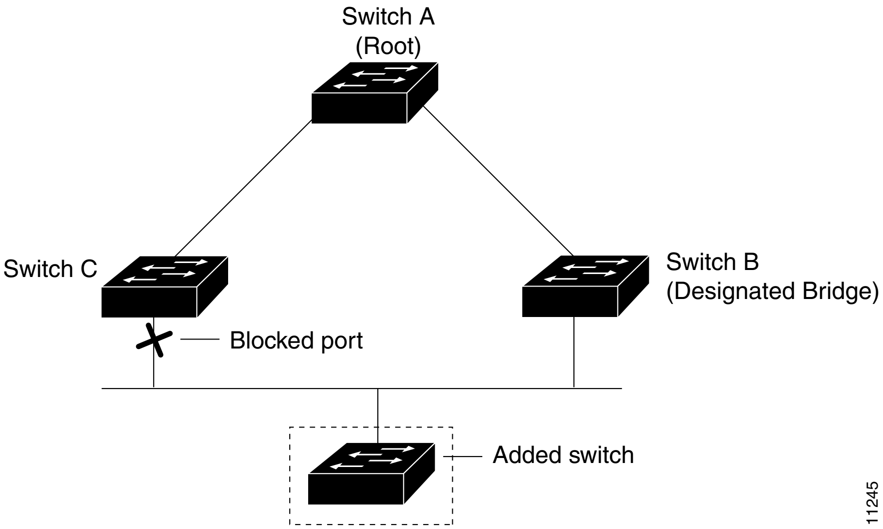
Figure 18-5 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 18-5 BackboneFast after Indirect Link Failure



If a new switch is introduced into a shared-medium topology as shown in Figure 18-6, BackboneFast is not activated, because the inferior BPDUs did not come from the recognized designated bridge (Switch B). The new switch begins sending inferior BPDUs that say it is the root switch. However, the other switches ignore these inferior BPDUs, and the new switch learns that Switch B is the designated bridge to Switch A, the root switch.

Figure 18-6 Adding a Switch in a Shared-Medium Topology



Enabling BackboneFast

Note For BackboneFast to work, you must enable it on all switches in the network. BackboneFast is supported for use with third-party switches but it is not supported on Token Ring VLANs.

To enable BackboneFast, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] spanning-tree backbonefast`	Enables BackboneFast. You can use the **no** keyword to disable BackboneFast.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show spanning-tree backbonefast`	Verifies that BackboneFast is enabled.

This example shows how to enable BackboneFast:

```
Switch(config)# spanning-tree backbonefast
Switch(config)# end
Switch#
```

This example shows how to verify that BackboneFast is enabled:

```
Switch# 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
Switch#
```

This example shows how to display a summary of port states:

```
Switch#show spanning-tree summary
Root bridge for:VLAN0001, VLAN1002-VLAN1005
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
EtherChannel misconfiguration guard is enabled
UplinkFast           is enabled
BackboneFast         is enabled
Pathcost method used is short

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
VLAN0001                  0        0         0        3          3
VLAN1002                  0        0         0        2          2
VLAN1003                  0        0         0        2          2
VLAN1004                  0        0         0        2          2
VLAN1005                  0        0         0        2          2
---------------------- -------- --------- -------- ---------- ----------
5 vlans                   0        0         0        11         11

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
Switch#
```

This example shows how to display the total lines of the spanning tree state section:

```
Switch#show spanning-tree summary totals
Root bridge for:VLAN0001, VLAN1002-VLAN1005
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
EtherChannel misconfiguration guard is enabled
```

```
UplinkFast           is enabled
BackboneFast         is enabled
Pathcost method used is short

Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
5 vlans                   0        0         0        11         11

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
Switch#
```

CHAPTER 19

Configuring EtherChannel

This chapter describes how to use the command-line interface (CLI) to configure EtherChannel on the Catalyst 4500 series switch Layer 2 or Layer 3 interfaces. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- EtherChannel Overview, page 19-1
- EtherChannel Configuration Guidelines and Restrictions, page 19-5
- Configuring EtherChannel, page 19-6

Note The commands in the following sections can be used on all Ethernet interfaces on a Catalyst 4500 series switch, including the uplink ports on the supervisor engine.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

EtherChannel Overview

EtherChannel bundles individual Ethernet links into a single logical link that provides bandwidth up to 1600 Mbps (Fast EtherChannel full duplex),16 Gbps (Gigabit EtherChannel), or 40 Gbps (10 Gigabit Etherchannel) between a Catalyst 4500 series switch and another switch or host.

A Catalyst 4500 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces across modules in a Catalyst 4500 series switch. All interfaces in each EtherChannel must be the same speed and must be configured as either Layer 2 or Layer 3 interfaces.

Note The network device to which a Catalyst 4500 series switch is connected may impose its own limits on the number of interfaces 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 the segment fails, an SNMP trap is sent, identifying 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.

Note The port channel link failure switchover for the Catalyst 4500 series switch was measured at 50 ms, giving you SONET-like link failure switchover time.

These subsections describe how EtherChannel works:

- Port-Channel Interfaces, page 19-2
- How EtherChannels Are Configured, page 19-2
- Load Balancing, page 19-4

Port-Channel Interfaces

Each EtherChannel has a numbered port-channel interface. A configuration applied to the port-channel interface affects all physical interfaces assigned to that interface.

Note QoS does not propagate to members. The defaults, QoS cos = 0 and QoS dscp = 0, apply on the portchannel. Input or output policies applied on individual interfaces are ignored.

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 physical interfaces affects only the interface where you apply the configuration. To change the parameters of all ports in an EtherChannel, apply configuration commands to the port-channel interface (such commands can be STP commands or commands to configure a Layer 2 EtherChannel as a trunk).

How EtherChannels Are Configured

These subsections describe how EtherChannels are configured:

- EtherChannel Configuration Overview, page 19-2
- Manual EtherChannel Configuration, page 19-3
- PAgP EtherChannel Configuration, page 19-3
- IEEE 802.3ad LACP EtherChannel Configuration, page 19-3

EtherChannel Configuration Overview

You can configure EtherChannels manually or you can use the Port Aggregation Control Protocol (PAgP) or, with Cisco IOS Release 12.2(25)EWA and later, 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. Ports configured to use PAgP cannot form EtherChannels with ports configured to use LACP and vice versa.

Table 19-1 lists the user-configurable EtherChannel modes.

Table 19-1 EtherChannel Modes

Mode	Description
on	Mode that forces the LAN port to channel unconditionally. In the **on** mode, a usable EtherChannel exists only when a LAN port group in the **on** mode is connected to another LAN port group in the **on** mode. Because ports configured in the **on** mode do not negotiate, there is no negotiation traffic between the ports.
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.
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.
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.

Manual EtherChannel Configuration

Manually configured EtherChannel ports do not exchange EtherChannel protocol packets. A manually configured EtherChannel forms only when you configure all ports in the EtherChannel compatibly.

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 initiates negotiation.

IEEE 802.3ad LACP EtherChannel Configuration

Cisco IOS Release 12.2(25)EWA and later releases support IEEE 802.3ad LACP EtherChannels. 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 initiates negotiation.

LACP uses the following parameters:

- LACP system priority—You may configure an LACP system priority on each switch running LACP. The system priority can be configured automatically or through the CLI. See the “Configuring the LACP System Priority and System ID” section on page 19-12. 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 Layer 2 EtherChannels” section on page 19-9. LACP uses the port priority with the port number to form the port identifier.
- 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

LACP tries to configure the maximum number of compatible ports in an EtherChannel up to the maximum allowed by the hardware (eight ports). If a port cannot be actively included in a channel, it is not included automatically if a channelled port fails.

Note Standby and “sub-channeling” are not supported in LACP and PAgP.

Load Balancing

EtherChannel can balance the traffic load across the links in the channel by reducing part of the binary pattern formed from the addresses or ports in the frame to a numerical value that selects one of the links in the channel. To balance the load, EtherChannel uses MAC addresses, IP addresses, or Layer 4 port numbers, and either the message source or message destination, or both.

Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination MAC address always chooses the same link in the channel; using source addresses or IP addresses might result in better load balancing.

Note Load balancing can only be configured globally. As a result, all channels (manually configured, PagP, or LACP) use the same load balancing method.

For additional information on load balancing, see the "Configuring EtherChannel Load Balancing" section on page 19-13.

EtherChannel Configuration Guidelines and Restrictions

If improperly configured, some EtherChannel interfaces are disabled automatically to avoid network loops and other problems. Follow these guidelines and restrictions to avoid configuration problems:

- All Ethernet interfaces on all modules support EtherChannel (maximum of eight interfaces) with no requirement that interfaces be physically contiguous or on the same module.
- Configure all interfaces in an EtherChannel to operate at the same speed and duplex mode.
- Enable all interfaces in an EtherChannel. Putting down an interface in an Ether Channel is treated as a link failure, and its traffic is transferred to one of the remaining interfaces in the EtherChannel.
- An EtherChannel does not form if one of the interfaces 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 physical interfaces in the channel.
- For Layer 2 EtherChannels:
  - Assign all interfaces in the EtherChannel to the same VLAN, or configure them as trunks.
  - If you configure an EtherChannel from trunk interfaces, verify that the trunking mode and the native VLAN is the same on all the trunks. Interfaces in an EtherChannel with different trunk modes or different native VLANs can have unexpected results.
  - An EtherChannel supports the same allowed range of VLANs on all the interfaces in a trunking Layer 2 EtherChannel. If the allowed ranges differ for selected interface differ, they do not form an EtherChannel.
  - Interfaces with different Spanning Tree Protocol (STP) port path costs can form an EtherChannel as long they are otherwise compatibly configured. Setting different STP port path costs does not, by itself, make interfaces incompatible for the formation of an EtherChannel.
- After you configure an EtherChannel, any configuration that you apply to the port-channel interface affects the EtherChannel; any configuration that you apply to the physical interfaces affects only the interface you configure.

  Storm Control is an exception to this rule. For example, you cannot configure Storm Control on some of the members of an EtherChannel; Storm Control must be configured on all or none of the ports. If you configure Storm Control on only some of the ports, those ports will be dropped from the EtherChannel interface (put in suspended state). Therefore, you should configure Storm Control at the port-channel interface level, and not at the physical interface level.

- A physical interface with port security enabled can join a Layer 2 EtherChannel only if port security is also enabled on the EtherChannel; otherwise the command is rejected by the CLI.
- You cannot configure a 802.1X port in an EtherChannel.

Configuring EtherChannel

These sections describe how to configure EtherChannel:

- Configuring Layer 3 EtherChannels, page 19-6
- Configuring Layer 2 EtherChannels, page 19-9
- Configuring the LACP System Priority and System ID, page 19-12
- Configuring EtherChannel Load Balancing, page 19-13
- Removing an Interface from an EtherChannel, page 19-14
- Removing an EtherChannel, page 19-14

Note Ensure that the interfaces are configured correctly. (See the "EtherChannel Configuration Guidelines and Restrictions" section on page 19-5.)

Configuring Layer 3 EtherChannels

To configure Layer 3 EtherChannels, create the port-channel logical interface and then put the Ethernet interfaces into the portchannel.

These sections describe Layer 3 EtherChannel configuration:

- Creating Port-Channel Logical Interfaces, page 19-6
- Configuring Physical Interfaces as Layer 3 EtherChannels, page 19-7

Creating Port-Channel Logical Interfaces

Note To move an IP address from a physical interface to an EtherChannel, you must delete the IP address from the physical interface before configuring it on the port-channel interface.

To create a port-channel interface for a Layer 3 EtherChannel, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface port-channel port_channel_number`	Creates the port-channel interface. The value for *port_channel_number* can range from 1 to 64.
Step 2	`Switch(config-if)# ip address ip_address mask`	Assigns an IP address and subnet mask to the EtherChannel.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show running-config interface port-channel port_channel_number`	Verifies the configuration.

This example shows how to create port-channel interface 1:

```
Switch# configure terminal
Switch(config)# interface port-channel 1
Switch(config-if)# ip address 172.32.52.10 255.255.255.0
Switch(config-if)# end
```

This example shows how to verify the configuration of port-channel interface 1:

```
Switch# 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

Switch#
```

Configuring Physical Interfaces as Layer 3 EtherChannels

To configure physical interfaces as Layer 3 EtherChannels, perform this task for each interface:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet	gigabitethernet	tengigabitethernet} slot/port`	Selects a physical interface to configure.
Step 2	`Switch(config-if)# no switchport`	Makes this a Layer 3 routed port.
Step 3	`Switch(config-if)# no ip address`	Ensures that no IP address is assigned to the physical interface.
Step 4	`Switch(config-if)# channel-group port_channel_number mode {active	on	auto	passive	desirable}`	Configures the interface in a portchannel and specifies the PAgP or LACP mode. If you use PAgP, enter the keywords **auto** or **desirable**. If you use LACP, enter the keywords **active** or **passive**.

	Command	Purpose
Step 5	`Switch(config-if)# end`	Exits configuration mode.

Step 6	`Switch# show running-config interface port-channel port_channel_number`
	`Switch# show running-config interface {fastethernet | gigabitethernet | tengigabitethernet} slot/port`
	`Switch# show interfaces {fastethernet | gigabitethernet | tengigabitethernet} slot/port etherchannel`
	`Switch# show etherchannel 1 port-channel`
		Verifies the configuration.

This example shows how to configure Fast Ethernet interfaces 5/4 and 5/5 into port-channel 1 with PAgP mode **desirable**:

```
Switch# configure terminal
Switch(config)# interface range fastethernet 5/4 - 5 (Note: Space is mandatory.)
Switch(config-if)# no switchport
Switch(config-if)# no ip address
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# end
```

Note See the "Configuring a Range of Interfaces" section on page 6-4 for information about the **range** keyword.

The following two examples show how to verify the configuration of Fast Ethernet interface 5/4:

```
Switch# show running-config interface fastethernet 5/4
Building configuration...

Current configuration:
!
interface FastEthernet5/4
 no ip address
 no switchport
 no ip directed-broadcast
 channel-group 1 mode desirable
end

Switch# show interfaces fastethernet 5/4 etherchannel
Port state    = EC-Enbld Up In-Bndl Usr-Config
Channel group = 1           Mode = Desirable     Gcchange = 0
Port-channel  = Po1         GC   = 0x00010001    Pseudo-port-channel = Po1
Port indx     = 0           Load = 0x55

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.
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
Fa5/4     SC    U6/S7           30s      1        128        Any      55
```

```
Partner's information:

          Partner               Partner          Partner         Partner Group
Port      Name                  Device ID        Port       Age  Flags   Cap.
Fa5/4     JAB031301             0050.0f10.230c   2/45         1s SAC     2D

Age of the port in the current state: 00h:54m:52s

Switch#
```

This example shows how to verify the configuration of port-channel interface 1 after the interfaces have been configured:

```
Switch# show etherchannel 1 port-channel

                Channel-group listing:
                ----------------------
Group: 1
------------

                Port-channels in the group:
                ----------------------
Port-channel: Po1
------------

Age of the Port-channel   = 01h:56m:20s
Logical slot/port   = 10/1           Number of ports = 2
GC                  = 0x00010001      HotStandBy port = null
Port state          = Port-channel L3-Ag Ag-Inuse

Ports in the Port-channel:

Index   Load   Port
-------------------
  1     00      Fa5/6
  0     00      Fa5/7

Time since last port bundled:    00h:23m:33s    Fa5/6

Switch#
```

Configuring Layer 2 EtherChannels

To configure Layer 2 EtherChannels, configure the Ethernet interfaces with the **channel-group** command. This creates the port-channel logical interface.

Note Cisco IOS software creates port-channel interfaces for Layer 2 EtherChannels when you configure Layer 2 Ethernet interfaces with the **channel-group** command.

To configure Layer 2 Ethernet interfaces as Layer 2 EtherChannels, perform this task for each interface:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet	gigabitethernet	tengigabitethernet} slot/port`	Selects a physical interface to configure.
Step 2	`Switch(config-if)# channel-group port_channel_number mode {active	on	auto	passive	desirable}`	Configures the interface in a portchannel and specifies the PAgP or LACP mode. If you use PAgP, enter the keywords **auto** or **desirable**. If you use LACP, enter the keywords **active** or **passive**.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show running-config interface {fastethernet	gigabitethernet} slot/port` `Switch# show interface {fastethernet	gigabitethernet	tengigabitethernet} slot/port etherchannel`	Verifies the configuration.

This example shows how to configure Fast Ethernet interfaces 5/6 and 5/7 into port-channel 2 with PAgP mode **desirable**:

```
Switch# configure terminal
Switch(config)# interface range fastethernet 5/6 - 7 (Note: Space is mandatory.)
Switch(config-if-range)# channel-group 2 mode desirable
Switch(config-if-range)# end
Switch# end
```

Note See the “Configuring a Range of Interfaces” section on page 6-4 for information about the **range** keyword.

This example shows how to verify the configuration of port-channel interface 2:

```
Switch# show running-config interface port-channel 2
Building configuration...

Current configuration:
!
interface Port-channel2
 switchport access vlan 10
 switchport mode access
end

Switch#
```

The following two examples show how to verify the configuration of Fast Ethernet interface 5/6:

```
Switch# show running-config interface fastethernet 5/6
Building configuration...

Current configuration:
!
interface FastEthernet5/6
 switchport access vlan 10
 switchport mode access
 channel-group 2 mode desirable
end
```

```
Switch# show interfaces fastethernet 5/6 etherchannel
Port state    = EC-Enbld Up In-Bndl Usr-Config
Channel group = 1           Mode = Desirable     Gcchange = 0
Port-channel  = Po1         GC   = 0x00010001
Port indx     = 0           Load = 0x55

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
Fa5/6     SC    U6/S7           30s      1        128        Any      56

Partner's information:

          Partner              Partner          Partner         Partner Group
Port      Name                 Device ID        Port       Age  Flags   Cap.
Fa5/6     JAB031301            0050.0f10.230c   2/47        18s SAC     2F

Age of the port in the current state: 00h:10m:57s
```

This example shows how to verify the configuration of port-channel interface 2 after the interfaces have been configured:

```
Switch# show etherchannel 2 port-channel
                Port-channels in the group:
                ----------------------

Port-channel: Po2
------------

Age of the Port-channel   = 00h:23m:33s
Logical slot/port   = 10/2           Number of ports in agport = 2
GC                  = 0x00020001      HotStandBy port = null
Port state          = Port-channel Ag-Inuse

Ports in the Port-channel:

Index   Load   Port
-------------------
  1     00      Fa5/6
  0     00      Fa5/7

Time since last port bundled:    00h:23m:33s    Fa5/6

Switch#
```

Configuring the LACP System Priority and System ID

The LACP system ID is the LACP system priority value combined with the MAC address of the switch.

To configure the LACP system priority and system ID, perform this task:

	Command	Purpose
Step 1	`Switch(config)# `**`lacp system-priority`** *`priority_value`*	(Optional for LACP) Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.
	`Switch(config)# `**`no system port-priority`**	Reverts to the default.
Step 2	`Switch(config)# `**`end`**	Exits configuration mode.
Step 3	`Switch# `**`show lacp sys-id`**	Verifies the configuration.

This example shows how to configure the LACP system priority:

```
Switch# configure terminal
Switch(config)# lacp system-priority 23456
Switch(config)# end
Switch# show module

Mod  Ports Card Type                              Model             Serial No.
----+-----+--------------------------------------+-----------------+-----------
 1      2  1000BaseX (GBIC) Supervisor(active)    WS-X4014          JAB063808YZ
 2     48  10/100BaseTX (RJ45)                    WS-X4148-RJ       JAB0447072W
 3     48  10/100BaseTX (RJ45)V                   WS-X4148-RJ45V    JAE061704J6
 4     48  10/100BaseTX (RJ45)V                   WS-X4148-RJ45V    JAE061704ML

 M MAC addresses                    Hw  Fw           Sw               Status
--+--------------------------------+---+------------+----------------+---------
 1 0005.9a39.7a80 to 0005.9a39.7a81 2.1 12.1(12r)EW  12.1(13)EW(0.26) Ok
 2 0002.fd80.f530 to 0002.fd80.f55f 0.1                               Ok
 3 0009.7c45.67c0 to 0009.7c45.67ef 1.6                               Ok
 4 0009.7c45.4a80 to 0009.7c45.4aaf 1.6                               Ok
```

This example shows how to verify the configuration:

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

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

Configuring EtherChannel Load Balancing

Note Load balancing can only be configured globally. As a result, all channels (manually configured, PagP, or LACP) use the same load balancing method.

To configure EtherChannel load balancing, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] 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. Use the **no** keyword to return EtherChannel load balancing to the default configuration.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show etherchannel load-balance`	Verifies the configuration.

The load-balancing keywords are:

- **src-mac**—Source MAC addresses
- **dst-mac**—Destination MAC addresses
- **src-dst-mac**—Source and destination MAC addresses
- **src-ip**—Source IP addresses
- **dst-ip**—Destination IP addresses
- **src-dst-ip**—Source and destination IP addresses (Default)
- **src-port**—Source Layer 4 port
- **dst-port**—Destination Layer 4 port
- **src-dst-port**—Source and destination Layer 4 port

This example shows how to configure EtherChannel to use source and destination IP addresses:

```
Switch# configure terminal
Switch(config)# port-channel load-balance src-dst-ip
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address
Switch#
```

Removing an Interface from an EtherChannel

To remove an Ethernet interface from an EtherChannel, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`interface`** `{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`*	Selects a physical interface to configure.
Step 2	`Switch(config-if)#` **`no channel-group`**	Removes the interface from the port-channel interface.
Step 3	`Switch(config-if)#` **`end`**	Exits configuration mode.

Step 4	`Switch#` **`show running-config interface`** `{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`*
	`Switch#` **`show interface`** `{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`* **`etherchannel`**
		Verifies the configuration.

This example shows how to remove Fast Ethernet interfaces 5/4 and 5/5 from port-channel 1:

```
Switch# configure terminal
Switch(config)# interface range fastethernet 5/4 - 5 (Note: Space is mandatory.)
Switch(config-if)# no channel-group 1
Switch(config-if)# end
```

Removing an EtherChannel

If you remove an EtherChannel, the member ports are shut down and removed from the channel group.

Note If you want to change an EtherChannel from Layer 2 to Layer 3, or Layer 3 to Layer 2, you must remove the EtherChannel and recreate it in the desired configuration.

To remove an EtherChannel, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`no interface port-channel`** *`port_channel_number`*	Removes the port-channel interface.
Step 2	`Switch(config)#` **`end`**	Exits configuration mode.
Step 3	`Switch#` **`show etherchannel summary`**	Verifies the configuration.

This example shows how to remove port-channel 1:

```
Switch# configure terminal
Switch(config)# no interface port-channel 1
Switch(config)# end
```

CHAPTER 20

Configuring IGMP Snooping and Filtering

This chapter describes how to configure Internet Group Management Protocol (IGMP) snooping on the Catalyst 4500 series switch. It provides guidelines, procedures, and configuration examples.

This chapter consists of the following major sections:

- Overview of IGMP Snooping, page 20-1
- Configuring IGMP Snooping, page 20-4
- Displaying IGMP Snooping Information, page 20-13
- Configuring IGMP Filtering, page 20-17
- Displaying IGMP Filtering Configuration, page 20-21

Note To support Cisco Group Management Protocol (CGMP) client devices, configure the switch as a CGMP server. For more information, see the chapters "IP Multicast" and "Configuring IP Multicast Routing" in the *Cisco IOS IP and IP Routing Configuration Guide*, Cisco IOS Release 12.2 at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ip_c/ipcprt3/1cdmulti.htm

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of IGMP Snooping

This section includes the following subsections:

- Immediate-Leave Processing, page 20-3
- IGMP Configurable-Leave Timer, page 20-3
- Explicit Host Tracking, page 20-4

Note Quality of service does not apply to IGMP packets.

IGMP snooping allows a switch to snoop or capture information from IGMP packets transmitted between hosts and a router. Based on this information, a switch adds or deletes multicast addresses from its address table, thereby enabling (or disabling) multicast traffic from flowing to individual host ports. IGMP snooping supports all versions of IGMP: IGMPv1, IGMPv2, and IGMPv3.

In contrast to IGMPv1 and IGMPv2, IGMPv3 snooping provides immediate-leave processing by default. It provides Explicit Host Tracking (EHT) and allows network administrators to deploy SSM functionality on Layer 2 devices that truly support IGMPv3. (See the "Explicit Host Tracking" section on page 20-4.)

In subnets where IGMP is configured, IGMP snooping manages multicast traffic at Layer 2. You can configure interfaces to dynamically forward multicast traffic only to those interfaces that are interested in receiving it by using the **switchport** keyword.

IGMP snooping restricts traffic in MAC multicast groups 0100.5e00.0001 to 01-00-5e-ff-ff-ff. IGMP snooping does not restrict Layer 2 multicast packets generated by routing protocols.

Note For more information on IP multicast and IGMP, refer to RFC 1112, RFC 2236, RFC 3376 (for IGMPv3).

IGMP (configured on a router) periodically sends out IGMP general queries. A host responds to these queries with IGMP membership reports for groups that it is interested in. When IGMP snooping is enabled, the switch creates one entry per VLAN in the Layer 2 forwarding table for each Layer 2 multicast group from which it receives an IGMP join request. All hosts interested in this multicast traffic send IGMP membership reports and are added to the forwarding table entry.

Layer 2 multicast groups learned through IGMP snooping are dynamic. However, you can statically configure Layer 2 multicast groups using the **ip igmp snooping static** command. If you specify group membership statically, your setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can contain both user-defined and IGMP snooping settings.

Groups with IP addresses in the range 224.0.0.0 to 224.0.0.255, which map to the multicast MAC address range 0100.5E00.0001 to 0100.5E00.00FF, are reserved for routing control packets. These groups are flooded to all forwarding ports of the VLAN with the exception of 224.0.0.22, which is used for IGMPv3 membership reports.

Note If a VLAN experiences a spanning-tree topology change, IP multicast traffic floods on all VLAN ports where PortFast is not enabled, as well as on ports with the **no igmp snooping tcn flood** command configured for a period of TCN query count.

For a Layer 2 IGMPv2 host interface to join an IP multicast group, a host sends an IGMP membership report for the IP multicast group. For a host to leave a multicast group, it can either ignore the periodic IGMP general queries or it can send an IGMP leave message. When the switch receives an IGMP leave message from a host, it sends out an IGMP group-specific query to determine whether any devices connected to that interface are interested in traffic for the specific multicast group. The switch then updates the table entry for that Layer 2 multicast group so that only those hosts interested in receiving multicast traffic for the group are listed.

In contrast, IGMPv3 hosts send IGMPv3 membership reports (with the **allow** group record mode) to join a specific multicast group. When IGMPv3 hosts send membership reports (with the **block** group record) to reject traffic from all sources in the previous source list, the last host on the port is removed by immediate-leave if EHT is enabled.

Immediate-Leave Processing

IGMP snooping immediate-leave processing allows the switch to remove an interface from the forwarding-table entry without first sending out IGMP group-specific queries to the interface. The VLAN interface is pruned from the multicast tree for the multicast group specified in the original IGMP leave message. Immediate-leave processing ensures optimal bandwidth management for all hosts on a switched network, even when multiple multicast groups are being used simultaneously.

When a switch with IGMP snooping enabled receives an IGMPv2 or IGMPv3 leave message, it sends an IGMP group-specific query from the interface where the leave message was received to determine when there are other hosts attached to that interface that are interested in joining the MAC multicast group. If the switch does not receive an IGMP join message within the query response interval, the interface is removed from the port list of the (MAC-group, VLAN) entry in the Layer 2 forwarding table.

Note By default all IGMP joins are forwarded to all multicast router ports.

With immediate-leave processing enabled on the VLAN, an interface can be removed immediately from the port list of the Layer 2 entry when the IGMP leave message is received, unless a multicast router was learned on the port.

Note When using IGMPv2 snooping, use immediate-leave processing only on VLANs where just one host is connected to each interface. If immediate-leave processing is enabled on VLANs where multiple hosts are connected to an interface, some hosts might be dropped inadvertently. When using IGMPv3, immediate-leave processing is enabled by default, and due to Explicit Host Tracking (see below), the switch can detect when a port has single or multiple hosts maintained by the switch for IGMPv3 hosts. As a result, the switch can perform immediate-leave processing when it detects a single host behind a given port.

Note IGMPv3 is interoperable with older versions of IGMP.

Use the **show ip igmp snooping querier vlan** command to display the IGMP version on a particular VLAN.

Use the **show ip igmp snooping vlan** command to display whether or not the switch supports IGMPv3 snooping.

Use the **ip igmp snooping immediate-leave** command to enable immediate-leave for IGMPv2.

Note Immediate-leave processing is enabled by default for IGMPv3.

IGMP Configurable-Leave Timer

Immediate-leave processing cannot be used on VLANs where multiple hosts may be connected to a single interface. To reduce leave latency in such a scenario, IGMPv3 provides a configurable leave timer.

In Cisco IOS Release 12.2(25)SG and earlier, the IGMP snooping leave time was based on query response time. If membership reports were not received by the switch before the query response time of the query expired, a port was removed from the multicast group membership.

In Cisco IOS Release 12.2(31)SG and later, you can configure the length of time that the switch waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group. The IGMP leave response time can be configured from 100 to 5000 milliseconds. The timer can be set either globally or per VLAN. The VLAN configuration of the leave time overrides the global configuration.

For configuration steps, see the "Configuring the IGMP Leave Timer" section on page 20-8.

Explicit Host Tracking

Explicit Host Tracking (EHT) monitors group membership by tracking hosts that are sending IGMPv3 membership reports. This tracking enables a switch to detect host information associated with the groups of each port. Furthermore, EHT enables the user to track the membership and various statistics.

EHT enables a switch to track membership on a per-port basis. Consequently, a switch is aware of the hosts residing on each port and can perform immediate-leave processing when there is only one host behind a port.

To determine whether or not EHT is enabled on a VLAN, use the **show ip igmp snoop vlan** command.

Configuring IGMP Snooping

Note When configuring IGMP, configure the VLAN in the VLAN database mode. (See Chapter 13, "Configuring VLANs, VTP, and VMPS".)

IGMP snooping allows switches to examine IGMP packets and make forwarding decisions based on their content.

These sections describe how to configure IGMP snooping:

- Default IGMP Snooping Configuration, page 20-4
- Enabling IGMP Snooping Globally, page 20-5
- Enabling IGMP Snooping on a VLAN, page 20-6
- Configuring Learning Methods, page 20-6
- Configuring a Static Connection to a Multicast Router, page 20-7
- Enabling IGMP Immediate-Leave Processing, page 20-8
- Configuring the IGMP Leave Timer, page 20-8
- Configuring Explicit Host Tracking, page 20-10
- Configuring a Host Statically, page 20-10
- Suppressing Multicast Flooding, page 20-10

Default IGMP Snooping Configuration

Table 20-1 shows the IGMP snooping default configuration values.

Table 20-1 IGMP Snooping Default Configuration Values

Feature	Default Value
IGMP snooping	Enabled
Multicast routers	None configured
Explicit Host Tracking	Enabled for IGMPv3; Not available for IGMPv2
Immediate-leave processing	Enabled for IGMPv3; Disabled for IGMPv2
Report Suppression	Enabled
IGMP snooping learning method	PIM/DVMRP[1]

[1] PIM/DVMRP = Protocol Independent Multicast/Distance Vector Multicast Routing Protocol

Enabling IGMP Snooping Globally

To enable IGMP snooping globally, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# [no] ip igmp snooping`	Enables IGMP snooping. Use the **no** keyword to disable IGMP snooping.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show ip igmp snooping | include`	Verifies the configuration.

This example shows how to enable IGMP snooping globally and verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping
Switch(config)# end
Switch# show ip igmp snooping
Global IGMP Snooping configuration:
-----------------------------------
IGMP snooping              : Enabled
IGMPv3 snooping            : Enabled
Report suppression         : Enabled
TCN solicit query          : Disabled
TCN flood query count      : 2

Vlan 1:
--------
IGMP snooping                   : Enabled
IGMPv2 immediate leave          : Disabled
Explicit host tracking          : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode      : IGMP_ONLY

Vlan 2:
--------
IGMP snooping                   : Enabled
IGMPv2 immediate leave          : Disabled
Explicit host tracking          : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode      : IGMP_ONLY
```

Enabling IGMP Snooping on a VLAN

To enable IGMP snooping on a VLAN, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] ip igmp snooping vlan vlan_ID`	Enables IGMP snooping. Use the **no** keyword to disable IGMP snooping.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show ip igmp snooping vlan vlan_ID`	Verifies the configuration.

This example shows how to enable IGMP snooping on VLAN 2 and verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 2
Switch(config)# end
Switch# show ip igmp snooping vlan 2
Global IGMP Snooping configuration:
-----------------------------------
IGMP snooping             : Enabled
IGMPv3 snooping           : Enabled
Report suppression        : Enabled
TCN solicit query         : Disabled
TCN flood query count     : 2

Vlan 2:
--------
IGMP snooping                  : Enabled
IGMPv2 immediate leave         : Disabled
Explicit host tracking         : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode     : IGMP_ONLY
```

Configuring Learning Methods

The following sections describe IGMP snooping learning methods:

- Configuring PIM/DVMRP Learning, page 20-6
- Configuring CGMP Learning, page 20-7

Configuring PIM/DVMRP Learning

To configure IGMP snooping to learn from PIM/DVMRP packets, perform this task:

Command	Purpose
`Switch(config)# ip igmp snooping vlan vlan_ID mrouter learn [cgmp	pim-dvmrp]`	Specifies the learning method for the VLAN.

This example shows how to configure IP IGMP snooping to learn from PIM/DVMRP packets:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 1 mrouter learn pim-dvmrp
Switch(config)# end
Switch#
```

Configuring CGMP Learning

To configure IGMP snooping to learn from CGMP self-join packets, perform this task:

Command	Purpose
`Switch(config)# ip igmp snooping vlan vlan_ID mrouter learn [cgmp	pim-dvmrp]`	Specifies the learning method for the VLAN.

This example shows how to configure IP IGMP snooping to learn from CGMP self-join packets:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 1 mrouter learn cgmp
Switch(config)# end
Switch#
```

Configuring a Static Connection to a Multicast Router

To configure a static connection to a multicast router, enter the **ip igmp snooping vlan mrouter interface** command on the switch.

To configure a static connection to a multicast router, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`witch(config)# ip igmp snooping vlan vlan_ID mrouter interface interface_num`	Specifies a static connection to a multicast router for the VLAN. **Note** The interface to the router must be in the VLAN where you are entering the command. The router and the line protocol must be up.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show ip igmp snooping mrouter vlan vlan_ID`	Verifies the configuration.

This example shows how to configure a static connection to a multicast router:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 200 mrouter interface fastethernet 2/10
Switch# show ip igmp snooping mrouter vlan 200
vlan   ports
-----+----------------------------------------
 200   Fa2/10
Switch#
```

Enabling IGMP Immediate-Leave Processing

When you enable IGMP immediate-leave processing on a VLAN, a switch removes an interface from the multicast group when it detects an IGMPv2 leave message on that interface.

Note For IGMPv3, immediate-leave processing is enabled by default with EHT.

To enable immediate-leave processing on an IGMPv2 interface, perform this task:

Command	Purpose
`Switch(config)# ip igmp snooping vlan vlan_ID immediate-leave`	Enables immediate-leave processing in the VLAN. **Note** This command applies only to IGMPv2 hosts.

This example shows how to enable IGMP immediate-leave processing on interface VLAN 200 and to verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 200 immediate-leave
Configuring immediate leave on vlan 200
Switch(config)# end
Switch# show ip igmp interface vlan 200 | include immediate leave
Immediate leave                : Disabled
Switch(config)#
```

Configuring the IGMP Leave Timer

Follows these guidelines when configuring the IGMP leave timer:

- You can configure the leave time globally or per VLAN.
- Configuring the leave time on a VLAN overrides the global setting.
- The default leave time is 1000 milliseconds.
- The IGMP configurable leave time is only supported on hosts running IGMP Version 2.
- The actual leave latency in the network is usually the configured leave time. However, the leave time *might* vary around the configured time, depending on real-time CPU load conditions, network delays and the amount of traffic sent through the interface.

To enable the IGMP configurable-leave timer, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# ip igmp snooping last-member-query-interval time`	Configure the IGMP leave timer globally. The range is 100 to 5000 milliseconds. The default is 1000 seconds. To globally reset the IGMP leave timer to the default setting, use the global configuration command **no ip igmp snooping last-member-query-interva**l.

	Command	Purpose
Step 3	`Switch(config)#` **`ip igmp snooping vlan`** *`vlan_ID`* **`last-member-query-interval`** *`time`*	(Optional) Configure the IGMP leave time on the VLAN interface. The range is 100 to 5000 milliseconds. To remove the configured IGMP leave-time setting from the specified VLAN, use the global configuration command **no ip igmp snooping vlan** *vlan-id* **last-member-query-interval** **Note** Configuring the leave time on a VLAN overrides the globally configured timer.
Step 4	`Switch(config)#` **`end`**	Return to privileged EXEC mode.
Step 5	`Switch#` **`show ip igmp snooping`**	(Optional) Display the configured IGMP leave time.
Step 6	`Switch#` **`copy running-config startup-config`**	(Optional) Save your entries in the configuration file.

This example shows how to enable the IGMP configurable-leave timer and to verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping last-member-query-interval 200
Switch(config)# ip igmp snooping vlan 10 last-member-query-interval 500
Switch(config)# end
Switch# show ip igmp snooping show ip igmp snooping
Global IGMP Snooping configuration:
--------------------------

IGMP snooping              : Enabled
IGMPv3 snooping            : Enabled
Report suppression         : Enabled
TCN solicit query          : Disabled
TCN flood query count      : 2
Last Member Query Interval : 200

Vlan 1:
--------
IGMP snooping                       : Enabled
IGMPv2 immediate leave              : Disabled
Explicit host tracking              : Enabled
Multicast router learning mode      : pim-dvmrp
Last Member Query Interval          : 200
CGMP interoperability mode          : IGMP_ONLY

Vlan 10:
--------
IGMP snooping                       : Enabled
IGMPv2 immediate leave              : Disabled
Explicit host tracking              : Enabled
Multicast router learning mode      : pim-dvmrp
Last Member Query Interval          : 500
CGMP interoperability mode          : IGMP_ONLY

Switch#
```

Configuring Explicit Host Tracking

For IGMPv3, EHT is enabled by default and can be disabled on a per-VLAN basis.

To disable EHT processing on a VLAN, perform this task:

Command	Purpose
`Switch(config)#[no] ip igmp snooping vlan vlan_ID explicit-tracking`	Enables EHT on a VLAN. The **no** keyword disables EHT.

This example shows how to disable IGMP EHT on VLAN 200 and to verify the configuration:

```
Switch# configure terminal
Switch(config)# no ip igmp snooping vlan 200 explicit-tracking
Switch(config)# end
Switch# show ip igmp snooping vlan 200 | include Explicit host tracking
Explicit host tracking         : Disabled
```

Configuring a Host Statically

Hosts normally join multicast groups dynamically, but you can also configure a host statically on an interface.

To configure a host statically on an interface, perform this task:

Command	Purpose
`Switch(config-if)# ip igmp snooping vlan vlan_ID static mac_address interface interface_num`	Configures a host statically in the VLAN. **Note** This command cannot be configured to receive traffic for specific source IP addresses.

This example shows how to configure a host statically in VLAN 200 on interface FastEthernet 2/11:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 200 static 0100.5e02.0203 interface fastethernet
2/11
Configuring port FastEthernet2/11 on group 0100.5e02.0203 vlan 200
Switch(config)# end
```

Suppressing Multicast Flooding

An IGMP snooping-enabled switch floods multicast traffic to all ports in a VLAN when a spanning-tree Topology Change Notification (TCN) is received. Multicast flooding suppression enables a switch to stop sending such traffic. To support flooding suppression, the following interface and global commands were introduced in Cisco IOS Release 12.1(11b)EW.

The interface command is as follows:

[**no** | **default**] **ip igmp snooping tcn flood**

The global commands are as follows:

[**no** | **default**] **ip igmp snooping tcn flood query count** [**1** - **10**]

[**no** | **default**] **ip igmp snooping tcn query solicit**

Prior to Cisco IOS Release 12.1(11b)EW, when a spanning tree topology change notification (TCN) was received by a switch, the multicast traffic was flooded to all the ports in a VLAN for a period of three IGMP query intervals. This was necessary for redundant configurations. In Cisco IOS Release 12.1(11b)EW, the default time period the switch waits before multicast flooding stops was changed to two IGMP query intervals.

This flooding behavior is undesirable if the switch that does the flooding has many ports that are subscribed to different groups. The traffic could exceed the capacity of the link between the switch and the end host, resulting in packet loss.

With the **no ip igmp snooping tcn flood** command, you can disable multicast flooding on a switch interface following a topology change. Only the multicast groups that have been joined by a port are sent to that port, even during a topology change.

With the **ip igmp snooping tcn flood query count** command, you can enable multicast flooding on a switch interface for a short period of time following a topology change by configuring an IGMP query threshold.

Typically, if a topology change occurs, the spanning tree root switch issues a global IGMP leave message (referred to as a "query solicitation") with the group multicast address 0.0.0.0. When a switch receives this solicitation, it floods this solicitation on all ports in the VLAN where the spanning tree change occurred. When the upstream router receives this solicitation, it immediately issues an IGMP general query.

With the **ip igmp snooping tcn query solicit** command, you can now direct a non-spanning tree root switch to issue the same query solicitation.

The following sections provide additional details on the new commands and illustrate how you can use them.

IGMP Snooping Interface Configuration

A topology change in a VLAN may invalidate previously learned IGMP snooping information. A host that was on one port before the topology change may move to another port after the topology change. When the topology changes, the Catalyst 4500 series switch takes special actions to ensure that multicast traffic is delivered to all multicast receivers in that VLAN.

When the spanning tree protocol is running in a VLAN, a spanning tree topology change notification (TCN) is issued by the root switch in the VLAN. A Catalyst 4500 series switch that receives a TCN in a VLAN for which IGMP snooping has been enabled immediately enters into "multicast flooding mode" for a period of time until the topology restabilizes and the new locations of all multicast receivers are learned.

While in "multicast flooding mode," IP multicast traffic is delivered to all ports in the VLAN, and not restricted to those ports on which multicast group members have been detected.

Starting with Cisco IOS Release 12.1(11b)EW, you can manually prevent IP multicast traffic from being flooded to a switchport by using the **no ip igmp snooping tcn flood** command on that port.

For trunk ports, the configuration applies to all VLANs.

By default, multicast flooding is enabled. Use the **no** keyword to disable flooding, and use **default** to restore the default behavior (flooding is enabled).

To disable multicast flooding on an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet | tengigabitethernet} slot/port`	Selects the interface to configure.
Step 2	`Switch(config-if)# no ip igmp snooping tcn flood`	Disables multicast flooding on the interface when TCNs are received by the switch. To enable multicast flooding on the interface, enter this command: **default ip igmp snooping tcn flood**
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show running interface {fastethernet | gigabitethernet | tengigabitethernet} slot/port`	Verifies the configuration.

This example shows how to disable multicast flooding on interface FastEthernet 2/11:

```
Switch(config)# interface fastethernet 2/11
Switch(config-if)# no ip igmp snooping tcn flood
Switch(config-if)# end
Switch#
```

IGMP Snooping Switch Configuration

By default, “flooding mode” persists until the switch receives two IGMP general queries. You can change this period of time by using the **ip igmp snooping tcn flood query count** *n* command, where *n* is a number between 1 and 10.

This command operates at the global configuration level.

The default number of queries is 2. The **no** and **default** keywords restore the default.

To establish an IGMP query threshold, perform this task:

	Command	Purpose
Step 1	`Switch(config)# ip igmp snooping tcn flood query count <n>`	Modifies the number of IGMP queries the switch waits for before it stops flooding multicast traffic. To return the switch to the default number of IGMP queries, enter this command: **default ip igmp snooping tcn flood query count** .
Step 2	`Switch(config)# end`	Exits configuration mode.

This example shows how to modify the switch to stop flooding multicast traffic after four queries:

```
Switch(config)# ip igmp snooping tcn flood query count 4
Switch(config)# end
Switch#
```

When a spanning tree root switch receives a topology change in an IGMP snooping-enabled VLAN, the switch issues a query solicitation that causes an IOS router to send out one or more general queries. The new command **ip igmp snooping tcn query solicit** causes the switch to send the query solicitation whenever it notices a topology change, even if that switch is not the spanning tree root.

This command operates at the global configuration level.

By default, query solicitation is disabled unless the switch is the spanning tree root. The **default** keyword restores the default behavior.

To direct a switch to send a query solicitation, perform this task:

	Command	Purpose

Step 1	`Switch(config)#`
	`ip igmp snooping tcn query solicit`
		Configures the switch to send a query solicitation when a TCN is detected.
		To stop the switch from sending a query solicitation (if it's not a spanning tree root switch), enter this command: **no ip igmp snooping tcn query solicit**

Step 2	`Switch(config)#` **`end`**	Exits configuration mode.

This example shows how to configure the switch to send a query solicitation upon detecting a TCN:

```
Switch(config)# ip igmp snooping tcn query solicit
Switch(config)# end
Switch#
```

Displaying IGMP Snooping Information

The following sections show how to display IGMP snooping information:

- Displaying Querier Information, page 20-13
- Displaying IGMP Host Membership Information, page 20-14
- Displaying Group Information, page 20-15
- Displaying Multicast Router Interfaces, page 20-16
- Displaying MAC Address Multicast Entries, page 20-16
- Displaying IGMP Snooping Information on a VLAN Interface, page 20-17

Displaying Querier Information

To display querier information, perform this task:

Command	Purpose
`Switch#` **`show ip igmp snooping querier`** `[`**`vlan`** *`vlan_ID`*`]`	Displays multicast router interfaces.

This example shows how to display the IGMP snooping querier information for all VLANs on the switch:

```
Switch# show ip igmp snooping querier
Vlan      IP Address     IGMP Version        Port
---------------------------------------------------
2         10.10.10.1     v2                  Router
3         172.20.50.22   v3                  Fa3/15
```

This example shows how to display the IGMP snooping querier information for VLAN 3:

```
Switch# show ip igmp snooping querier vlan 3
Vlan      IP Address     IGMP Version        Port
---------------------------------------------------
3         172.20.50.22   v3                  Fa3/15
```

Displaying IGMP Host Membership Information

Note By default, EHT maintains a maximum of 1000 entries in the EHT database. Once this limit is reached, no additional entries are created. To create additional entries, clear the database with the **clear ip igmp snooping membership vlan** command.

To display host membership information, perform this task:

Command	Purpose
Switch# **show ip igmp snooping membership** [**interface** *interface_num*][**vlan** *vlan_ID*] [**reporter** *a.b.c.d*] [**source** *a.b.c.d* **group** *a.b.c.d*]	Displays Explicit Host Tracking information. **Note** This command is valid only if EHT is enabled on the switch.

This example shows how to display host membership information for VLAN 20 and to delete the EHT database:

```
Switch# show ip igmp snooping membership vlan 20
#channels: 5
#hosts : 1
Source/Group Interface Reporter Uptime Last-Join Last-Leave

40.40.40.2/224.10.10.10 Gi4/1 20.20.20.20 00:23:37 00:06:50 00:20:30
40.40.40.3/224.10.10.10 Gi4/2 20.20.2020 00:23:37 00:06:50 00:20:30
40.40.40.4/224.10.10.10Gi4/1 20.20.20.20 00:39:42 00:09:17 -

40.40.40.5/224.10.10.10Fa2/1 20.20.20.20 00:39:42 00:09:17 -
40.40.40.6/224.10.10.10 Fa2/1 20.20.20.20 00:09:47 00:09:17 -

Switch# clear ip igmp snooping membership vlan 20
```

This example shows how to display host membership for interface gi4/1:

```
Switch# show ip igmp snooping membership interface gi4/1
#channels: 5
#hosts : 1
Source/Group Interface Reporter Uptime Last-Join Last-Leave

40.40.40.2/224.10.10.10 Gi4/1 20.20.20.20 00:23:37 00:06:50 00:20:30
40.40.40.4/224.10.10.10Gi4/1 20.20.20.20 00:39:42 00:09:17 -
```

This example shows how to display host membership for VLAN 20 and group 224.10.10.10:

```
Switch# show ip igmp snooping membership vlan 20 source 40.40.40.2 group 224.10.10.10
#channels: 5
#hosts : 1
Source/Group Interface Reporter Uptime Last-Join Last-Leave

40.40.40.2/224.10.10.10 Gi4/1 20.20.20.20 00:23:37 00:06:50 00:20:30
```

Displaying Group Information

To display detailed IGMPv3 information associated with a group, perform one of the following tasks:

Command	Purpose
Switch# **show ip igmp snooping groups** [**vlan** *vlan_ID*]	Displays groups, the type of reports that were received for the group (Host Type), and the list of ports on which reports were received. The report list includes neither the multicast router ports nor the complete forwarding port set for the group. Rather, it lists the ports on which the reports have been received. To display the complete forwarding port set for the group, display the CLI output for the MAC address that maps to this group by using the **show mac-address-table multicast** command.
Switch# **show ip igmp snooping groups** [**vlan** *vlan_ID a.b.c.d*] [**summary**|**sources**|**hosts**]	Displays information specific to a group address, providing details about the current state of the group with respect to sources and hosts. **Note** This command applies only to full IGMPv3 snooping support and can be used for IGMPv1, IGMPv2, or IGMPv3 groups.
Switch# **show ip igmp snooping groups** [**vlan** *vlan_ID*] [**count**]	Displays the total number of group addresses learned by the system on a global or per-VLAN basis.

This example shows how to display the host types and ports of a group in VLAN 1:

```
Switch# show ip igmp snooping groups vlan 10 226.6.6.7
Vlan      Group          Version     Ports
---------------------------------------------------------
10        226.6.6.7      v3          Fa7/13, Fa7/14
Switch>
```

This example shows how to display the current state of a group with respect to a source IP address:

```
Switch# show ip igmp snooping groups vlan 10 226.6.6.7 sources
Source information for group 226.6.6.7:
Timers: Expired sources are deleted on next IGMP General Query

SourceIP       Expires   Uptime    Inc Hosts Exc Hosts
-------------------------------------------------------
2.0.0.1        00:03:04  00:03:48  2         0
2.0.0.2        00:03:04  00:02:07  2         0
Switch>
```

This example shows how to display the current state of a group with respect to a host MAC address:

```
Switch# show ip igmp snooping groups vlan 10 226.6.6.7 hosts
IGMPv3 host information for group 226.6.6.7
Timers: Expired hosts are deleted on next IGMP General Query
```

```
Host (MAC/IP)  Filter mode    Expires   Uptime    # Sources
-------------------------------------------------------------
175.1.0.29     INCLUDE        stopped   00:00:51       2
175.2.0.30     INCLUDE        stopped   00:04:14       2
```

This example shows how to display summary information for an IGMPv3 group:

```
Switch# show ip igmp snooping groups vlan 10 226.6.6.7 summary
Group Address (Vlan 10)        : 226.6.6.7
Host type                      : v3
Member Ports                   : Fa7/13, Fa7/14
Filter mode                    : INCLUDE
Expires                        : stopped
Sources                        : 2
Reporters (Include/Exclude)    : 2/0
```

This example shows how to display the total number of group addresses learned by the system globally:

```
Switch# show ip igmp snooping groups count
    Total number of groups:   54
```

This example shows how to display the total number of group addresses learned on VLAN 5:

```
Switch# show ip igmp snooping groups vlan 5 count
    Total number of groups:   30
```

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
Switch# **show ip igmp snooping mrouter vlan** *vlan_ID*	Displays multicast router interfaces.

This example shows how to display the multicast router interfaces in VLAN 1:

```
Switch# show ip igmp snooping mrouter vlan 1
vlan            ports
-----+----------------------------------------
  1          Gi1/1,Gi2/1,Fa3/48,Router
Switch#
```

Displaying MAC Address Multicast Entries

To display MAC address multicast entries for a VLAN, perform this task:

Command	Purpose
Switch# **show mac-address-table multicast vlan** *vlan_ID* [**count**]	Displays MAC address multicast entries for a VLAN.

This example shows how to display MAC address multicast entries for VLAN 1:

```
Switch# show mac-address-table multicast vlan 1
Multicast Entries
 vlan    mac address     type    ports
-------+---------------+-------+-------------------------------------------
   1    0100.5e01.0101     igmp Switch,Gi6/1
   1    0100.5e01.0102     igmp Switch,Gi6/1
   1    0100.5e01.0103     igmp Switch,Gi6/1
   1    0100.5e01.0104     igmp Switch,Gi6/1
   1    0100.5e01.0105     igmp Switch,Gi6/1
   1    0100.5e01.0106     igmp Switch,Gi6/1
Switch#
```

This example shows how to display a total count of MAC address entries for VLAN 1:

```
Switch# show mac-address-table multicast vlan 1 count
Multicast MAC Entries for vlan 1:    4
Switch#
```

Displaying IGMP Snooping Information on a VLAN Interface

To display IGMP snooping information on a VLAN, perform this task:

Command	Purpose
`Switch# show ip igmp snooping vlan vlan_ID`	Displays IGMP snooping information on a VLAN interface.

This example shows how to display IGMP snooping information on VLAN 5:

```
Switch# show ip igmp snooping vlan 5
Global IGMP Snooping configuration:
-----------------------------------
IGMP snooping              :Enabled
IGMPv3 snooping support    :Full
Report suppression         :Enabled
TCN solicit query          :Disabled
TCN flood query count      :2

Vlan 5:
--------
IGMP snooping                   :Enabled
Immediate leave                 :Disabled
Explicit Host Tracking          :Disabled
Multicast router learning mode :pim-dvmrp
CGMP interoperability mode      :IGMP_ONLY
```

Configuring IGMP Filtering

This section includes the following subsections:

- Default IGMP Filtering Configuration, page 20-18
- Configuring IGMP Profiles, page 20-18
- Applying IGMP Profiles, page 20-19
- Setting the Maximum Number of IGMP Groups, page 20-20

Note The IGMP filtering feature works for IGMPv1 and IGMPv2 only.

In some environments, for example metropolitan or multiple-dwelling unit (MDU) installations, an administrator might want to control the multicast groups to which a user on a switch port can belong. This allows the administrator to control the distribution of multicast services, such as IP/TV, based on some type of subscription or service plan.

With the IGMP filtering feature, an administrator can exert this type of control. With this feature, you can filter multicast joins on a per-port basis by configuring IP multicast profiles and associating them with individual switch ports. An IGMP profile can contain one or more multicast groups and specifies whether access to the group is permitted or denied. If an IGMP profile denying access to a multicast group is applied to a switch port, the IGMP join report requesting the stream of IP multicast traffic is dropped, and the port is not allowed to receive IP multicast traffic from that group. If the filtering action permits access to the multicast group, the IGMP report from the port is forwarded for normal processing.

IGMP filtering controls only IGMP membership join reports and has no relationship to the function that directs the forwarding of IP multicast traffic.

You can also set the maximum number of IGMP groups that a Layer 2 interface can join with the **ip igmp max-groups <***n***>** command.

Default IGMP Filtering Configuration

Table 20-2 shows the default IGMP filtering configuration.

Table 20-2 Default IGMP Filtering Settings

Feature	Default Setting
IGMP filters	No filtering
IGMP maximum number of IGMP groups	No limit
IGMP profiles	None defined

Configuring IGMP Profiles

To configure an IGMP profile and to enter IGMP profile configuration mode, use the **ip igmp profile** global configuration command. From the IGMP profile configuration mode, you can specify the parameters of the IGMP profile to be used for filtering IGMP join requests from a port. When you are in IGMP profile configuration mode, you can create the profile using these commands:

- **deny**: Specifies that matching addresses are denied; this is the default condition.
- **exit**: Exits from igmp-profile configuration mode.
- **no**: Negates a command or sets its defaults.
- **permit**: Specifies that matching addresses are permitted.
- **range**: Specifies a range of IP addresses for the profile. You can enter a single IP address or a range with starting and ending addresses.

By default, no IGMP profiles are configured. When a profile is configured with neither the **permit** nor the **deny** keyword, the default is to deny access to the range of IP addresses.

To create an IGMP profile for a port, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`ip igmp profile`** *`profile number`*	Enters IGMP profile configuration mode~~,~~ and assigns a number to the profile you are configuring. The range is from 1 to 4,294,967,295.
Step 3	`Switch(config-igmp-profile)#` **`permit | deny`**	(Optional) Sets the action to permit or deny access to the IP multicast address. If no action is configured, the default for the profile is to deny access.
Step 4	`Switch(config-igmp-profile)#` **`range`** *`ip multicast address`*	Enters the IP multicast address or range of IP multicast addresses to which access is being controlled. If entering a range, enter the low IP multicast address, a space, and the high IP multicast address. You can use the **range** command multiple times to enter multiple addresses or ranges of addresses.
Step 5	`Switch(config-igmp-profile)#` **`end`**	Returns to privileged EXEC mode.
Step 6	`Switch#` **`show ip igmp profile`** *`profile number`*	Verifies the profile configuration.
Step 7	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To delete a profile, use the **no ip igmp profile** *profile number* global configuration command.

To delete an IP multicast address or range of IP multicast addresses, use the **no range** *ip multicast address* IGMP profile configuration command.

This example shows how to create IGMP profile 4 (allowing access to the single IP multicast address) and how to verify the configuration. If the action were to deny (the default), it would not appear in the **show ip igmp profile** command output.

```
Switch# configure terminal
Switch(config)# ip igmp profile 4
Switch(config-igmp-profile)# permit
Switch(config-igmp-profile)# range 229.9.9.0
Switch(config-igmp-profile)# end
Switch# show ip igmp profile 4
IGMP Profile 4
    permit
    range 229.9.9.0 229.9.9.0
```

Applying IGMP Profiles

To control access as defined in an IGMP profile, use the **ip igmp filter** interface configuration command to apply the profile to the appropriate interfaces. You can apply a profile to multiple interfaces, but each interface can only have one profile applied to it.

Note You can apply IGMP profiles to Layer 2 ports only. You cannot apply IGMP profiles to routed ports (or SVIs) or to ports that belong to an EtherChannel port group.

To apply an IGMP profile to a switch port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode, and enter the physical interface to configure, for example **fastethernet2/3**. The interface must be a Layer 2 port that does not belong to an EtherChannel port group.
Step 3	`Switch(config-if)# ip igmp filter profile number`	Applies the specified IGMP profile to the interface. The profile number can be from 1 to 4,294,967,295.
Step 4	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show running configuration interface interface-id`	Verifies the configuration.
Step 6	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To remove a profile from an interface, use the **no ip igmp filter** command.

This example shows how to apply IGMP profile 4 to an interface and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet2/12
Switch(config-if)# ip igmp filter 4
Switch(config-if)# end
Switch# show running-config interface fastethernet2/12
Building configuration...

Current configuration : 123 bytes
!
interface FastEthernet2/12
 no ip address
 shutdown
 snmp trap link-status
 ip igmp max-groups 25
 ip igmp filter 4
end
```

Setting the Maximum Number of IGMP Groups

You can set the maximum number of IGMP groups that a Layer 2 interface can join by using the **ip igmp max-groups** interface configuration command. Use the **no** form of this command to set the maximum back to the default, which is no limit.

Note This restriction can be applied to Layer 2 ports only. You cannot set a maximum number of IGMP groups on routed ports (or SVIs) or on ports that belong to an EtherChannel port group.

To apply an IGMP profile on a switch port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode, and enter the physical interface to configure, for example **gigabitethernet1/1**. The interface must be a Layer 2 port that does not belong to an EtherChannel group.
Step 3	`Switch(config-if)# ip igmp max-groups number`	Sets the maximum number of IGMP groups that the interface can join. The range is from 0 to 4,294,967,294. By default, no maximum is set. To remove the maximum group limitation and return to the default of no maximum, use the **no ip igmp max-groups** command.
Step 4	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show running-configuration interface interface-id`	Verifies the configuration.
Step 6	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to limit the number of IGMP groups that an interface can join to 25.

```
Switch# configure terminal
Switch(config)# interface fastethernet2/12
Switch(config-if)# ip igmp max-groups 25
Switch(config-if)# end
Switch# show running-config interface fastethernet2/12
Building configuration...

Current configuration : 123 bytes
!
interface FastEthernet2/12
 no ip address
 shutdown
 snmp trap link-status
 ip igmp max-groups 25
 ip igmp filter 4
end
```

Displaying IGMP Filtering Configuration

You can display IGMP profile and maximum group configuration for all interfaces on the switch or for a specified interface.

To display IGMP profiles, perform this task:

Command	Purpose
`Switch# show ip igmp profile [profile number]`	Displays the specified IGMP profile or all IGMP profiles defined on the switch.

To display interface configuration, perform this task:

Command	Purpose
Switch# **show running-configuration** [**interface** *interface-id*]	Displays the configuration of the specified interface or all interfaces on the switch, including (if configured) the maximum number of IGMP groups to which an interface can belong and the IGMP profile applied to the interface.

This is an example of the **show ip igmp profile** privileged EXEC command when no profile number is entered. All profiles defined on the switch are displayed.

```
Switch# show ip igmp profile
IGMP Profile 3
    range 230.9.9.0 230.9.9.0
IGMP Profile 4
    permit
    range 229.9.9.0 229.255.255.255
```

This is an example of the **show running-config** privileged EXEC command when an interface is specified with IGMP maximum groups configured and IGMP profile 4 has been applied to the interface.

```
Switch# show running-config interface fastethernet2/12
Building configuration...
Current configuration : 123 bytes
!
interface FastEthernet2/12
 no ip address
 shutdown
 snmp trap link-status
 ip igmp max-groups 25
 ip igmp filter 4
end
```

CHAPTER 21

Configuring IPv6 MLD Snooping

Note IPv6 MLD Snooping is *only* supported on Supervisor Engine 6-E.

You can use Multicast Listener Discovery (MLD) snooping to enable efficient distribution of IP version 6 (IPv6) multicast data to clients and routers in a switched network on the Catalyst 4500 series switch.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

This chapter includes these sections:

- "Understanding MLD Snooping" section on page 21-1
- "Configuring IPv6 MLD Snooping" section on page 21-5
- "Displaying MLD Snooping Information" section on page 21-11

Understanding MLD Snooping

In IP version 4 (IPv4), Layer 2 switches can use Internet Group Management Protocol (IGMP) snooping to limit the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. In IPv6, MLD snooping performs a similar function. With MLD snooping, IPv6 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This list is constructed by snooping IPv6 multicast control packets.

MLD is a protocol used by IPv6 multicast routers to discover the presence of multicast listeners (nodes wishing to receive IPv6 multicast packets) on its directly attached links and to discover which multicast packets are of interest to neighboring nodes. MLD is derived from IGMP; MLD version 1 (MLDv1) is equivalent to IGMPv2 and MLD version 2 (MLDv2) is equivalent to IGMPv3. MLD is a subprotocol of Internet Control Message Protocol version 6 (ICMPv6), and MLD messages are a subset of ICMPv6 messages, identified in IPv6 packets by a preceding Next Header value of 58.

The switch supports two versions of MLD snooping:

- MLDv1 snooping detects MLDv1 control packets and sets up traffic bridging based on IPv6 destination multicast addresses.
- MLDv2 basic snooping (MBSS) uses MLDv2 control packets to set up traffic forwarding based on IPv6 destination multicast addresses.

The switch can snoop on both MLDv1 and MLDv2 protocol packets and bridge IPv6 multicast data based on destination IPv6 multicast addresses.

Note The switch does not support MLDv2 enhanced snooping (MESS), which sets up IPv6 source and destination multicast address-based forwarding.

MLD snooping can be enabled or disabled globally or per VLAN. When MLD snooping is enabled, a per-VLAN IPv6 multicast MAC address table is constructed in software and a per-VLAN IPv6 multicast address table is constructed in software and hardware. The switch then performs IPv6 multicast-address based bridging in hardware.

These sections describe some parameters of IPv6 MLD snooping:

- MLD Messages, page 21-2
- MLD Queries, page 21-3
- Multicast Client Aging Robustness, page 21-3
- Multicast Router Discovery, page 21-3
- MLD Reports, page 21-4
- MLD Done Messages and Immediate-Leave, page 21-4
- Topology Change Notification Processing, page 21-5

MLD Messages

MLDv1 supports three types of messages:

- Listener Queries are the equivalent of IGMPv2 queries and are either General Queries or Multicast-Address-Specific Queries (MASQs).
- Multicast Listener Reports are the equivalent of IGMPv2 reports
- Multicast Listener Done messages are the equivalent of IGMPv2 leave messages.

MLDv2 supports MLDv2 queries and reports, as well as MLDv1 Report and Done messages.

Message timers and state transitions resulting from messages being sent or received are the same as those of IGMPv2 messages. MLD messages that do not have valid link-local IPv6 source addresses are ignored by MLD routers and switches.

MLD Queries

The switch sends out MLD queries, constructs an IPv6 multicast address database, and generates MLD group-specific and MLD group-and-source-specific queries in response to MLD Done messages. The switch also supports report suppression, report proxying, Immediate-Leave functionality, and static IPv6 multicast MAC-address configuration.

When MLD snooping is disabled, all MLD queries are flooded in the ingress VLAN.

When MLD snooping is enabled, received MLD queries are flooded in the ingress VLAN, and a copy of the query is sent to the CPU for processing. From the received query, MLD snooping builds the IPv6 multicast address database. It detects multicast router ports, maintains timers, sets report response time, learns the querier IP source address for the VLAN, learns the querier port in the VLAN, and maintains multicast-address aging.

When a group exists in the MLD snooping database, the switch responds to a group-specific query by sending an MLDv1 report. When the group is unknown, the group-specific query is flooded to the ingress VLAN.

When a host wants to leave a multicast group, it can send out an MLD Done message (equivalent to IGMP Leave message). When the switch receives an MLDv1 Done message, if Immediate- Leave is not enabled, the switch sends an MASQ to the port from which the message was received to determine if other devices connected to the port should remain in the multicast group.

Multicast Client Aging Robustness

You can configure port membership removal from addresses based on the number of queries. A port is removed from membership to an address only when there are no reports to the address on the port for the configured number of queries. The default number is 2.

Multicast Router Discovery

Like IGMP snooping, MLD snooping performs multicast router discovery, with these characteristics:

- Ports configured by a user never age out.
- Dynamic port learning results from MLDv1 snooping queries and IPv6 PIMv2 packets.
- If there are multiple routers on the same Layer 2 interface, MLD snooping tracks a single multicast router on the port (the router that most recently sent a router control packet).
- Dynamic multicast router port aging is based on a default timer of 5 minutes; the multicast router is deleted from the router port list if no control packet is received on the port for 5 minutes.
- IPv6 multicast router discovery only takes place when MLD snooping is enabled on the switch.

- Received IPv6 multicast router control packets are always flooded to the ingress VLAN, whether or not MLD snooping is enabled on the switch.
- After the discovery of the first IPv6 multicast router port, unknown IPv6 multicast data is forwarded only to the discovered router ports (before that time, all IPv6 multicast data is flooded to the ingress VLAN).

MLD Reports

The processing of MLDv1 join messages is essentially the same as with IGMPv2. When no IPv6 multicast routers are detected in a VLAN, reports are not processed or forwarded from the switch. When IPv6 multicast routers are detected and an MLDv1 report is received, an IPv6 multicast group address and an IPv6 multicast MAC address are entered in the VLAN MLD database. Then all IPv6 multicast traffic to the group within the VLAN is forwarded using this address. When MLD snooping is disabled, reports are flooded in the ingress VLAN.

When MLD snooping is enabled, MLD report suppression, called listener message suppression, is automatically enabled. With report suppression, the switch forwards the first MLDv1 report received by a group to IPv6 multicast routers; subsequent reports for the group are not sent to the routers. When MLD snooping is disabled, report suppression is disabled, and all MLDv1 reports are flooded to the ingress VLAN.

The switch also supports MLDv1 proxy reporting. When an MLDv1 MASQ is received, the switch responds with MLDv1 reports for the address on which the query arrived if the group exists in the switch on another port and if the port on which the query arrived is not the last member port for the address.

MLD Done Messages and Immediate-Leave

When the Immediate-Leave feature is enabled and a host sends an MLDv1 Done message (equivalent to an IGMP leave message), the port on which the Done message was received is immediately deleted from the group.You enable Immediate-Leave on VLANs and (as with IGMP snooping), you should only use the feature on VLANs where a single host is connected to the port. If the port was the last member of a group, the group is also deleted, and the leave information is forwarded to the detected IPv6 multicast routers.

When Immediate Leave is not enabled in a VLAN (which would be the case when there are multiple clients for a group on the same port) and a Done message is received on a port, an MASQ is generated on that port. The user can control when a port membership is removed for an existing address in terms of the number of MASQs. A port is removed from membership to an address when there are no MLDv1 reports to the address on the port for the configured number of queries.

The number of MASQs generated is configured by using the **ipv6 mld snooping last-listener-query count** global configuration command. The default number is 2.

The MASQ is sent to the IPv6 multicast address for which the Done message was sent. If there are no reports sent to the IPv6 multicast address specified in the MASQ during the switch maximum response time, the port on which the MASQ was sent is deleted from the IPv6 multicast address database. The maximum response time is the time configured by using the **ipv6 mld snooping last-listener-query-interval** global configuration command. If the deleted port is the last member of the multicast address, the multicast address is also deleted, and the switch sends the address leave information to all detected multicast routers.

Topology Change Notification Processing

When topology change notification (TCN) solicitation is enabled by using the **ipv6 mld snooping tcn query solicit** global configuration command, MLDv1 snooping sets the VLAN to flood all IPv6 multicast traffic with a configured number of MLDv1 queries before it begins sending multicast data only to selected ports. You set this value by using the **ipv6 mld snooping tcn flood query count** global configuration command. The default is to send two queries. The switch also generates MLDv1 global Done messages with valid link-local IPv6 source addresses when the switch becomes the STP root in the VLAN or when it is configured by the user. This is same as done in IGMP snooping.

Configuring IPv6 MLD Snooping

These sections describe how to configure IPv6 MLD snooping:

- Default MLD Snooping Configuration, page 21-5
- MLD Snooping Configuration Guidelines, page 21-6
- Enabling or Disabling MLD Snooping, page 21-6
- Configuring a Static Multicast Group, page 21-7
- Configuring a Multicast Router Port, page 21-8
- Enabling MLD Immediate Leave, page 21-8
- Configuring MLD Snooping Queries, page 21-9
- Disabling MLD Listener Message Suppression, page 21-10

Default MLD Snooping Configuration

Table 21-1 shows the default MLD snooping configuration.

Table 21-1 Default MLD Snooping Configuration

Feature	Default Setting
MLD snooping (Global)	Disabled.
MLD snooping (per VLAN)	Enabled. MLD snooping must be globally enabled for VLAN MLD snooping to take place.
IPv6 Multicast addresses	None configured.
IPv6 Multicast router ports	None configured.
MLD snooping Immediate Leave	Disabled.
MLD snooping robustness variable	Global: 2; Per VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query count	Global: 2; Per VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.

Table 21-1 Default MLD Snooping Configuration (continued)

Feature	Default Setting
Last listener query interval	Global: 1000 (1 second); VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval.
TCN query solicit	Disabled.
TCN query count	2.
MLD listener suppression	Disabled.

MLD Snooping Configuration Guidelines

When configuring MLD snooping, consider these guidelines:

- You can configure MLD snooping characteristics at any time, but you must globally enable MLD snooping by using the **ipv6 mld snooping** global configuration command for the configuration to take effect.
- MLD snooping and IGMP snooping act independently of each other. You can enable both features at the same time on the switch. The total number of IPv4 and IPv6 multicast groups entries that can coexist on the Catalyst 4500 series switch is limited to 16384.
- The Supervisor Engine 6-E with 512MB of memory supports about 11000 MLD Snooping multicast groups whereas a Supervisor Engine 6-E with 1GB memory supports 16384 MLD Snooping multicast groups.

Enabling or Disabling MLD Snooping

By default, IPv6 MLD snooping is globally disabled on the switch and enabled on all VLANs. When MLD snooping is globally disabled, it is also disabled on all VLANs. When you globally enable MLD snooping, the VLAN configuration overrides the global configuration. That is, MLD snooping is enabled only on VLAN interfaces in the default state (enabled).

You can enable and disable MLD snooping on a per-VLAN basis or for a range of VLANs, but if you globally disable MLD snooping, it is disabled in all VLANs. If global snooping is enabled, you can enable or disable VLAN snooping.

To globally enable MLD snooping on the switch, follow these steps:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# ipv6 mld snooping`	Globally enables MLD snooping on the switch.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To globally disable MLD snooping on the switch, use the **no ipv6 mld snooping** global configuration command.

To enable MLD snooping on a VLAN, follow these steps

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# ipv6 mld snooping`	Globally enables MLD snooping on the switch.
Step 3	`Switch(config)# ipv6 mld snooping vlan` *vlan-id*	Enables MLD snooping on the VLAN.The VLAN ID range is 1 to 1001 and 1006 to 4094. **Note** MLD snooping must be globally enabled for VLAN snooping to be enabled.
Step 4	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To disable MLD snooping on a VLAN interface, use the **no ipv6 mld snooping vlan** *vlan-id* global configuration command for the specified VLAN number.

Configuring a Static Multicast Group

Hosts or Layer 2 ports normally join multicast groups dynamically, but you can also statically configure an IPv6 multicast address and member ports for a VLAN.

To add a Layer 2 port as a member of a multicast group, follow these steps :

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode
Step 2	`Switch(config)# ipv6 mld snooping vlan` *vlan-id* `static` *ipv6_multicast_address* `interface` *interface-id*	Statically configures a multicast group with a Layer 2 port as a member of a multicast group: • *vlan-id* is the multicast group VLAN ID. The VLAN ID range is 1 to 1001 and 1006 to 4094. • *ipv6_multicast_address* is the 128-bit group IPv6 address. The address must be in the form specified in RFC 2373. • *interface-id* is the member port. It can be a physical interface or a port channel (1 to 64).
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show mac-address-table multicast mld-snooping`	Verifies the static member port and the IPv6 address.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To remove a Layer 2 port from the multicast group, use the **no ipv6 mld snooping vlan** *vlan-id* **static** *mac-address* **interface** *interface-id* global configuration command. If all member ports are removed from a group, the group is deleted.

This example shows how to statically configure an IPv6 MAC address:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping vlan 2 static 3333.0000.0003 interface
gigabitethernet1/1
Switch(config)# end
```

Configuring a Multicast Router Port

Although MLD snooping learns about router ports through MLD queries and PIMv6 queries, you can also use the command-line interface (CLI) to add a multicast router port to a VLAN. To add a multicast router port (add a static connection to a multicast router), use the **ipv6 mld snooping vlan mrouter** global configuration command on the switch.

Note Static connections to multicast routers are supported only on switch ports.

To add a multicast router port to a VLAN, follow these steps:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# ipv6 mld snooping vlan` *`vlan-id`* `mrouter interface` *interface-id*	Specifies the multicast router VLAN ID, and specify the interface to the multicast router. • The VLAN ID range is 1 to 1001 and 1006 to 4094. • The interface can be a physical interface or a port channel. The port-channel range is 1 to 64.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show ipv6 mld snooping mrouter` [`vlan` *`vlan-id`*]	Verifies that IPv6 MLD snooping is enabled on the VLAN interface.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To remove a multicast router port from the VLAN, use the **no ipv6 mld snooping vlan** *vlan-id* **mrouter interface** *interface-id* global configuration command.

This example shows how to add a multicast router port to VLAN 200:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping vlan 200 mrouter interface gigabitethernet1/0/2
Switch(config)# exit
```

Enabling MLD Immediate Leave

When you enable MLDv1 Immediate Leave, the switch immediately removes a port from a multicast group when it detects an MLD Done message on that port. You should only use the Immediate-Leave feature when there is a single receiver present on every port in the VLAN. When there are multiple clients for a multicast group on the same port, you should not enable Immediate-Leave in a VLAN.

To enable MLDv1 Immediate Leave, follow these steps:

	Command	Purpose
Step 1	`Switch# `**`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)# `**`ipv6 mld snooping vlan`** *`vlan-id`* **`immediate-leave`**	Enables MLD Immediate Leave on the VLAN interface.
Step 3	`Switch(config)# `**`end`**	Returns to privileged EXEC mode.
Step 4	`Switch# `**`show ipv6 mld snooping vlan`** *vlan-id*	Verifies that Immediate Leave is enabled on the VLAN interface.
Step 5	`Switch# `**`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To disable MLD Immediate Leave on a VLAN, use the **no ipv6 mld snooping vlan** *vlan-id* **immediate-leave** global configuration command.

This example shows how to enable MLD Immediate Leave on VLAN 130:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping vlan 130 immediate-leave
Switch(config)# exit
```

Configuring MLD Snooping Queries

When Immediate Leave is not enabled and a port receives an MLD Done message, the switch generates MASQs on the port and sends them to the IPv6 multicast address for which the Done message was sent. You can optionally configure the number of MASQs that are sent and the length of time the switch waits for a response before deleting the port from the multicast group.

To configure MLD snooping query characteristics for the switch or for a VLAN, follow these steps:

	Command	Purpose
Step 1	`Switch# `**`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)# `**`ipv6 mld snooping robustness-variable`** *`value`*	(Optional) Sets the number of queries that are sent before switch will deletes a listener (port) that does not respond to a general query. The range is 1 to 3; the default is 2.
Step 3	`Switch(config)# `**`ipv6 mld snooping vlan`** *vlan-id* **`robustness-variable`** *`value`*	(Optional) Sets the robustness variable on a VLAN basis, which determines the number of general queries that MLD snooping sends before aging out a multicast address when there is no MLD report response. The range is 1 to 3; the default is 0. When set to 0, the number used is the global robustness variable value.
Step 4	`Switch(config)# `**`ipv6 mld snooping last-listener-query-count`** *`count`*	(Optional) Sets the number of MASQs that the switch sends before aging out an MLD client. The range is 1 to 7; the default is 2. The queries are sent 1 second apart.
Step 5	`Switch(config)# `**`ipv6 mld snooping vlan`** *vlan-id* **`last-listener-query-count`** *`count`*	(Optional) Sets the last-listener query count on a VLAN basis. This value overrides the value configured globally. The range is 1 to 7; the default is 0. When set to 0, the global count value is used. Queries are sent 1 second apart.

	Command	Purpose
Step 6	Switch(config)# **ipv6 mld snooping last-listener-query-interval** *interval*	(Optional) Sets the maximum response time that the switch waits after sending out a MASQ before deleting a port from the multicast group. The range is 100 to 32,768 thousands of a second. The default is 1000 (1 second).
Step 7	Switch(config)# **ipv6 mld snooping vlan** *vlan-id* **last-listener-query-interval** *interval*	(Optional) Sets the last-listener query interval on a VLAN basis. This value overrides the value configured globally. The range is 0 to 32,768 thousands of a second. The default is 0. When set to 0, the global last-listener query interval is used.
Step 8	Switch(config)# **ipv6 mld snooping tcn query solicit**	(Optional) Enables topology change notification (TCN) solicitation, which means that VLANs flood all IPv6 multicast traffic for the configured number of queries before sending multicast data to only those ports requesting to receive it. The default is for TCN to be disabled.
Step 9	Switch(config)# **ipv6 mld snooping tcn flood query count** *count*	(Optional) When TCN is enabled, specifies the number of TCN queries to be sent. The range is from 1 to 10; the default is 2.
Step 10	Switch(config)# **end**	Returns to privileged EXEC mode.
Step 11	Switch# **show ipv6 mld snooping querier** [**vlan** *vlan-id*]	(Optional) Verifies that the MLD snooping querier information for the switch or for the VLAN.
Step 12	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to set the MLD snooping global robustness variable to 3:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping robustness-variable 3
Switch(config)# exit
```

This example shows how to set the MLD snooping last-listener query count for a VLAN to 3:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping vlan 200 last-listener-query-count 3
Switch(config)# exit
```

This example shows how to set the MLD snooping last-listener query interval (maximum response time) to 2000 (2 seconds):

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping last-listener-query-interval 2000
Switch(config)# exit
```

Disabling MLD Listener Message Suppression

MLD snooping listener message suppression is enabled by default. When it is enabled, the switch forwards only one MLD report per multicast router query. When message suppression is disabled, multiple MLD reports could be forwarded to the multicast routers.

To disable MLD listener message suppression, follow these steps:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`no ipv6 mld snooping listener-message-suppression`**	Disables MLD message suppression.
Step 3	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Switch#` **`show ipv6 mld snooping`**	Verifies that IPv6 MLD snooping report suppression is disabled.
Step 5	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To re-enable MLD message suppression, use the **ipv6 mld snooping listener-message-suppression** global configuration command.

Displaying MLD Snooping Information

You can display MLD snooping information for dynamically learned and statically configured router ports and VLAN interfaces. You can also display MAC address multicast entries for a VLAN configured for MLD snooping.

To display MLD snooping information, use one or more of the privileged EXEC commands in Table 21-2.

Table 21-2 Commands for Displaying MLD Snooping Information

Command	Purpose
`show ipv6 mld snooping` `[`**`vlan`** *`vlan-id`*`]`	Display the MLD snooping configuration information for all VLANs on the switch or for a specified VLAN. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
`show ipv6 mld snooping mrouter` `[`**`vlan`** *`vlan-id`*`]`	Display information on dynamically learned and manually configured multicast router interfaces. When you enable MLD snooping, the switch automatically learns the interface to which a multicast router is connected. These are dynamically learned interfaces. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
`show ipv6 mld snooping querier` `[`**`vlan`** *`vlan-id`*`]`	Display information about the IPv6 address and incoming port for the most-recently received MLD query messages in the VLAN. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN.The VLAN ID range is 1 to 1001 and 1006 to 4094.

CHAPTER 22

Configuring 802.1Q and Layer 2 Protocol Tunneling

Virtual private networks (VPNs) provide enterprise-scale connectivity on a shared infrastructure, often Ethernet-based, with the same security, prioritization, reliability, and manageability requirements of private networks. Tunneling is a feature designed for service providers who carry traffic of multiple customers across their networks and who are required to maintain the VLAN and Layer 2 protocol configurations of each customer without impacting the traffic of other customers. The Catalyst 4500 series switch supports IEEE 802.1Q tunneling and Layer 2 protocol tunneling.

Note Supervisor Engine 6-E does *not* support Layer 2 Protocol Tunneling.

Note Be aware that 802.1Q requires the Cisco Catalyst 4948, the Cisco Catalyst 4948-10GE, or the Catalyst 4500 series switch supervisor engines II-Plus-10GE V, or V-10GE; Layer 2 protocol tunneling is supported on all supervisor engines.

This chapter contains these sections:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Understanding 802.1Q Tunneling

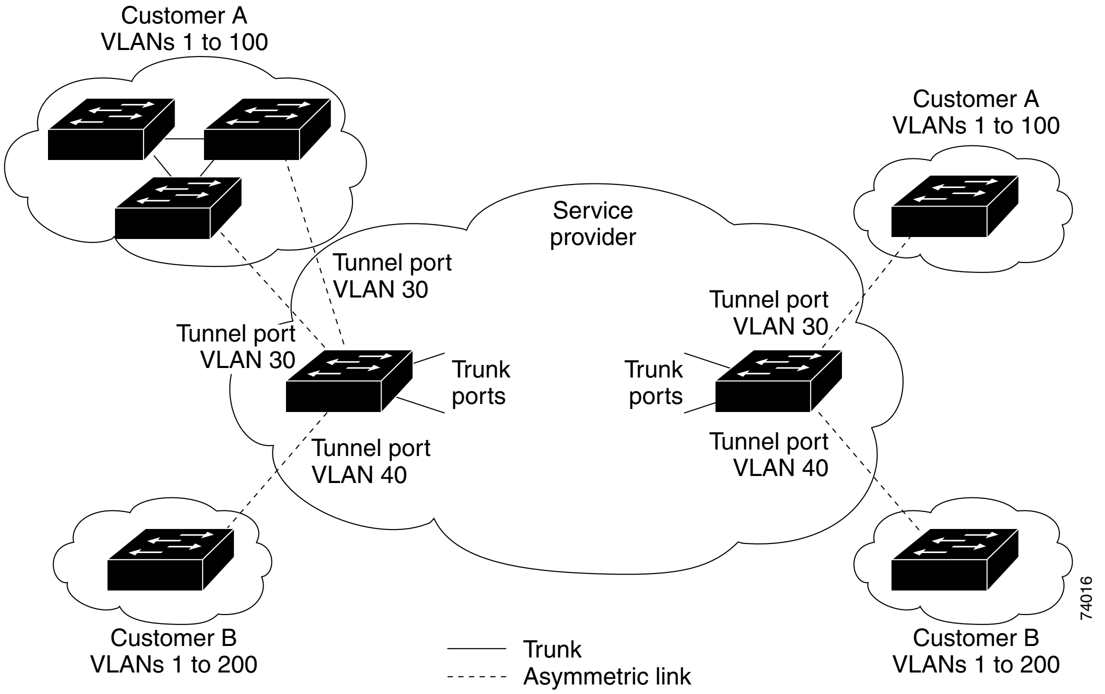
The VLAN ranges required by different customers in the same Service Provider network might overlap, and customer traffic through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit (4096) of the 802.1Q specification.

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 ID that is dedicated to tunneling. Each customer requires a separate Service Provider VLAN ID, but that Service Provider VLAN ID supports VLANs of all the customers.

Customer traffic tagged in the normal way with appropriate VLAN IDs comes from an 802.1Q trunk port on the customer device and into a tunnel port on the Service Provider edge switch. The link between the customer device and the edge switch is asymmetric 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 interface to an access VLAN ID that is unique to each customer. See Figure 22-1.

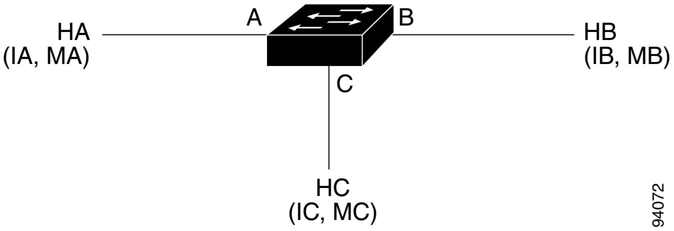
Figure 22-1 802.1Q Tunnel Ports in a Service Provider Network



Packets coming from the customer trunk port into the tunnel port on the Service Provider edge switch are normally 802.1Q-tagged with the appropriate VLAN ID. When the tagged packets exit the trunk port into the Service Provider network, they are encapsulated with another layer of an 802.1Q tag (called the *metro tag*) that contains the VLAN ID that is unique to the customer. The original customer 802.1Q tag is preserved in the encapsulated packet. Therefore, packets entering the Service Provider network are double-tagged, with the metro tag containing the customer's access VLAN ID, and the inner VLAN ID being that of the incoming traffic.

When the double-tagged packet enters another trunk port in a Service Provider core switch, the metro tag is stripped as the switch processes the packet. When the packet exits another trunk port on the same core switch, the same metro tag is again added to the packet. Figure 22-2 shows the tag structures of the Ethernet packets starting with the original, or normal, frame.

Figure 22-2 Original (Normal), 802.1Q, and Double-Tagged Ethernet Packet Formats



When the packet enters the trunk port of the Service Provider egress switch, the metro tag is again stripped as the switch processes the packet. However, the metro tag is not added when the packet is sent out the tunnel port on the edge switch into the customer network. The packet is sent as a normal 802.1Q-tagged frame to preserve the original VLAN numbers in the customer network.

All packets entering the Service Provider network through a tunnel port on an edge switch are treated as untagged packets, whether they are untagged or already tagged with 802.1Q headers. The packets are encapsulated with the metro tag VLAN ID (set to the access VLAN of the tunnel port) when they are sent through the Service Provider network on an 802.1Q trunk port. The priority field on the metro tag is set to the interface class of service (CoS) priority configured on the tunnel port. (The default is zero if none is configured.)

In Figure 22-1, Customer A was assigned VLAN 30, and Customer B was assigned VLAN 40. Packets entering the edge-switch tunnel ports with 802.1Q tags are double-tagged when they enter the Service Provider network, with the metro tag containing VLAN ID 30 or 40, appropriately, and the inner tag containing the original customer VLAN number, for example, VLAN 100. Even if Customers A and B both have VLAN 100 in their networks, the traffic remains segregated within the Service Provider network because the metro tag is different. Each customer controls its own VLAN numbering space, which is independent of the VLAN numbering space used by other customers and the VLAN numbering space used by the Service Provider network.

Configuring 802.1Q Tunneling

These sections describe 802.1Q tunneling configuration:

- 802.1Q Tunneling Configuration Guidelines, page 22-4
- 802.1Q Tunneling and Other Features, page 22-5
- Configuring an 802.1Q Tunneling Port, page 22-6

Note By default, 802.1Q tunneling is disabled because the default switch port mode is dynamic auto. Tagging of 802.1Q native VLAN packets on all 802.1Q trunk ports is also disabled.

802.1Q Tunneling Configuration Guidelines

When you configure 802.1Q tunneling, you should always use asymmetrical links for traffic going through a tunnel and should dedicate one VLAN for each tunnel. You should also be aware of configuration requirements for native VLANs and maximum transmission units (MTUs). For more information about MTUs, see the "System MTU" section on page 22-5.

Native VLANs

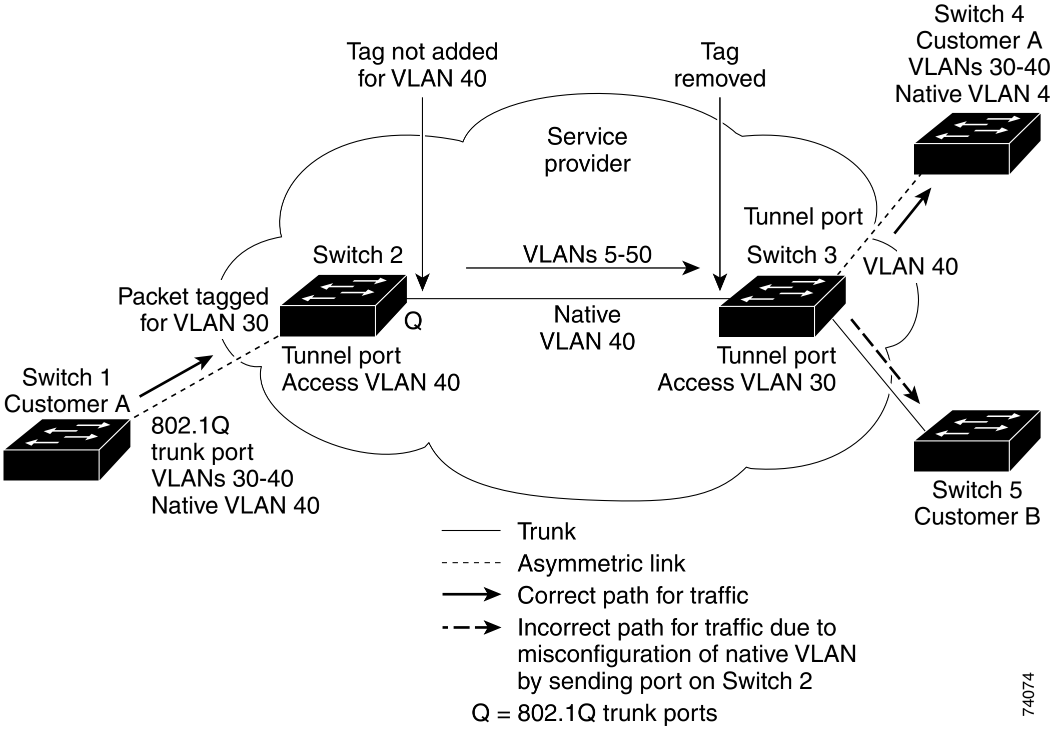
When configuring 802.1Q tunneling on an edge switch, you must use 802.1Q trunk ports for sending packets into the Service Provider network. However, packets going through the core of the Service Provider network can be carried through 802.1Q trunks, ISL trunks, or nontrunking links. When 802.1Q trunks are used in these core switches, the native VLANs of the 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same switch because traffic on the native VLAN would not be tagged on the 802.1Q sending trunk port.

See Figure 22-3. VLAN 40 is configured as the native VLAN for the 802.1Q trunk port from Customer A at the ingress edge switch in the Service Provider network (Switch 2). Switch 1 of Customer A sends a tagged packet on VLAN 30 to the ingress tunnel port of Switch 2 in the Service Provider network, which belongs to access VLAN 40. Because the access VLAN of the tunnel port (VLAN 40) is the same as the native VLAN of the edge-switch trunk port (VLAN 40), the metro tag is not added to tagged packets received from the tunnel port. The packet carries only the VLAN 30 tag through the Service Provider network to the trunk port of the egress-edge switch (Switch 3) and is misdirected through the egress switch tunnel port to Customer B.

These are some ways to solve this problem:

- Use ISL trunks between core switches in the Service Provider network. Although customer interfaces connected to edge switches must be 802.1Q trunks, we recommend using ISL trunks for connecting switches in the core layer.
- Use the **switchport trunk native vlan tag** per-port command and the **vlan dot1q tag native** global configuration command to configure the edge switch so that all packets going out an 802.1Q trunk, including the native VLAN, are tagged. If the switch is configured to tag native VLAN packets on all 802.1Q trunks, the switch ensures that all packets exiting the trunk are tagged and prevents the reception of untagged packets on the trunk port.
- Ensure that the native VLAN ID on the edge-switch trunk port is not within the customer VLAN range. For example, if the trunk port carries traffic of VLANs 100 to 200, assign the native VLAN a number outside that range.

Figure 22-3 Potential Problem with 802.1Q Tunneling and Native VLANs


Tag not added for VLAN 40
Tag removed
Switch 4 Customer A VLANs 30-40 Native VLAN 40
Service provider
Tunnel port
Switch 2
VLANs 5-50
Switch 3
VLAN 40
Packet tagged for VLAN 30
Q
Native VLAN 40
Tunnel port Access VLAN 40
Tunnel port Access VLAN 30
Switch 1 Customer A
802.1Q trunk port VLANs 30-40 Native VLAN 40
Switch 5 Customer B
Trunk
Asymmetric link
Correct path for traffic
Incorrect path for traffic due to misconfiguration of native VLAN by sending port on Switch 2
Q = 802.1Q trunk ports
74074

System MTU

The default system MTU for traffic on the Catalyst 4500 series switch is 1500 bytes. You can configure the switch to support larger frames by using the **system mtu** global configuration command. Because the 802.1Q tunneling feature increases the frame size by 4 bytes when the metro tag is added, you must configure all switches in the Service Provider network to be able to process larger frames by increasing the switch system MTU size to at least 1504 bytes. The maximum allowable system MTU for Catalyst 4500 Gigabit Ethernet switches is 9198 bytes; the maximum system MTU for Fast Ethernet switches is 1552 bytes.

802.1Q Tunneling and Other Features

Although 802.1Q tunneling works well for Layer 2 packet switching, there are incompatibilities between some Layer 2 features and Layer 3 switching.

- A tunnel port cannot be a routed port.
- IP routing is not supported on a VLAN that includes 802.1Q ports. Packets received from a tunnel port are forwarded based only on Layer 2 information. If routing is enabled on a switch virtual interface (SVI) that includes tunnel ports, untagged IP packets received from the tunnel port are recognized and routed by the switch. Customers can access the Internet through the native VLAN. If this access is not needed, you should not configure SVIs on VLANs that include tunnel ports.
- Tunnel ports do not support IP access control lists (ACLs).
- Layer 3 quality of service (QoS) ACLs and other QoS features related to Layer 3 information are not supported on tunnel ports. MAC-based QoS is supported on tunnel ports.

- EtherChannel port groups are compatible with tunnel ports as long as the 802.1Q configuration is consistent within an EtherChannel port group.
- Port Aggregation Protocol (PAgP), Link Aggregation Control Protocol (LACP), and UniDirectional Link Detection (UDLD) are supported on 802.1Q tunnel ports.
- Dynamic Trunking Protocol (DTP) is not compatible with 802.1Q tunneling because you must manually configure asymmetric links with tunnel ports and trunk ports.
- Loopback detection is supported on 802.1Q tunnel ports.
- When a port is configured as an 802.1Q tunnel port, spanning-tree bridge protocol data unit (BPDU) filtering is automatically enabled on the interface. Cisco Discovery Protocol (CDP) is automatically disabled on the interface.

Configuring an 802.1Q Tunneling Port

To configure a port as an 802.1Q tunnel port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode and the interface to be configured as a tunnel port. This should be the edge port in the Service Provider network that connects to the customer switch. Valid interfaces include physical interfaces and port-channel logical interfaces (port channels 1 to 64).
Step 3	`Switch(config-if)# switchport access vlan vlan-id`	Specifies the default VLAN, which is used if the interface stops trunking. This VLAN ID is specific to the particular customer.
Step 4	`Switch(config-if)# switchport mode dot1q-tunnel`	Sets the interface as an 802.1Q tunnel port.
Step 5	`Switch(config-if)# exit`	Returns to global configuration mode.
Step 6	`Switch(config)# vlan dot1q tag native`	(Optional) Sets the switch to enable tagging of native VLAN packets on all 802.1Q trunk ports. When not set, and a customer VLAN ID is the same as the native VLAN, the trunk port does not apply a metro tag, and packets could be sent to the wrong destination.
Step 7	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 8	`Switch# show dot1q-tunnel`	Displays the tunnel ports on the switch.
Step 9	`Switch# show vlan dot1q tag native`	Displays 802.1Q native-VLAN tagging status.
Step 10	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Use the **no vlan dot1q tag native** global command and the **no switchport mode dot1q-tunnel** interface configuration command to return the port to the default state of dynamic auto. Use the **no vlan dot1q tag native** global configuration command to disable tagging of native VLAN packets.

This example shows how to configure an interface as a tunnel port, enable tagging of native VLAN packets, and verify the configuration. In this configuration, the VLAN ID for the customer connected to Gigabit Ethernet interface 2/7 is VLAN 22.

```
Switch(config)# interface gigabitethernet2/7
Switch(config-if)# switchport access vlan 22
% Access VLAN does not exist. Creating vlan 22
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# exit
Switch(config)# vlan dot1q tag native
Switch(config)# end
Switch# show dot1q-tunnel interface gigabitethernet2/7
Port
-----
LAN Port(s)
-----
Gi2/7
Switch# show vlan dot1q tag native
dot1q native vlan tagging is enabled globally
```

Understanding Layer 2 Protocol Tunneling

Note Supervisor Engine 6-E does *not* support Layer 2 Protocol Tunneling.

Customers at different sites connected across a Service Provider network need to use various Layer 2 protocols to scale their topologies to include all remote and local sites. STP must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the Service Provider network. Cisco Discovery Protocol (CDP) must discover neighboring Cisco devices from local and remote sites. VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

When protocol tunneling is enabled, edge switches on the inbound side of the Service Provider network encapsulate Layer 2 protocol packets with a special MAC address and send them across the Service Provider network. Core switches in the network do not process these packets but forward them as normal packets. Layer 2 protocol data units (PDUs) for CDP, STP, or VTP cross the Service Provider network and are delivered to customer switches on the outbound side of the Service Provider network. Identical packets are received by all customer ports on the same VLANs with these results:

- Users on each of a customer's sites can properly run STP, and every VLAN can build a correct spanning tree, based on parameters from all sites and not just from the local site.
- CDP discovers and shows information about the other Cisco devices connected through the Service Provider network.
- VTP provides consistent VLAN configuration throughout the customer network, propagating to all switches through the Service Provider.

Layer 2 protocol tunneling can enabled on trunk, access and tunnel ports. If protocol tunneling is not enabled, remote switches at the receiving end of the Service Provider network do not receive the PDUs and cannot properly run STP, CDP, and VTP. When protocol tunneling is enabled, Layer 2 protocols within each customer's network are totally separate from those running within the Service Provider network.

As an example, Customer A in Figure 22-4 has four switches in the same VLAN that are connected through the Service Provider network. If the network does not tunnel PDUs, switches on the far ends of the network cannot properly run STP, CDP, and VTP. For example, STP for a VLAN on a switch in

Customer A's Site 1 will build a spanning tree on the switches at that site without considering convergence parameters based on Customer A's switch in Site 2. Figure 22-5 shows one possible spanning tree topology.

Figure 22-4 Layer 2 Protocol Tunneling

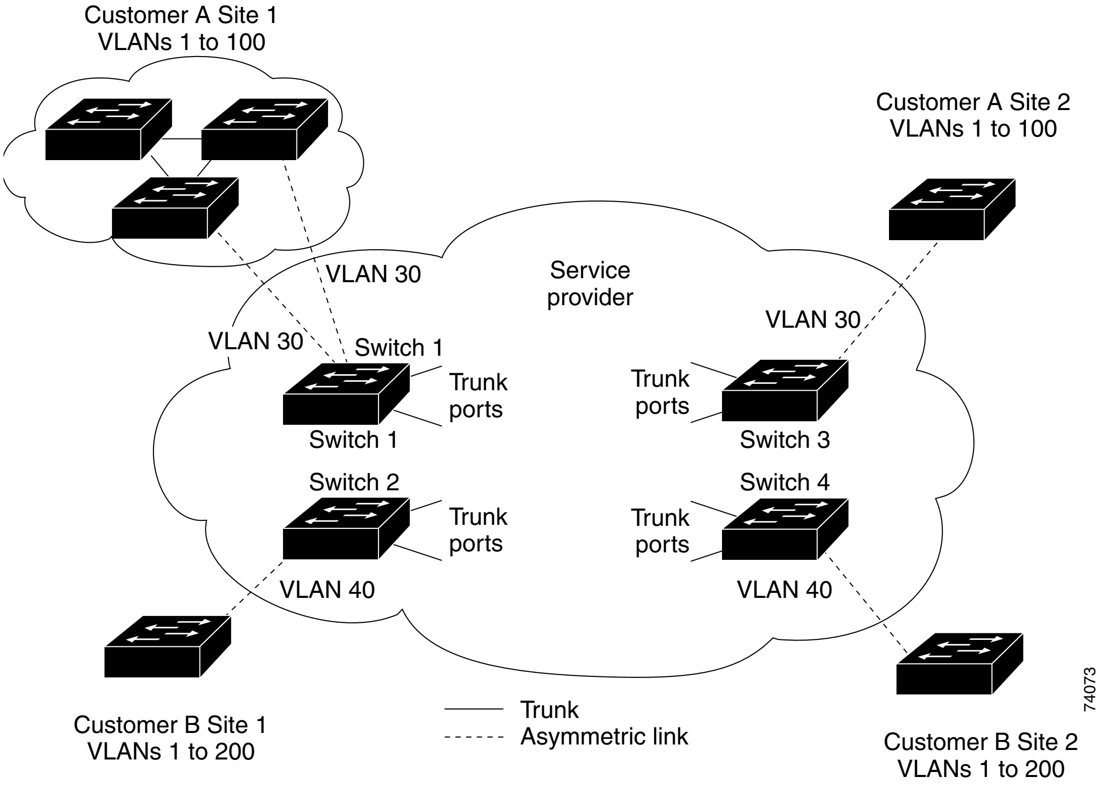
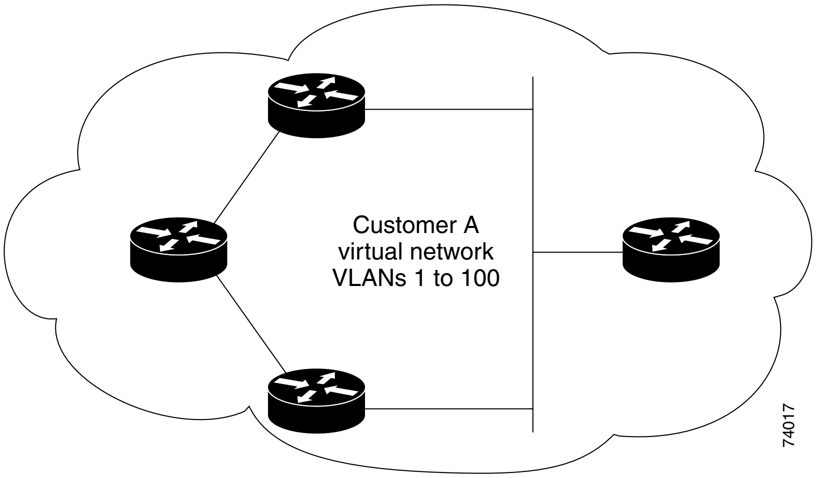


Figure 22-5 Layer 2 Network Topology without Proper Convergence



Configuring Layer 2 Protocol Tunneling

You can enable Layer 2 protocol tunneling (by protocol) on access ports, tunnel ports, or trunk ports that are connected to the customer in the edge switches of the Service Provider network. The Service Provider edge switches connected to the customer switch perform the tunneling process. Edge-switch tunnel ports or normal trunk ports can be connected to customer 802.1Q trunk ports. Edge-switch access ports are connected to customer access ports.

When the Layer 2 PDUs that entered the Service Provider inbound edge switch port exit through the trunk port into the Service Provider network, the switch overwrites the customer PDU-destination MAC address with a well-known Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If 802.1Q tunneling is enabled on the ingress port, packets are also double-tagged. The outer tag is the customer metro tag, and the inner tag is the customer's VLAN tag.

When the Layer 2 PDUs that entered the Service Provider inbound edge switch through the tunnel port or the access port exit through its the trunk port into the Service Provider network, the switch overwrites the customer PDU-destination MAC address with a well-known Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If 802.1Q tunneling is enabled, packets are also double-tagged; the outer tag is the customer metro tag, and the inner tag is the customer’s VLAN tag. The core switches ignore the inner tags and forward the packet to all trunk ports in the same metro VLAN. The edge switches on the outbound side restore the proper Layer 2 protocol and MAC address information and forward the packets to all tunnel or access ports in the same metro VLAN. Therefore, the Layer 2 PDUs remain intact and are delivered across the Service Provider network to the other side of the customer network.

See Figure 22-4, with Customer A and Customer B in access VLANs 30 and 40, respectively. Asymmetric links connect the Customers in Site 1 to edge switches in the Service Provider network. The Layer 2 PDUs (for example, BPDUs) coming into Switch 2 from Customer B in Site 1 are forwarded to the infrastructure as double-tagged packets with the well-known MAC address as the destination MAC address. These double-tagged packets have the metro VLAN tag of 40, as well as an inner VLAN tag (for example, VLAN 100). When the double-tagged packets enter Switch 4, the metro VLAN tag 40 is removed. The well-known MAC address is replaced with the respective Layer 2 protocol MAC address, and the packet is sent to Customer B on Site 2 as a single-tagged frame in VLAN 100.

You can also enable Layer 2 protocol tunneling on access ports on the edge switch connected to access ports on the customer switch. In this case, the encapsulation and de-encapsulation process is the same as described in the previous paragraph, except that the packets are not double-tagged in the Service Provider network. The single tag is the customer-specific access VLAN tag.

This section contains the following subsections:

- Default Layer 2 Protocol Tunneling Configuration, page 22-9
- Layer 2 Protocol Tunneling Configuration Guidelines, page 22-10
- Configuring Layer 2 Tunneling, page 22-11

Default Layer 2 Protocol Tunneling Configuration

Table 22-1 shows the default configuration for Layer 2 protocol tunneling.

Table 22-1 Default Layer 2 Ethernet Interface VLAN Configuration

Feature	Default Setting
Layer 2 protocol tunneling	Disabled.
Shutdown threshold	None set.

Table 22-1 Default Layer 2 Ethernet Interface VLAN Configuration (continued)

Feature	Default Setting
Drop threshold	None set.
CoS value	If a CoS value is configured on the interface for data packets, that value is the default used for Layer 2 PDUs. If none is configured, the default is 5.

Layer 2 Protocol Tunneling Configuration Guidelines

These are some configuration guidelines and operating characteristics of Layer 2 protocol tunneling:

- The switch supports tunneling of CDP, STP, including multiple STP (MSTP), and VTP. Protocol tunneling is disabled by default but can be enabled for the individual protocols on 802.1Q tunnel ports, access ports or trunk ports.
- Dynamic Trunking Protocol (DTP) is not compatible with Layer 2 protocol tunneling because you must manually configure asymmetric links with tunnel ports and trunk ports.
- EtherChannel port groups are compatible with tunnel ports when the 802.1Q configuration is consistent within an EtherChannel port group.
- If an encapsulated PDU (with the proprietary destination MAC address) is received on a port with Layer 2 tunneling enabled, the port is shut down to prevent loops.
- The port also shuts down when a configured shutdown threshold for the protocol is reached. You can manually re-enable the port (by entering a shutdown and a no shutdown command sequence). If errdisable recovery is enabled, the operation is retried after a specified time interval.
- Only decapsulated PDUs are forwarded to the customer network. The spanning-tree instance running on the Service Provider network does not forward BPDUs to Layer 2 protocol tunneling ports. CDP packets are not forwarded from Layer 2 protocol tunneling ports.
- When protocol tunneling is enabled on an interface, you can set a per-protocol, per-port, shutdown threshold for the PDUs generated by the customer network. If the limit is exceeded, the port shuts down. You can also limit the BPDU rate by using QoS ACLs and policy maps on a Layer 2 protocol tunneling port.
- When protocol tunneling is enabled on an interface, you can set a per-protocol, per-port, drop threshold for the PDUs generated by the customer network. If the limit is exceeded, the port drops PDUs until the rate at which it receives them is below the drop threshold.
- Because tunneled PDUs (especially STP BPDUs) must be delivered to all remote sites so that the customer virtual network operates properly, you can give PDUs higher priority within the Service Provider network than data packets received from the same tunnel port. By default, the PDUs use the same CoS value as data packets.

Configuring Layer 2 Tunneling

To configure a port for Layer 2 protocol tunneling, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode, and enter the interface to be configured as a tunnel port. This should be the edge port in the Service Provider network that connects to the customer switch. Valid interfaces can be physical interfaces and port-channel logical interfaces (port channels 1 to 64).

Step 3	`Switch(config-if)#` **`switchport mode access`**
	`or`
	`Switch(config-if)#` **`switchport mode dot1q-tunnel`**
	`or`
	`Switch(config-if)#` **`switchport mode trunk`**
		Configures the interface as an access port, an 802.1Q tunnel port or a trunk port.

Step 4	`Switch(config-if)#` **`l2protocol-tunnel`** `[`**`cdp`** `|` **`stp`** `|` **`vtp`**`]`	Enables protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all three Layer 2 protocols.
Step 5	`Switch(config-if)#` **`l2protocol-tunnel shutdown-threshold`** `[`**`cdp`** `|` **`stp`** `|` **`vtp`**`]` *`value`*	(Optional) Configures the threshold for packets-per-second accepted for encapsulation. The interface is disabled if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured. **Note** If you also set a drop threshold on this interface, the shutdown-threshold value must be greater than or equal to the drop-threshold value.
Step 6	`Switch(config-if)#` **`l2protocol-tunnel drop-threshold`** `[`**`cdp`** `|` **`stp`** `|` **`vtp`**`]` *`value`*	(Optional) Configures the threshold for packets-per-second accepted for encapsulation. The interface drops packets if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured. **Note** If you also set a shutdown threshold on this interface, the drop-threshold value must be less than or equal to the shutdown-threshold value.
Step 7	`Switch(config-if)#` **`exit`**	Returns to global configuration mode.
Step 8	`Switch(config)#` **`errdisable recovery cause l2ptguard`**	(Optional) Configures the recovery mechanism from a Layer 2 maximum-rate error so that the interface is re-enabled and can try again. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.
Step 9	`Switch(config)#` **`l2protocol-tunnel cos`** *`value`*	(Optional) Configures the CoS value for all tunneled Layer 2 PDUs. The range is 0 to 7; the default is the default CoS value for the interface. If none is configured, the default is 5.
Step 10	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.

	Command	Purpose
Step 11	`Switch# show l2protocol`	Displays the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 12	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Use the **no l2protocol-tunnel** [**cdp** | **stp** | **vtp**] interface configuration command to disable protocol tunneling for one of the Layer 2 protocols or for all three. Use the **no l2protocol-tunnel shutdown-threshold** [**cdp** | **stp** | **vtp**] and the **no l2protocol-tunnel drop-threshold** [**cdp** | **stp** | **vtp**] commands to return the shutdown and drop thresholds to the default settings.

This example shows how to configure Layer 2 protocol tunneling on an 802.1Q tunnel port for CDP, STP, and VTP and how to verify the configuration:

```
Switch(config)# interface FastEthernet2/1
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# l2protocol-tunnel cdp
Switch(config-if)# l2protocol-tunnel stp
Switch(config-if)# l2protocol-tunnel vtp
Switch(config-if)# l2protocol-tunnel shutdown-threshold 1500
Switch(config-if)# l2protocol-tunnel drop-threshold 1000
Switch(config-if)# exit
Switch(config)# l2protocol-tunnel cos 7
Switch(config)# end
Switch# show l2protocol
COS for Encapsulated Packets: 7
Port    Protocol Shutdown  Drop      Encapsulation Decapsulation Drop
                 Threshold Threshold Counter       Counter       Counter
------- -------- --------- --------- ------------- ------------- -------------
Fa2/11  cdp           1500      1000 2288          2282          0
        stp           1500      1000 116           13            0
        vtp           1500      1000 3             67            0
```

Monitoring and Maintaining Tunneling Status

Table 22-2 shows the commands for monitoring and maintaining 802.1Q and Layer 2 protocol tunneling.

Table 22-2 Commands for Monitoring and Maintaining Tunneling

Command	Purpose
`Switch# clear l2protocol-tunnel counters`	Clears the protocol counters on Layer 2 protocol tunneling ports.
`Switch# show dot1q-tunnel`	Displays 802.1Q tunnel ports on the switch.
`Switch# show dot1q-tunnel interface` *`interface-id`*	Verifies if a specific interface is a tunnel port.
`Switch# show l2protocol-tunnel`	Displays information about Layer 2 protocol tunneling ports.
`Switch# show errdisable recovery`	Verifies if the recovery timer from a Layer 2 protocol-tunnel error disable state is enabled.
`Switch# show l2protocol-tunnel interface` *`interface-id`*	Displays information about a specific Layer 2 protocol tunneling port.
`Switch# show l2protocol-tunnel summary`	Displays only Layer 2 protocol summary information.
`Switch# show vlan dot1q native`	Displays the status of native VLAN tagging on the switch.

Note With Cisco IOS Release 12.2(20)EW, the BPDU filtering configuration for both dot1q and Layer 2 protocol tunneling is no longer visible in the running configuration as "spanning-tree bpdufilter enable.” Instead, it is visible in the output of the **show spanning tree int detail** command as shown below.

```
Switch# show spann int f6/1 detail
 Port 321 (FastEthernet6/1) of VLAN0001 is listening
   Port path cost 19, Port priority 128, Port Identifier 128.321.
   Designated root has priority 32768, address 0008.e341.4600
   Designated bridge has priority 32768, address 0008.e341.4600
   Designated port id is 128.321, designated path cost 0
   Timers: message age 0, forward delay 2, hold 0
   Number of transitions to forwarding state: 0
   Link type is point-to-point by default
   ** Bpdu filter is enabled internally **
   BPDU: sent 0, received 0
```

CHAPTER 23

Configuring CDP

This chapter describes how to configure Cisco Discovery Protocol (CDP) on the Catalyst 4500 series switch. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- Overview of CDP, page 23-1
- Configuring CDP, page 23-2

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cr/hcf_r/index.htm

and the *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.4:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124tcr/tcf_r/index.htm

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of CDP

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. CDP 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

The following sections describe how to configure CDP:

- Enabling CDP Globally, page 23-2
- Displaying the CDP Global Configuration, page 23-2
- Enabling CDP on an Interface, page 23-3
- Displaying the CDP Interface Configuration, page 23-3
- Monitoring and Maintaining CDP, page 23-3

Enabling CDP Globally

To enable CDP globally, perform this task:

Command	Purpose
`Switch(config)# [no] cdp run`	Enables CDP globally. Use the **no** keyword to disable CDP globally.

This example shows how to enable CDP globally:

```
Switch(config)# cdp run
```

Displaying the CDP Global Configuration

To display the CDP configuration, perform this task:

Command	Purpose
`Switch# show cdp`	Displays global CDP information.

This example shows how to display the CDP configuration:

```
Switch# show cdp
Global CDP information:
        Sending CDP packets every 120 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is enabled
Switch#
```

For additional CDP **show** commands, see the “Monitoring and Maintaining CDP” section on page 23-3.

Enabling CDP on an Interface

To enable CDP on an interface, perform this task:

Command	Purpose
`Switch(config-if)# [no] cdp enable`	Enables CDP on an interface. Use the **no** keyword to disable CDP on an interface.

This example shows how to enable CDP on Fast Ethernet interface 5/1:

```
Switch(config)# interface fastethernet 5/1
Switch(config-if)# cdp enable
```

This example shows how to disable CDP on Fast Ethernet interface 5/1:

```
Switch(config)# interface fastethernet 5/1
Switch(config-if)# no cdp enable
```

Displaying the CDP Interface Configuration

To display the CDP configuration for an interface, perform this task:

Command	Purpose
`Switch# show cdp interface [type/number]`	Displays information about interfaces where CDP is enabled.

This example shows how to display the CDP configuration of Fast Ethernet interface 5/1:

```
Switch# 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
Switch#
```

Monitoring and Maintaining CDP

To monitor and maintain CDP on your device, perform one or more of these tasks:

Command	Purpose
`Switch# clear cdp counters`	Resets the traffic counters to zero.
`Switch# clear cdp table`	Deletes the CDP table of information about neighbors.
`Switch# show cdp`	Displays global information such as frequency of transmissions and the holdtime for packets being transmitted.

Command	Purpose
Switch# **show cdp entry** *entry_name* [**protocol** | **version**]	Displays information about a specific neighbor. The display can be limited to protocol or version information.
Switch# **show cdp interface** [*type*/*number*]	Displays information about interfaces on which CDP is enabled.
Switch# **show cdp neighbors** [*type*/*number*] [**detail**]	Displays information about neighboring equipment. The display can be limited to neighbors on a specific interface and expanded to provide more detailed information.
Switch# **show cdp traffic**	Displays CDP counters, including the number of packets sent and received and checksum errors.
Switch# **show debugging**	Displays information about the types of debugging that are enabled for your switch.

This example shows how to clear the CDP counter configuration on your switch:

```
Switch# clear cdp counters
```

This example shows how to display information about the neighboring equipment:

```
Switch# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
JAB023807H1      Fas 5/3            127         T S       WS-C2948  2/46
JAB023807H1      Fas 5/2            127         T S       WS-C2948  2/45
JAB023807H1      Fas 5/1            127         T S       WS-C2948  2/44
JAB023807H1      Gig 1/2            122         T S       WS-C2948  2/50
JAB023807H1      Gig 1/1            122         T S       WS-C2948  2/49
JAB03130104      Fas 5/8            167         T S       WS-C4003  2/47
JAB03130104      Fas 5/9            152         T S       WS-C4003  2/48
```

CHAPTER 24

Configuring UDLD

This chapter describes how to configure the UniDirectional Link Detection (UDLD) and Unidirectional Ethernet on the Catalyst 4000 family switch. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- Overview of UDLD, page 24-1
- Default UDLD Configuration, page 24-2
- Configuring UDLD on the Switch, page 24-2

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of UDLD

UDLD allows devices connected through fiber-optic or copper Ethernet cables (for example, Category 5 cabling) to monitor the physical configuration of the cables and detect when a unidirectional link exists. 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. When a unidirectional link is detected, UDLD shuts down the affected interface 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 mechanisms 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 interfaces. 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.

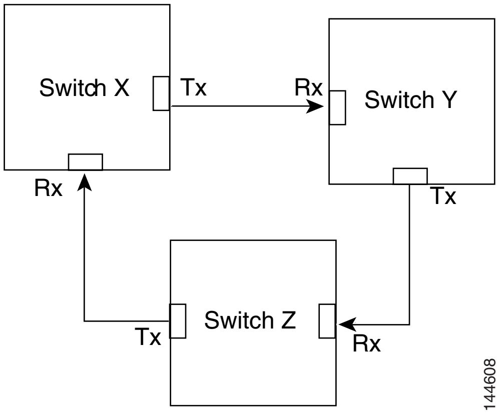
If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally from a Layer 1 perspective, then UDLD at Layer 2 determines whether or not those fibers are connected correctly and whether or not traffic is flowing bidirectionally between the right neighbors. This check cannot be performed by autonegotiation because autonegotiation operates at Layer 1.

The switch periodically transmits UDLD packets to neighbor devices on interfaces 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 interface 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 interfaces to avoid sending unnecessary control traffic on this type of media, since it is often used for access interfaces.

Figure 24-1 shows an example of a unidirectional link condition. Each switch can send packets to a neighbor switch but is not able to receive packets from the same switch that it is sending packets to. UDLD detects and disables these one-way connections.

Figure 24-1 Unidirectional Link



Default UDLD Configuration

Table 24-1 shows the UDLD default configuration.

Table 24-1 UDLD Default Configuration

Feature	Default Status
UDLD global enable state	Globally disabled
UDLD per-interface enable state for fiber-optic media	Enabled on all Ethernet fiber-optic interfaces
UDLD per-interface enable state for twisted-pair (copper) media	Disabled on all Ethernet 10/100 and 1000BaseTX interfaces

Configuring UDLD on the Switch

The following sections describe how to configure UDLD:

- Enabling UDLD Globally, page 24-3
- Enabling UDLD on Individual Interfaces, page 24-3

- Disabling UDLD on Non-Fiber-Optic Interfaces, page 24-3
- Disabling UDLD on Fiber-Optic Interfaces, page 24-4
- Resetting Disabled Interfaces, page 24-4

Enabling UDLD Globally

To enable UDLD globally on all fiber-optic interfaces on the switch, perform this task:

Command	Purpose
`Switch(config)# [no] udld enable`	Enables UDLD globally on fiber-optic interfaces on the switch. Use the **no** keyword to globally disable UDLD on fiber-optic interfaces. **Note** This command configures only fiber-optic interfaces. An individual interface configuration overrides the setting of this command.

Enabling UDLD on Individual Interfaces

To enable UDLD on individual interfaces, perform this task:

	Command	Purpose
Step 1	`Switch(config-if)# udld enable`	Enables UDLD on a specific interface. On a fiber-optic interface, this command overrides the **udld enable** global configuration command setting.
Step 2	`Switch# show udld interface`	Verifies the configuration.

Disabling UDLD on Non-Fiber-Optic Interfaces

To disable UDLD on individual non-fiber-optic interfaces, perform this task:

	Command	Purpose
Step 1	`Switch(config-if)# no udld enable`	Disables UDLD on a non-fiber-optic interface. **Note** On fiber-optic interfaces, the **no udld enable** command reverts the interface configuration to the **udld enable** global configuration command setting.
Step 2	`Switch# show udld interface`	Verifies the configuration.

Disabling UDLD on Fiber-Optic Interfaces

To disable UDLD on individual fiber-optic interfaces, perform this task:

	Command	Purpose
Step 1	`Switch(config-if)# udld disable`	Disables UDLD on a fiber-optic interface. **Note** This command is not supported on nonfiber-optic interfaces. Use the **no** keyword to revert to the **udld enable** global configuration command setting.
Step 2	`Switch# show udld interface`	Verifies the configuration.

Resetting Disabled Interfaces

To reset all interfaces that have been shut down by UDLD, perform this task:

Command	Purpose
`Switch# udld reset`	Resets all interfaces that have been shut down by UDLD.

CHAPTER 25

Configuring Unidirectional Ethernet

Note Supervisor Engine 6-E does *not* support this feature.

This chapter describes how to configure Unidirectional Ethernet on the Catalyst 4000 family switch and contains these sections:

- Overview of Unidirectional Ethernet, page 25-1
- Configuring Unidirectional Ethernet, page 25-1

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of Unidirectional Ethernet

You can set stubless Gigabit Ethernet ports to unidirectionally transmit or receive traffic. Unidirectional Ethernet uses only one strand of fiber for either transmitting or receiving one-way traffic for the GigaPort, instead of two strands of fiber for a full-duplex GigaPort Ethernet. Configuring your GigaPorts either to transmit or receive traffic effectively doubles the amount of traffic capabilities for applications, such as video streaming, where most traffic is sent as unacknowledged unidirectional video broadcast streams.

Configuring Unidirectional Ethernet

Note You must configure Unidirectional Ethernet on the non-blocking GigaPort, which will automatically disable UDLD on the port.

To enable Unidirectional Ethernet, perform this task:

	Command	Purpose
Step 1	Switch(config)# **interface** {**vlan** *vlan_ID* | {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/interface* | **Port-channel** *number*}	Selects the interface to configure.
Step 2	Switch(config-if)# [**no**] **unidirectional** {**send-only** | **receive-only**}	Enables Unidirectional Ethernet. Use the **no** keyword to disable Unidirectional Ethernet.
Step 3	Switch(config-if)# **end**	Exits configuration mode.
Step 4	Switch# **show interface** {**vlan** *vlan_ID* | {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/interface*} **unidirectional**	Verifies the configuration.

This example shows how to set Gigabit Ethernet interface 1/1 to unidirectionally send traffic:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# unidirectional send-only
Switch(config-if)# end

Warning!

Enable l2 port unidirectional mode will automatically disable port udld.
You must manually ensure that the unidirectional link does not create
a spanning tree loop in the network.

Enable l3 port unidirectional mode will automatically disable ip routing
on the port. You must manually configure static ip route and arp entry
in order to route ip traffic.
```

This example shows how to set Gigabit Ethernet interface 1/1 to receive traffic unidirectionally:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# unidrectional receive-only
Switch(config-if)# end

Warning!

Enable l2 port unidirectional mode will automatically disable port udld.
You must manually ensure that the unidirectional link does not create
a spanning tree loop in the network.

Enable l3 port unidirectional mode will automatically disable ip routing
on the port. You must manually configure static ip route and arp entry
in order to route ip traffic.
```

This example shows how to verify the configuration

```
Switch>show interface gigabitethernet 1/1 unidirectional
  show interface gigabitethernet 1/1 unidirectional
  Unidirectional configuration mode: send only
  CDP neighbour unidirectional configuration mode: receive only
```

This example shows how to disable Unidirectional Ethernet on Gigabit Ethernet interface 1/1:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# no unidirectional
Switch(config-if)# end
```

This example shows the result of issuing the **show interface** command for a port that does not support Unidirectional Ethernet:

```
Switch#show interface f6/1 unidirectional
Unidirectional Ethernet is not supported on FastEthernet6/1
```

CHAPTER 26

Configuring Layer 3 Interfaces

This chapter describes the Layer 3 interfaces on a Catalyst 4500 series switch. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of Layer 3 Interfaces

This section contains the following subsections:

The Catalyst 4500 family switch supports Layer 3 interfaces with the Cisco IOS IP and IP routing protocols. Layer 3, the *network* layer, is primarily responsible for the routing of data in packets across logical internetwork paths.

Layer 2, the *data link* layer, contains the protocols that control the *physical* layer (Layer 1) and how data is framed before being transmitted on the medium. The Layer 2 function of filtering and forwarding data in frames between two segments on a LAN is known as *bridging*.

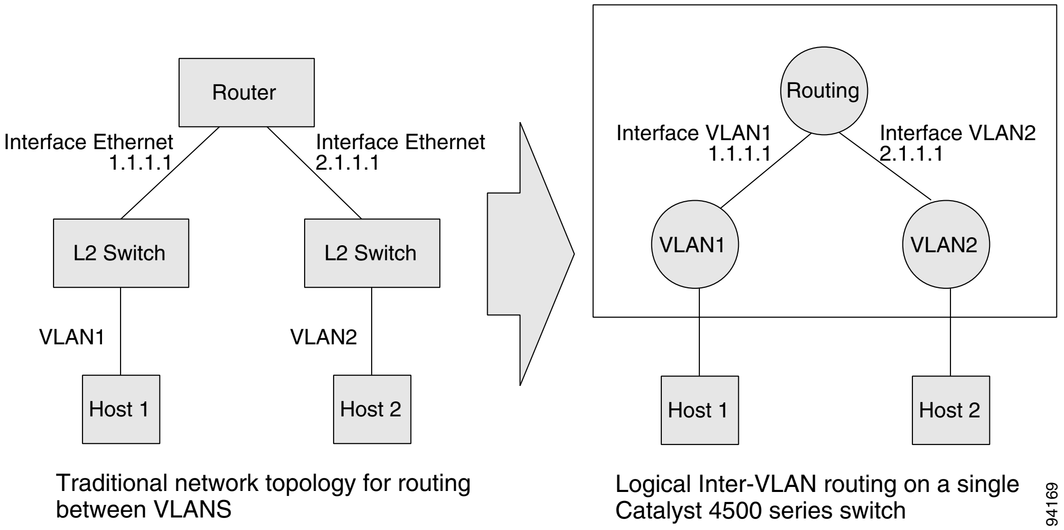
The Catalyst 4500 series switch supports two types of Layer 3 interfaces. The logical Layer 3 VLAN interfaces integrate the functions of routing and bridging. The physical Layer 3 interfaces allow the Catalyst 4500 series switch to be configured like a traditional router.

Logical Layer 3 VLAN Interfaces

The logical Layer 3 VLAN interfaces provide logical routing interfaces to VLANs on Layer 2 switches. A traditional network requires a physical interface from a router to a switch to perform inter-VLAN routing. The Catalyst 4500 series switch supports inter-VLAN routing by integrating the routing and bridging functions on a single Catalyst 4500 series switch.

Figure 26-1 shows how the routing and bridging functions in the three physical devices of the traditional network are performed logically on one Catalyst 4500 series switch.

Figure 26-1 Logical Layer 3 VLAN Interfaces for the Catalyst 4500 Series Switch

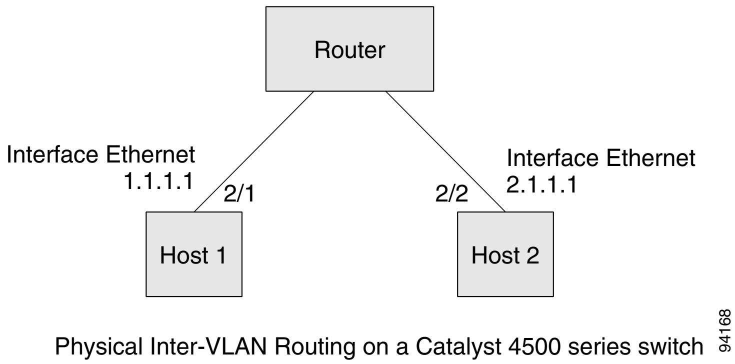

Router
Interface Ethernet 1.1.1.1
Interface Ethernet 2.1.1.1
L2 Switch
L2 Switch
VLAN1
VLAN2
Host 1
Host 2
Traditional network topology for routing between VLANS
Routing
Interface VLAN1 1.1.1.1
Interface VLAN2 2.1.1.1
VLAN1
VLAN2
Host 1
Host 2
Logical Inter-VLAN routing on a single Catalyst 4500 series switch
94169

Physical Layer 3 Interfaces

The physical Layer 3 interfaces support capabilities equivalent to a traditional router. These Layer 3 interfaces provide hosts with physical routing interfaces to a Catalyst 4500 series switch.

Figure 26-2 shows how the Catalyst 4500 series switch functions as a traditional router.

Figure 26-2 Physical Layer 3 Interfaces for the Catalyst 4500 Series Switch



Understanding SVI Autostate Exclude

Note Supervisor Engine 6-E does *not* support SVI Autostate Exclude.

To be "up/up," a router VLAN interface must fulfill the following general conditions:

- The VLAN exists and is "active" on the VLAN database of the switch.
- The VLAN interface exists on the router and is not administratively down.
- At least one Layer 2 (access port or trunk) port exists, has a link "up" on this VLAN and is in spanning-tree forwarding state on the VLAN.

Note The protocol line state for the VLAN interfaces will come up when the first switchport belonging to the corresponding VLAN link comes up and is in spanning-tree forwarding state.

Ordinarily, when a VLAN interface has multiple ports in the VLAN, the SVI will go "down" when all the ports in the VLAN go "down." The SVI Autostate exclude feature provides a knob to mark a port so that it is not counted in the SVI "up and down" calculation and applies to all VLANs that are enabled on that port.

A VLAN interface will be brought up after the Layer 2 port has had time to converge (that is, transition from listening-learning to forwarding). This will prevent routing protocols and other features from using the VLAN interface as if it were fully operational. This also prevents other problems from occurring, such as routing black holes.

Understanding Layer 3 Interface Counters

Note Supervisor Engine 6-E does *not* support Layer 2 Interface Counters. However, it does support Layer 3 (SVI) Interface Counters.

With Supervisor Engine 6-E, IPv4 and v6 packets are routed in hardware by the forwarding engine. This engine supports statistics for counting routed packets for up to 4095 interfaces. These statistics include:

- Input unicast
- Input multicast
- Output unicast
- Output multicast

For each type of counter, both the number of packets and the total number of bytes received or transmitted are counted.

Because the total number of counters supported is lower than the total number of Layer 3 interfaces supported, it is not always possible for all Layer 3 interfaces to have counters. For this reason, the user assigns counters to Layer 3 interfaces, and the default configuration for a Layer 3 interface will have no counters.

Note To enable Layer 3 Interface Counters, you need to issue the **counter** command in interface mode. Refer to the "Configuring Layer 3 Interface Counters" section on page 26-10 for instructions on how to configure Layer 3 Interface Counters.

These hardware counters are displayed in the output of the **show interface** command. For example:

```
Switch# show interface vlan 1
Vlan1 is up, line protocol is up
  Hardware is Ethernet SVI, address is 0005.9a38.6cff (bia 0005.9a38.6cff)
  Internet address is 10.0.0.1/8
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes     <====
  L3 out Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes    <====
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     1 packets output, 46 bytes, 0 underruns
     0 output errors, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Configuration Guidelines

The Catalyst 4500 series switch supports AppleTalk routing and IPX routing. For AppleTalk routing and IPX routing information, refer to “Configuring AppleTalk” and “Configuring Novell IPX” in the *Cisco IOS AppleTalk and Novell IPX C onfiguration Guide* at the following URL:

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

Note Supervisor Engine 6-E does *not* support AppleTalk and IPX routing.

A Catalyst 4500 series switch does not support subinterfaces or the **encapsulation** keyword on Layer 3 Fast Ethernet, Gigabit Ethernet, TenGigabitEthernet interfaces.

Note As with any Layer 3 interface running Cisco IOS software, the IP address and network assigned to an SVI cannot overlap those assigned to any other Layer 3 interface on the switch.

Configuring Logical Layer 3 VLAN Interfaces

Note Before you can configure logical Layer 3 VLAN interfaces, you must create and configure the VLANs on the switch, assign VLAN membership to the Layer 2 interfaces, enable IP routing if IP routing is disabled, and specify an IP routing protocol.

To configure logical Layer 3 VLAN interfaces, perform this task:

	Command	Purpose
Step 1	Switch(config)# **vlan** *vlan_ID*	Creates the VLAN.
Step 2	Switch(config)# **interface vlan** *vlan_ID*	Selects an interface to configure.
Step 3	Switch(config-if)# **ip address** ip_address *subnet_mask*	Configures the IP address and IP subnet.
Step 4	Switch(config-if)# **no shutdown**	Enables the interface.
Step 5	Switch(config-if)# **end**	Exits configuration mode.
Step 6	Switch# **copy running-config startup-config**	Saves your configuration changes to NVRAM.
Step 7	Switch# **show interfaces** [*type slot/interface*] Switch# **show ip interfaces** [*type slot/interface*] Switch# **show running-config interfaces** [*type slot/interface*] Switch# **show running-config interfaces vlan** vlan_ID	Verifies the configuration.

This example shows how to configure the logical Layer 3 VLAN interface vlan 2 and assign an IP address:

```
Switch> enable
Switch# config term
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# vlan 2
Switch(config)# interface vlan 2
Switch(config-if)# ip address 10.1.1.1 255.255.255.248
Switch(config-if)# no shutdown
Switch(config-if)# end
```

This example uses the **show interfaces** command to display the interface IP address configuration and status of Layer 3 VLAN interface vlan 2:

```
Switch# show interfaces vlan 2
Vlan2 is up, line protocol is down
  Hardware is Ethernet SVI, address is 00D.588F.B604 (bia 00D.588F.B604)
  Internet address is 172.20.52.106/29
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 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 packets output, 0 bytes, 0 underruns
     0 output errors, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
Switch#
```

This example uses the **show running-config** command to display the interface IP address configuration of Layer 3 VLAN interface vlan 2:

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

Current configuration : !
interface Vlan2
 ip address 10.1.1.1 255.255.255.248
!
ip classless
no ip http server
!
!
line con 0
line aux 0
line vty 0 4
!
end
```

Configuring VLANs as Layer 3 Interfaces

This section consists of the following subsections:

- Configuring SVI Autostate Exclude, page 26-7
- Configuring IP MTU Sizes, page 26-8
- Configuring Layer 3 Interface Counters, page 26-10

Configuring SVI Autostate Exclude

Note Supervisor Engine 6-E does *not* support the SVI Autostate Exclude feature.

Note The SVI Autostate exclude feature is enabled by default and is synchronized with the STP state.

The SVI Autostate exclude feature shuts down (or brings up) the Layer 3 interfaces of a switch when the following port configuration changes occur:

- When the last port on a VLAN goes down, the Layer 3 interface on that VLAN is shut down (SVI- autostated).
- When the first port on the VLAN is brought back up, the Layer 3 interface on the VLAN that was previously shut down is brought up.

SVI Autostate exclude enables you to exclude the access ports/trunks in defining the status of the SVI (up or down) even if it belongs to the same VLAN. Moreover, even if the excluded access port/trunk is in up state and other ports are in down state in the VLAN, the SVI state is changed to down.

At least one port in the VLAN should be up and not excluded to make the SVI state “up.” This will help to exclude the monitoring port status when you are determining the status of the SVI.

To apply SVI Autostate exclude, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode.
Step 3	`Switch(config-if)# switchport autostate exclude`	Exclude the access ports/trunks in defining the status of an SVI (up or down).
Step 4	`Switch(config)# end`	Exits configuration mode.
Step 5	`Switch# show run int g3/4`	Displays the running configuration.
Step 6	`Switch# show int g3/4 switchport`	Verifies the configuration.

The following example shows how to apply SVI Autostate Exclude on interface g3/1:

```
Switch# conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface g3/1
Switch(config-if)# switchport autostate exclude
Switch(config-if)# end
Switch# show run int g3/4
Building configuration...

Current configuration : 162 bytes
!
interface GigabitEthernet3/4
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2,3
 switchport autostate exclude                                  <=====
 switchport mode trunk
end

Switch# show int g3/4 switchport
Name: Gi3/4
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On Access Mode VLAN: 1 (default) Trunking Native Mode VLAN: 1
(default) Administrative Native VLAN tagging: enabled Voice VLAN: none Administrative
private-vlan host-association: none Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none Administrative private-vlan trunk
Native VLAN tagging: enabled Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none Administrative private-vlan trunk
associations: none Administrative private-vlan trunk mappings: none Operational
private-vlan: none Trunking VLANs Enabled: 2,3 Pruning VLANs Enabled: 2-1001 Capture Mode
Disabled Capture VLANs Allowed: ALL
Autostate mode exclude                                     <======

Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
Switch#
```

For information on troubleshooting SVI autostate exclude, refer to the “Troubleshooting SVI Autostate Exclude” section on page 51-49.

Configuring IP MTU Sizes

You can set the protocol-specific maximum transmission unit (MTU) size of IPv4 or IPv6 packets sent on an interface.

For information on MTU limitations, refer to Understanding Maximum Transmission Units, page 6-16.

Note To set the nonprotocol-specific MTU value for an interface, use the **mtu** interface configuration command. Changing the MTU value (with the **mtu** interface configuration command) can affect the IP MTU value. If the current IP MTU value matches the MTU value, and you change the MTU value, the IP MTU value will be modified automatically to match the new MTU. However, the reverse is not true; changing the IP MTU value has no effect on the value for the **mtu** command.

For information on how to configure MTU size, refer to Configuring MTU Sizes, page 6-18.

To set the protocol-specific maximum transmission unit (MTU) size of IPv4 or IPv6 packets sent on an interface, perform the following task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode.

Step 3	`Switch(config-if)# [`**`no`**`]` **`ip mtu`** `<`*`mtu_size`*`>`
	or
	`Switch(config-if)# [`**`no`**`]` **`ipv6 mtu`** `<`*`mtu_size`*`>`
		Configures the IPv4 MTU size
		Configures the IPv6 MTU size.
		The no form of the command reverts to the default MTU size (1500 bytes).

Step 4	`Switch(config-if)#` **`exit`**	Exits configuration interface mode.
Step 5	`Switch(config)#` **`end`**	Exits configuration mode.
Step 6	`Switch#` **`show run interface`** *`interface-id`*	Displays the running configuration.

The following example shows how to configure IPv4 MTU on an interface:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# ip mtu 68
Switch(config-if)# exit
Switch(config)# end
Switch# show ip interface vlan 1
Vlan1 is up, line protocol is up
  Internet address is 10.10.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 68 bytes
  Helper address is not set
.........................(continued)
```

```
The following example shows how to configure ipv6 mtu on an interface
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# ipv6 mtu 1280
Switch(config)# end
Switch# show ipv6 nterface vlan 1

This example shows how to verify the configuration
Vlan1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::214:6AFF:FEBC:DEEA
  Global unicast address(es):
    1001::1, subnet is 1001::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:1
    FF02::1:FFBC:DEEA
  MTU is 1280 bytes
...................(continued)
```

Note When ipv6 is enabled on an interface via any CLI, it is possible to see the following message: `% Hardware MTU table exhausted` In such a scenario, the ipv6 MTU value programmed in hardware will differ from the ipv6 interface MTU value. This will happen if there is no room in the hardware MTU table to store additional values. You must free up some space in the table by unconfiguring some unused MTU values and subsequently disable/re-enable ipv6 on the interface or reaply the MTU configuration.

Configuring Layer 3 Interface Counters

Note Supervisor Engine 6-E does *not* support interface counters.

Note When a linecard is *removed*, the Layer 3 counters previously enabled on ports of that line card are unconfigured. This means that to re-enable Layer 3 counters upon re-insertion of the linecard, you need to reconfigure the counter CLI for Layer 3 ports of that line card.

To configure Layer 3 Interface Counters (assign counters to a Layer 3 interface), perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode.
Step 3	`Switch(config-if)#` **`counter`**	Enables counters.
Step 4	`Switch(config)#` **`end`**	Exits configuration mode.
Step 5	`Switch#` **`show run interface`** *`interface-id`*	Displays the running configuration.

The following example shows how to enable counters on interface VLAN 1

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# counter
Switch(config-if)# end
Switch#
00:17:15: %SYS-5-CONFIG_I: Configured from console by console
Switch# show run interface vlan 1
Building configuration...

Current configuration : 63 bytes
!
interface Vlan1
 ip address 10.0.0.1 255.0.0.0
 counter
end
```

Note To remove the counters, use the "no" form of the **counter** command.

If the maximum number of counters has already been assigned, the **counter** command fails and displays an error message:

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fa3/2
Switch(config-if)# no switchport
Switch(config-if)# counter
 Counter resource exhausted
Switch(config-if)# end
Switch#
00:24:18: %SYS-5-CONFIG_I: Configured from console by console
```

In this situation, you must release a counter from another interface for use by the new interface.

Configuring Physical Layer 3 Interfaces

Note Before you can configure physical Layer 3 interfaces, you must enable IP routing if IP routing is disabled, and specify an IP routing protocol.

To configure physical Layer 3 interfaces, perform this task:

	Command	Purpose
Step 1	`Switch(config)#`**`ip routing`**	Enables IP routing (Required only if disabled.)
Step 2	`Switch(config)#` **`interface`** `{`**`fastethernet`** `|` **`gigabitethernet`** `|` **`tengigabitethernet`**`}` *`slot/port`*`}` `|` `{`**`port-channel`** *`port_channel_number`*`}`	Selects an interface to configure.
Step 3	`Switch(config-if)#`**`no switchport`**	Converts this port from physical Layer 2 port to physical Layer 3 port.
Step 4	`Switch(config-if)#` **`ip address`** `ip_address` *`subnet_mask`*	Configures the IP address and IP subnet.
Step 5	`Switch(config-if)#` **`no shutdown`**	Enables the interface.
Step 6	`Switch(config-if)#` **`end`**	Exits configuration mode.
Step 7	`Switch#` **`copy running-config startup-config`**	Saves your configuration changes to NVRAM.

Step 8	`Switch#` **`show interfaces`** `[`*`type slot/interface`*`]`
	`Switch#` **`show ip interfaces`** `[`*`type slot/interface`*`]`
	`Switch#` **`show running-config interfaces`** `[`*`type slot/interface`*`]`
		Verifies the configuration.

This example shows how to configure an IP address on Fast Ethernet interface 2/1:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface fastethernet 2/1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.1.1.1 255.255.255.248
Switch(config-if)# no shutdown
Switch(config-if)# end
Switch#
```

This example uses the **show running-config** command to display the interface IP address configuration of Fast Ethernet interface 2/1:

```
Switch# show running-config
Building configuration...
!
interface FastEthernet2/1
 no switchport
 ip address 10.1.1.1 255.255.255.248
!
…
ip classless
no ip http server
!
!
line con 0
line aux 0
line vty 0 4
!
end
```

Configuring EIGRP Stub Routing

This section consists of the following subsections:

- Overview, page 26-12
- How to Configure EIGRP Stub Routing, page 26-13
- Monitoring and Maintaining EIGRP, page 26-18
- EIGRP Configuration Examples, page 26-19

Overview

The EIGRP stub routing feature, available in all images, reduces resource utilization by moving routed traffic closer to the end user.

The IP base image contains only EIGRP stub routing. The IP services image contains complete EIGRP routing.

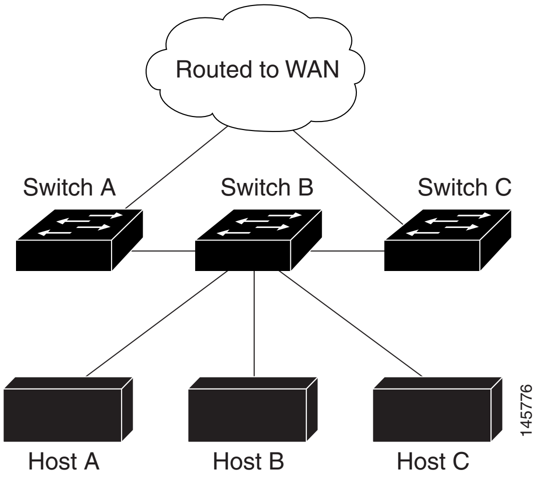
In a network using EIGRP stub routing, the only route for IP traffic to follow to the user is through a switch that is configured with EIGRP stub routing. The switch sends the routed traffic to interfaces that are configured as user interfaces or are connected to other devices.

When using EIGRP stub routing, you need to configure the distribution and remote routers to use EIGRP, and to configure only the switch as a stub. Only specified routes are propagated from the switch. The switch responds to all queries for summaries, connected routes, and routing updates.

Any neighbor that receives a packet informing it of the stub status does not query the stub router for any routes, and a router that has a stub peer does not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

In Figure 26-3, switch B is configured as an EIGRP stub router. Switches A and C are connected to the rest of the WAN. Switch B advertises connected, static, redistribution, and summary routes from switch A and C to Hosts A, B, and C. Switch B does not advertise any routes learned from switch A (and the reverse).

Figure 26-3 EIGRP Stub Router Configuration



For more information about EIGRP stub routing, see “Configuring EIGRP Stub Routing” part of the *Cisco IOS IP Configuration Guide, Volume 2 of 3: Routing Protocols*, Release 12.2.

How to Configure EIGRP Stub Routing

The EIGRP Stub Routing feature improves network stability, reduces resource utilization, and simplifies stub router configuration.

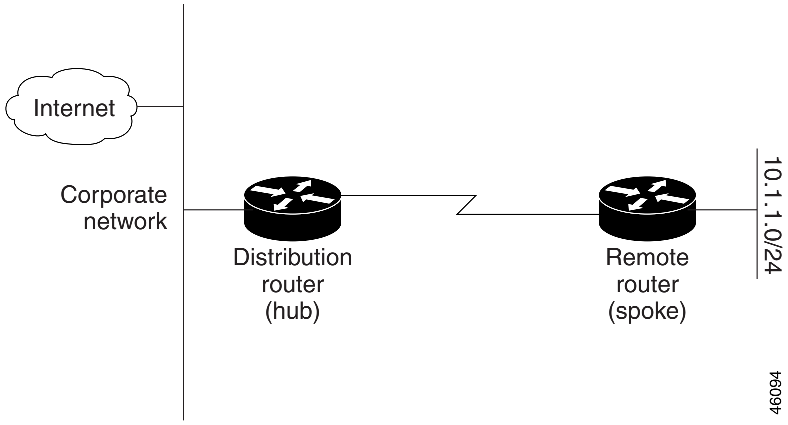
Stub routing is commonly used in a hub-and-spoke network topology. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers. The only route for IP traffic to follow into the remote router is through a distribution router. This type of configuration is commonly used in WAN topologies where the distribution router is directly connected to a WAN. The distribution router can be connected to many more remote routers. Often, the distribution router will be connected to 100 or more remote routers. In a hub-and-spoke topology, the remote router must forward all nonlocal traffic to a distribution router, so it becomes unnecessary for the remote router to hold a complete routing table. Generally, the distribution router need not send anything more than a default route to the remote router.

When using the EIGRP Stub Routing feature, you need to configure the distribution and remote routers to use EIGRP, and to configure only the remote router as a stub. Only specified routes are propagated from the remote (stub) router. The stub router responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” A router that is configured as a stub will send a special peer information packet to all neighboring routers to report its status as a stub router.

Any neighbor that receives a packet informing it of the stub status will not query the stub router for any routes, and a router that has a stub peer will not query that peer. The stub router will depend on the distribution router to send the proper updates to all peers.

Figure 26-4 shows a simple hub-and-spoke configuration.

Figure 26-4 Simple Hub-and-Spoke Network



The stub routing feature by itself does not prevent routes from being advertised to the remote router. In the example in Figure 26-4, the remote router can access the corporate network and the Internet through the distribution router only. Having a full route table on the remote router, in this example, would serve no functional purpose because the path to the corporate network and the Internet would always be through the distribution router. The larger route table would only reduce the amount of memory required by the remote router. Bandwidth and memory can be conserved by summarizing and filtering routes in the distribution router. The remote router need not receive routes that have been learned from other networks because the remote router must send all nonlocal traffic, regardless of destination, to the distribution router. If a true stub network is desired, the distribution router should be configured to send only a default route to the remote router. The EIGRP Stub Routing feature does not automatically enable summarization on the distribution router. In most cases, the network administrator will need to configure summarization on the distribution routers.

Note When configuring the distribution router to send only a default route to the remote router, you must use the **ip classless** command on the remote router. By default, the **ip classless** command is enabled in all Cisco IOS images that support the EIGRP Stub Routing feature.

Without the stub feature, even after the routes that are sent from the distribution router to the remote router have been filtered or summarized, a problem might occur. If a route is lost somewhere in the corporate network, EIGRP could send a query to the distribution router, which in turn will send a query to the remote router even if routes are being summarized. If there is a problem communicating over the WAN link between the distribution router and the remote router, an EIGRP stuck in active (SIA) condition could occur and cause instability elsewhere in the network. The EIGRP Stub Routing feature allows a network administrator to prevent queries from being sent to the remote router.

Dual-Homed Remote Topology

In addition to a simple hub-and-spoke network where a remote router is connected to a single distribution router, the remote router can be dual-homed to two or more distribution routers. This configuration adds redundancy and introduces unique issues, and the stub feature helps to address some of these issues.

A dual-homed remote router will have two or more distribution (hub) routers. However, the principles of stub routing are the same as they are with a hub-and-spoke topology. Figure 26-5 shows a common dual-homed remote topology with one remote router, but 100 or more routers could be connected on the same interfaces on distribution router 1 and distribution router 2. The remote router will use the best route to reach its destination. If distribution router 1 experiences a failure, the remote router can still use distribution router 2 to reach the corporate network.

Figure 26-5 Simple Dual-Homed Remote Topology

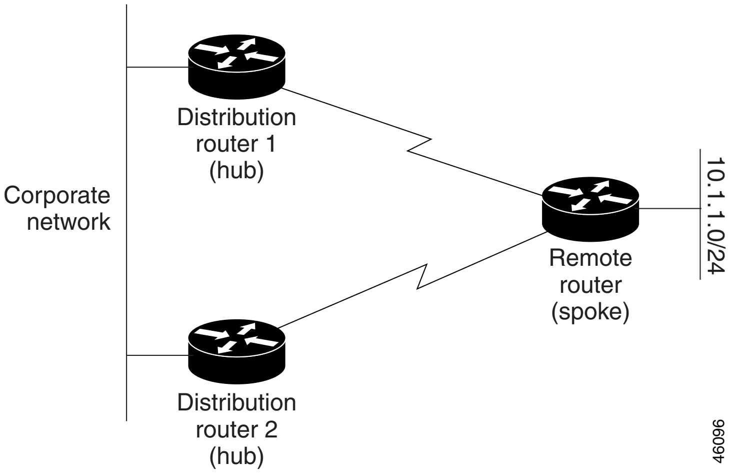


Figure 26-5 shows a simple dual-homed remote with one remote router and two distribution routers. Both distribution routers maintain routes to the corporate network and stub network 10.1.1.0/24.

Dual-homed routing can introduce instability into an EIGRP network. In Figure 26-6, distribution router 1 is directly connected to network 10.3.1.0/24. If summarization or filtering is applied on distribution router 1, the router will advertise network 10.3.1.0/24 to all of its directly connected EIGRP neighbors (distribution router 2 and the remote router).

Figure 26-6 Dual-Homed Remote Topology With Distribution Router 1 Connected to Two Networks

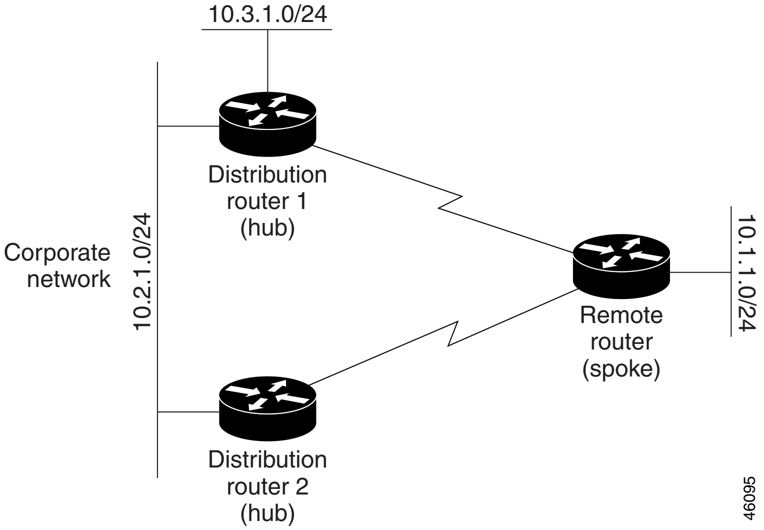
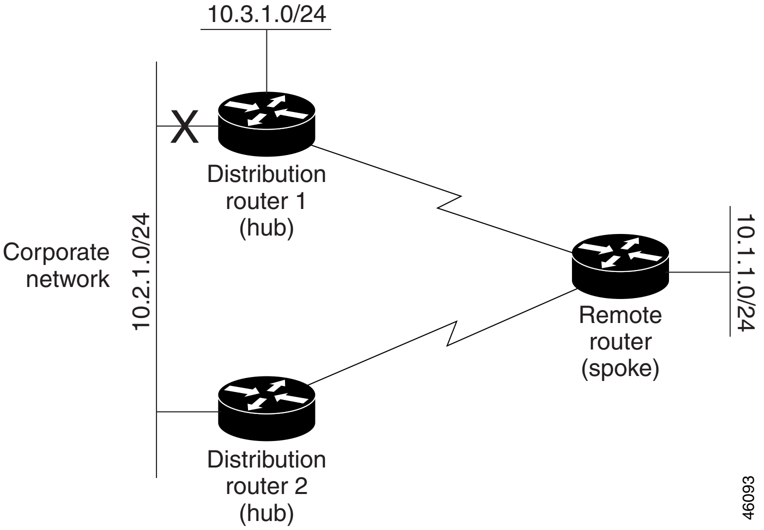


Figure 26-6 shows a simple dual-homed remote router where distribution router 1 is connected to both network 10.3.1.0/24 and network 10.2.1.0/24.

If the 10.2.1.0/24 link between distribution router 1 and distribution router 2 has failed, the lowest cost path to network 10.3.1.0/24 from distribution router 2 is through the remote router (see Figure 26-7). This route is not desirable because the traffic that was previously traveling across the corporate network 10.2.1.0/24 would now be sent across a much lower bandwidth connection. The over utilization of the lower bandwidth WAN connection can cause a number of problems that might affect the entire corporate network. The use of the lower bandwidth route that passes through the remote router might cause WAN EIGRP distribution routers to be dropped. Serial lines on distribution and remote routers could also be dropped, and EIGRP SIA errors on the distribution and core routers could occur.

Figure 26-7 Dual-Homed Remote Topology with a Failed Route to a Distribution Router



It is not desirable for traffic from distribution router 2 to travel through any remote router in order to reach network 10.3.1.0/24. If the links are sized to handle the load, it would be acceptable to use one of the backup routes. However, most networks of this type have remote routers located at remote offices with relatively slow links. This problem can be prevented if proper summarization is configured on the distribution router and remote router.

It is typically undesirable for traffic from a distribution router to use a remote router as a transit path. A typical connection from a distribution router to a remote router would have much less bandwidth than a connection at the network core. Attempting to use a remote router with a limited bandwidth connection as a transit path would generally produce excessive congestion to the remote router. The EIGRP Stub Routing feature can prevent this problem by preventing the remote router from advertising core routes back to distribution routers. Routes learned by the remote router from distribution router 1 will not be advertised to distribution router 2. Since the remote router will not advertise core routes to distribution router 2, the distribution router will not use the remote router as a transit for traffic destined for the network core.

The EIGRP Stub Routing feature can help to provide greater network stability. In the event of network instability, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit routers. Instead, distribution routers to which the stub router is connected answer the query on behalf of the stub router. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links. The EIGRP Stub Routing feature also simplifies the configuration and maintenance of hub-and-spoke networks. When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote routers to prevent those remote routers from appearing as transit paths to the hub routers.

Caution EIGRP Stub Routing should only be used on stub routers. A stub router is defined as a router connected to the network core or distribution layer through which core transit traffic should not flow. A stub router should not have any EIGRP neighbors other than distribution routers. Ignoring this restriction will cause undesirable behavior.

Note Multi-access interfaces, such as ATM, Ethernet, Frame Relay, ISDN PRI, and X.25, are supported by the EIGRP Stub Routing feature only when all routers on that interface, except the hub, are configured as stub routers.

EIGRP Stub Routing Configuration Task List

To configure EIGRP Stub Routing, perform the tasks described in the following sections. The tasks in the first section are required; the task in the last section is optional.

- Configuring EIGRP Stub Routing (required)
- Verifying EIGRP Stub Routing (optional)

Configuring EIGRP Stub Routing

To configure a remote or spoke router for EIGRP stub routing, use the following commands beginning in router configuration mode:

	Command	Purpose
Step 1	`router(config)# router eigrp 1`	Configures a remote or distribution router to run an EIGRP process.
Step 2	`router(config-router)# network network-number`	Specifies the network address of the EIGRP distribution router.
Step 3	`router(config-router)# eigrp stub [receive-only	connected	static	summary]`	Configures a remote router as an EIGRP stub router.

Verifying EIGRP Stub Routing

To verify that a remote router has been configured as a stub router with EIGRP, use the **show ip eigrp neighbor detail** command from the distribution router in privileged EXEC mode. The last line of the output will show the stub status of the remote or spoke router. The following example shows output is from the **show ip eigrp neighbor detail** command:

```
router# show ip eigrp neighbor detail

IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
0   10.1.1.2                Se3/1         11 00:00:59    1  4500  0  7
   Version 12.1/1.2, Retrans: 2, Retries: 0
   Stub Peer Advertising ( CONNECTED SUMMARY ) Routes
```

Monitoring and Maintaining EIGRP

To delete neighbors from the neighbor table, use the following command in EXEC mode:

Command	Purpose
`Router# clear ip eigrp neighbors [ip-address	interface]`	Deletes neighbors from the neighbor table.

To display various routing statistics, use the following commands in EXEC mode:

Command	Purpose
`Router# show ip eigrp interfaces [interface] [as-number]`	Displays information about interfaces configured for EIGRP.
`Router# show ip eigrp neighbors [type|number|static]`	Displays the EIGRP discovered neighbors.
`Router# show ip eigrp topology [autonomous-system-number	[[ip-address] mask]]`	Displays the EIGRP topology table for a given process.
`Router# show ip eigrp traffic [autonomous-system-number]`	Displays the number of packets sent and received for all or a specified EIGRP process.

EIGRP Configuration Examples

This section contains the following examples:

- Route Summarization Example
- Route Authentication Example
- Stub Routing Example

Route Summarization Example

The following example configures route summarization on the interface and also configures the autosummary feature. This configuration causes EIGRP to summarize network 10.0.0.0 out Ethernet interface 0 only. In addition, this example disables autosummarization.

```
interface Ethernet 0
 ip summary-address eigrp 1 10.0.0.0 255.0.0.0
!
router eigrp 1
 network 172.16.0.0
 no auto-summary
```

Note You should not use the **ip summary-address eigrp** summarization command to generate the default route (0.0.0.0) from an interface. This causes the creation of an EIGRP summary default route to the null 0 interface with an administrative distance of 5. The low administrative distance of this default route can cause this route to displace default routes learned from other neighbors from the routing table. If the default route learned from the neighbors is displaced by the summary default route, or if the summary route is the only default route present, all traffic destined for the default route will not leave the router, instead, this traffic will be sent to the null 0 interface where it is dropped.

The recommended way to send only the default route out a given interface is to use a **distribute-list** command. You can configure this command to filter all outbound route advertisements sent out the interface with the exception of the default (0.0.0.0).

Route Authentication Example

The following example enables MD5 authentication on EIGRP packets in autonomous system 1.

Router A

```
interface ethernet 1
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 holly
key chain holly
 key 1
  key-string 0987654321
  accept-lifetime 04:00:00 Dec 4 1996 infinite
  send-lifetime 04:00:00 Dec 4 1996 04:48:00 Dec 4 1996
 exit
 key 2
  key-string 1234567890
  accept-lifetime 04:00:00 Dec 4 1996 infinite
  send-lifetime 04:45:00 Dec 4 1996 infinite
```

Router B

```
interface ethernet 1
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 mikel
key chain mikel
 key 1
  key-string 0987654321
  accept-lifetime 04:00:00 Dec 4 1996 infinite
  send-lifetime 04:00:00 Dec 4 1996 infinite
 exit
 key 2
  key-string 1234567890
  accept-lifetime 04:00:00 Dec 4 1996 infinite
  send-lifetime 04:45:00 Dec 4 1996 infinite
```

Router A will accept and attempt to verify the MD5 digest of any EIGRP packet with a key equal to 1. It will also accept a packet with a key equal to 2. All other MD5 packets will be dropped. Router A will send all EIGRP packets with key 2.

Router B will accept key 1 or key 2, and will send key 1. In this scenario, MD5 will authenticate.

Stub Routing Example

A router that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor routers by default. Four optional keywords can be used with the **eigrp stub** command to modify this behavior:

- **receive-only**
- **connected**
- **static**
- **summary**

This section provides configuration examples for all forms of the **eigrp stub** command. The **eigrp stub** command can be modified with several options, and these options can be used in any combination except for the **receive-only** keyword. The **receive-only** keyword will restrict the router from sharing any of its routes with any other router in that EIGRP autonomous system, and the **receive-only** keyword will not permit any other option to be specified because it prevents any type of route from being sent. The three other optional keywords (**connected**, **static**, and **summary**) can be used in any combination but cannot be used with the **receive-only** keyword. If any of these three keywords is used individually with the **eigrp stub** command, connected and summary routes will not be sent automatically.

The **connected** keyword will permit the EIGRP Stub Routing feature to send connected routes. If the connected routes are not covered by a network statement, it may be necessary to redistribute connected routes with the **redistribute connected** command under the EIGRP process. This option is enabled by default.

The **static** keyword will permit the EIGRP Stub Routing feature to send static routes. Without this option, EIGRP will not send any static routes, including internal static routes that normally would be automatically redistributed. It will still be necessary to redistribute static routes with the **redistribute static** command.

The **summary** keyword will permit the EIGRP Stub Routing feature to send summary routes. Summary routes can be created manually with the **summary address** command or automatically at a major network border router with the **auto-summary** command enabled. This option is enabled by default.

In the following example, the **eigrp stub** command is used to configure the router as a stub that advertises connected and summary routes:

```
router eigrp 1
network 10.0.0.0
eigrp stub
```

In the following example, the **eigrp stub connected static** command is used to configure the router as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
router eigrp 1
network 10.0.0.0
eigrp stub connected static
```

In the following example, the **eigrp stub receive-only** command is used to configure the router as a stub, and connected, summary, or static routes will not be sent:

```
router eigrp 1
network 10.0.0.0 eigrp
stub receive-only
```

CHAPTER 27

Configuring Cisco Express Forwarding

This chapter describes Cisco Express Forwarding (CEF) on the Catalyst 4000 family switch. It also provides guidelines, procedures, and examples to configure this feature.

This chapter includes the following major sections:

- Overview of CEF, page 27-1
- Catalyst 4500 Series Switch Implementation of CEF, page 27-3
- CEF Configuration Restrictions, page 27-6
- Configuring CEF, page 27-6
- Monitoring and Maintaining CEF, page 27-8

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of CEF

This section contains information on the two primary components that comprise the CEF operation:

- Benefits of CEF, page 27-1
- Forwarding Information Base, page 27-2
- Adjacency Tables, page 27-2

Benefits of CEF

CEF is advanced Layer 3 IP switching technology that optimizes performance and scalability for large networks with dynamic traffic patterns or networks with intensive web-based applications and interactive sessions.

CEF provides the following benefits:

- Improves performance over the caching schemes of multilayer switches, which often flush the entire cache when information changes in the routing tables.
- Provides load balancing that distributes packets across multiple links based on Layer 3 routing information. If a network device discovers multiple paths to a destination, the routing table is updated with multiple entries for that destination. Traffic to that destination is then distributed among the various paths.

CEF stores information in several data structures rather than the route cache of multilayer switches. The data structures optimize lookup for efficient packet forwarding.

Forwarding Information Base

The Forwarding Information Base (FIB) is a table that contains a copy of the forwarding information in the IP routing table. When routing or topology changes occur in the network, the route processor updates the IP routing table and CEF updates the FIB. Because there is a one-to-one correlation between FIB entries and routing table entries, the FIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths, such as fast switching and optimum switching. CEF uses the FIB to make IP destination-based switching decisions and maintain next-hop address information based on the information in the IP routing table.

On the Catalyst 4500 series switches, CEF loads the FIB in to the Integrated Switching Engine hardware to increase the performance of forwarding. The Integrated Switching Engine has a finite number of forwarding slots for storing routing information. If this limit is exceeded, CEF is automatically disabled and all packets are forwarded in software. In this situation, you should reduce the number of routes on the switch and then reenable hardware switching with the **ip cef** command.

Adjacency Tables

In addition to the FIB, CEF uses adjacency tables to prepend Layer 2 addressing information. Nodes in the network are said to be *adjacent* if they are within a single hop from each other. The adjacency table maintains Layer 2 next-hop addresses for all FIB entries.

Adjacency Discovery

The adjacency table is populated as new adjacent nodes are discovered. Each time an adjacency entry is created (such as through the Address Resolution Protocol (ARP), a link-layer header for that adjacent node is stored in the adjacency table. Once a route is determined, the link-layer header points to a next hop and corresponding adjacency entry. The link-layer header is subsequently used for encapsulation during CEF switching of packets.

Adjacency Resolution

A route might have several paths to a destination prefix, such as when a router is configured for simultaneous load balancing and redundancy. For each resolved path, a pointer is added for the adjacency corresponding to the next-hop interface for that path. This mechanism is used for load balancing across several paths.

Adjacency Types That Require Special Handling

In addition to adjacencies for next-hop interfaces (host-route adjacencies), other types of adjacencies are used to expedite switching when certain exception conditions exist. When the prefix is defined, prefixes requiring exception processing are cached with one of the special adjacencies listed in Table 27-1.

Table 27-1 Adjacency Types for Exception Processing

This adjacency type...	Receives this processing...
Null adjacency	Packets destined for a Null0 interface are dropped. A Null0 interface can be used as an effective form of access filtering.
Glean adjacency	When a router is connected directly to several hosts, the FIB table on the router maintains a prefix for the subnet rather than for each individual host. The subnet prefix points to a glean adjacency. When packets need to be forwarded to a specific host, the adjacency database is gleaned for the specific prefix.
Punt adjacency	Features that require special handling or features that are not yet supported by CEF switching are sent (punted) to the next higher switching level.
Discard adjacency	Packets are discarded.
Drop adjacency	Packets are dropped.

Unresolved Adjacency

When a link-layer header is prepended to packets, FIB requires the prepend to point to an adjacency corresponding to the next hop. If an adjacency was created by FIB and was not discovered through a mechanism such as ARP, the Layer 2 addressing information is not known and the adjacency is considered incomplete. When the Layer 2 information is known, the packet is forwarded to the route processor, and the adjacency is determined through ARP.

Catalyst 4500 Series Switch Implementation of CEF

This section contains the following subsections:

- Hardware and Software Switching, page 27-4
- Load Balancing, page 27-6
- Software Interfaces, page 27-6

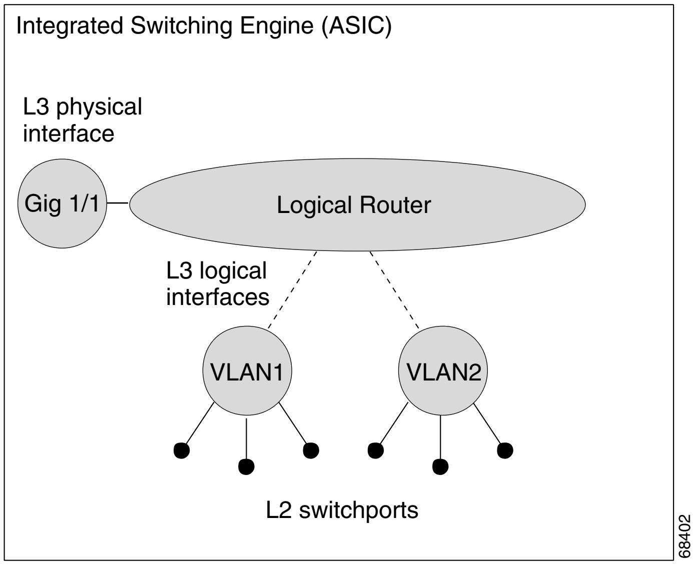
Catalyst 4000 Family switches support an ASIC-based Integrated Switching Engine that provides these features:

- Ethernet bridging at Layer 2
- IP routing at Layer 3

Because the ASIC is specifically designed to forward packets, the Integrated Switching Engine hardware can run this process much faster than CPU subsystem software.

Figure 27-1 shows a high-level view of the ASIC-based Layer 2 and Layer 3 switching process on the Integrated Switching Engine.

Figure 27-1 Logical L2/L3 Switch Components



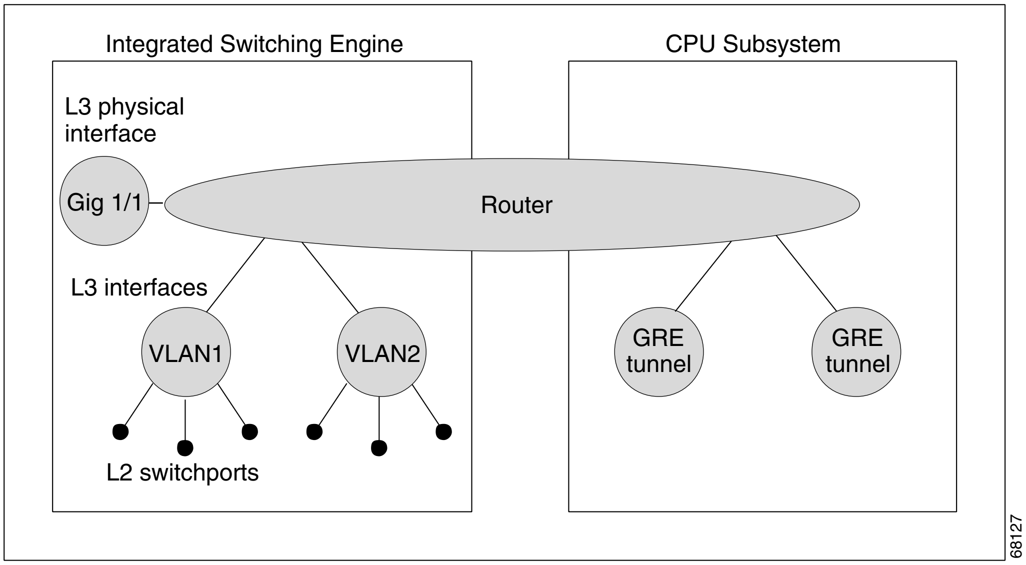
The Integrated Switching Engine performs inter-VLAN routing on logical Layer 3 interfaces with the ASIC hardware. The ASIC hardware also supports a physical Layer 3 interface that can be configured to connect with a host, a switch, or a router.

Hardware and Software Switching

For the majority of packets, the Integrated Switching Engine performs the packet forwarding function in hardware. These packets are hardware-switched at very high rates. Exception packets are forwarded by the CPU subsystem software. Statistic reports should show that the Integrated Switching Engine is forwarding the vast majority of packets in hardware. Software forwarding is significantly slower than hardware forwarding, but packets forwarded by the CPU subsystem do not reduce hardware forwarding speed.

Figure 27-2 shows a logical view of the Integrated Switching Engine and the CPU subsystem switching components.

Figure 27-2 Hardware and Software Switching Components



The Integrated Switching Engine performs inter-VLAN routing in hardware. The CPU subsystem software supports Layer 3 interfaces to VLANs that use Subnetwork Access Protocol (SNAP) encapsulation. The CPU subsystem software also supports generic routing encapsulation (GRE) tunnel.

Hardware Switching

Hardware switching is the normal operation for the Supervisor Engine III and Supervisor Engine IV.

Software Switching

Software switching occurs when traffic cannot be processed in hardware. The following types of exception packets are processed in software at a much slower rate:

- Packets that use IP header options

Note Packets that use TCP header options are switched in hardware because they do not affect the forwarding decision.

- Packets that have an expiring IP time-to-live (TTL) counter
- Packets that are forwarded to a tunnel interface
- Packets that arrive with non-supported encapsulation types
- Packets that are routed to an interface with non-supported encapsulation types
- Packets that exceed the MTU of an output interface and must be fragmented
- Packets that require an IGMP redirect to be routed
- 802.3 Ethernet packets

Load Balancing

The Catalyst 4000 family switch supports load balancing for routing packets in the Integrated Switching Engine hardware. Load balancing is always enabled. It works when multiple routes for the same network with different next-hop addresses are configured. These routes can be configured either statically or through a routing protocol such as OSPF or EIGRP.

The hardware makes a forwarding decision by using a hardware load sharing hash function to compute a value, based on the source and destination IP addresses and the source and destination TCP port numbers (if available). This load sharing hash value is then used to select which route to use to forward the packet. All hardware switching within a particular flow (such as a TCP connection) will be routed to the same next hop, thereby reducing the chance that packet reordering will occur. Up to eight different routes for a particular network are supported.

Software Interfaces

Cisco IOS for the Catalyst 4000 family switch supports GRE and IP tunnel interfaces that are not part of the hardware forwarding engine. All packets that flow to or from these interfaces must be processed in software and will have a significantly lower forwarding rate than that of hardware-switched interfaces. Also, Layer 2 features are not supported on these interfaces.

CEF Configuration Restrictions

The Integrated Switching Engine supports only ARPA and ISL/802.1q encapsulation types for Layer 3 switching in hardware. The CPU subsystem supports a number of encapsulations such as SNAP for Layer 2 switching that you can use for Layer 3 switching in software.

Configuring CEF

These sections describe how to configure CEF:

- Enabling CEF, page 27-6
- Configuring Load Balancing for CEF, page 27-7

Enabling CEF

By default, CEF is enabled globally on the Catalyst 4000 family switch. No configuration is required.

To reenable CEF, perform this task:

Command	Purpose
`Switch(config)# ip cef distributed`	Enables standard CEF operation.

Configuring Load Balancing for CEF

CEF load balancing is based on a combination of source and destination packet information; it allows you to optimize resources by distributing traffic over multiple paths for transferring data to a destination. You can configure load balancing on a per-destination basis. Load-balancing decisions are made on the outbound interface. You can configure per-destination load balancing for CEF on outbound interfaces.

The following topics are discussed:

- Configuring Per-Destination Load Balancing, page 27-7
- Configuring Load Sharing Hash Function, page 27-7
- Viewing CEF Information, page 27-8

Configuring Per-Destination Load Balancing

Per-destination load balancing is enabled by default when you enable CEF. To use per-destination load balancing, you do not perform any additional tasks once you enable CEF.

Per-destination load balancing allows the router to use multiple paths to achieve load sharing. Packets for a given source-destination host pair are guaranteed to take the same path, even if multiple paths are available. Traffic destined for different pairs tend to take different paths. Per-destination load balancing is enabled by default when you enable CEF; it is the load balancing method of choice in most situations.

Because per-destination load balancing depends on the statistical distribution of traffic, load sharing becomes more effective as the number of source-destination pairs increases.

You can use per-destination load balancing to ensure that packets for a given host pair arrive in order. All packets for a certain host pair are routed over the same link or links.

Configuring Load Sharing Hash Function

When multiple unicast routes exist to a particular destination IP prefix, the hardware will send packets matching that prefix across all possible routes, thereby sharing the load across all next hop routers. By default, the route used is chosen by computing a hash of the source and destination IP addresses and using the resulting value to select the route. This preserves packet ordering for packets within a flow by ensuring that all packets within a single IP source/destination flow are sent on the same route, but it provides a near-random distribution of flows to routes.

The load-sharing hash function can be changed, so that in addition to the source and destination IP addresses, the source TCP/UDP port, the destination TCP/UDP port, or both can also be included in the hash.

To the configure load sharing hash function to use the source and/or destination ports, perform this task:

Command	Purpose
`Switch (config)# [no] ip cef load-sharing algorithm include-ports source destination]`	Enables load sharing hash function to use source and destination ports. Use the **no** keyword to set the switch to use the default Cisco IOS load-sharing algorithm.

For more information on load sharing, refer to the *Configuring Cisco Express Forwarding* module of the Cisco IOS documentation at this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fwitch_c/swprt1/xcfcefc.htm

Note The **include-ports** option does not apply to software-switched traffic on the Catalyst 4500 series switches.

Viewing CEF Information

You can view the collected CEF information. To view CEF information, perform this task:

Command	Purpose
`Switch# show ip cef`	Displays the collected CEF information.

Monitoring and Maintaining CEF

To display information about IP traffic, perform this task:

Command	Purpose
`Switch# show interface type slot/interface | begin L3`	Displays a summary of IP unicast traffic.

This example shows how to display information about IP unicast traffic on interface Fast Ethernet 3/3:

```
Switch# 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...>
Switch#
```

Note The IP unicast packet count is updated approximately every five seconds.

Displaying IP Statistics

IP unicast statistics are gathered on a per-interface basis. To display IP statistics, perform this task:

Command	Purpose
`Switch# show interface type number counters detail`	Displays IP statistics.

This example shows how to display IP unicast statistics for Part 3/1:

```
Switch# show interface fastethernet 3/1 counters detail

Port                InBytes       InUcastPkts      InMcastPkts       InBcastPkts
Fa3/1            7263539133           5998222          6412307               156

Port               OutBytes      OutUcastPkts     OutMcastPkts      OutBcastPkts
Fa3/1            7560137031           5079852         12140475                38

Port              InPkts 64        OutPkts 64    InPkts 65-127    OutPkts 65-127
Fa3/1                 11274            168536          7650482          12395769

Port         InPkts 128-255   OutPkts 128-255   InPkts 256-511   OutPkts 256-511
Fa3/1                 31191             55269            26923             65017

Port        InPkts 512-1023  OutPkts 512-1023
Fa3/1                133807            151582

Port       InPkts 1024-1518 OutPkts 1024-1518 InPkts 1519-1548 OutPkts 1519-1548
Fa3/1                   N/A               N/A              N/A               N/A

Port       InPkts 1024-1522 OutPkts 1024-1522 InPkts 1523-1548 OutPkts 1523-1548
Fa3/1               4557008           4384192                0                 0

Port       Tx-Bytes-Queue-1  Tx-Bytes-Queue-2 Tx-Bytes-Queue-3  Tx-Bytes-Queue-4
Fa3/1                    64                 0            91007        7666686162

Port       Tx-Drops-Queue-1  Tx-Drops-Queue-2 Tx-Drops-Queue-3  Tx-Drops-Queue-4
Fa3/1                     0                 0                0                 0

Port         Rx-No-Pkt-Buff     RxPauseFrames    TxPauseFrames   PauseFramesDrop
Fa3/1                     0                 0                0               N/A

Port       UnsupOpcodePause
Fa3/1                     0
Switch#
```

To display CEF (software switched) and hardware IP unicast adjacency table information, perform this task:

Command	Purpose
Switch# **show adjacency** [*interface*] [**detail** | **internal** | **summary**]	Displays detailed adjacency information, including Layer 2 information, when the optional **detail** keyword is used.

This example shows how to display adjacency statistics:

```
Switch# show adjacency gigabitethernet 3/5 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 10 seconds.

CHAPTER 28

Configuring Unicast Reverse Path Forwarding

This chapter describes the Unicast Reverse Path Forwarding (Unicast RPF) feature. The Unicast RPF feature helps to mitigate problems that are caused by malformed or forged IP source addresses that are passing through a router.

For a complete description of the Unicast RPF commands in this chapter, refer to the chapter "Unicast Reverse Path Forwarding Commands" of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

- About Unicast Reverse Path Forwarding
- Unicast RPF Configuration Task List
- Monitoring and Maintaining Unicast RPF
- Monitoring and Maintaining Unicast RPF
- Unicast RPF Configuration Examples

About Unicast Reverse Path Forwarding

The Unicast RPF feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

This section covers the following information:

- How Unicast RPF Works
- Implementing Unicast RPF
- Restrictions
- Related Features and Technologies
- Prerequisites to Configuring Unicast RPF

How Unicast RPF Works

When Unicast RPF is enabled on an interface, the router examines all packets received as input on that interface to make sure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This "look backwards" ability is available only when Cisco express forwarding (CEF) is enabled on the router, because the lookup relies on the presence of the Forwarding Information Base (FIB). CEF generates the FIB as part of its operation.

Note Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

Unicast RPF checks to see if any packet received at a router interface arrives on the best return path (return route) to the source of the packet. Unicast RPF does this by doing a reverse lookup in the CEF table. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, it might mean that the source address was modified. If Unicast RPF does not find a reverse path for the packet, the packet is dropped.

Note With Unicast RPF, all equal-cost "best" return paths are considered valid. This means that Unicast RPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where EIGRP variants are being used and unequal candidate paths back to the source IP address exist.

When a packet is received at the interface where Unicast RPF and ACLs have been configured, the following actions occur:

Step 1 Input ACLs configured on the inbound interface are checked.

Step 2 Unicast RPF checks to see if the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.

Step 3 CEF table (FIB) lookup is carried out for packet forwarding.

Step 4 Output ACLs are checked on the outbound interface.

Step 5 The packet is forwarded.

This section provides information about Unicast RPF enhancements:

- Access Control Lists and Logging
- Per-Interface Statistics

Figure 28-1 illustrates how Unicast RPF and CEF work together to validate IP source addresses by verifying packet return paths. In this example, a customer has sent a packet having a source address of 192.168.1.1 from interface Gigabit Ethernet 1/1. Unicast RPF checks the FIB to see if 192.168.1.1 has a path to Gigabit Ethernet 1/1. If there is a matching path, the packet is forwarded. If there is no matching path, the packet is dropped.

Figure 28-1 Unicast RPF Validating IP Source Addresses

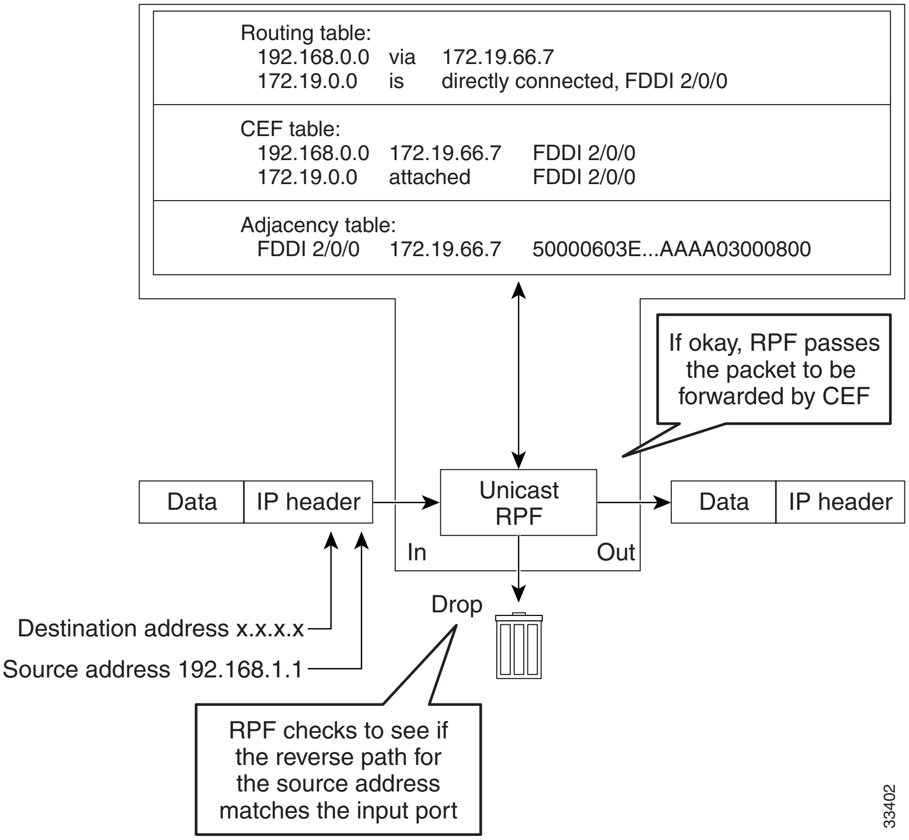
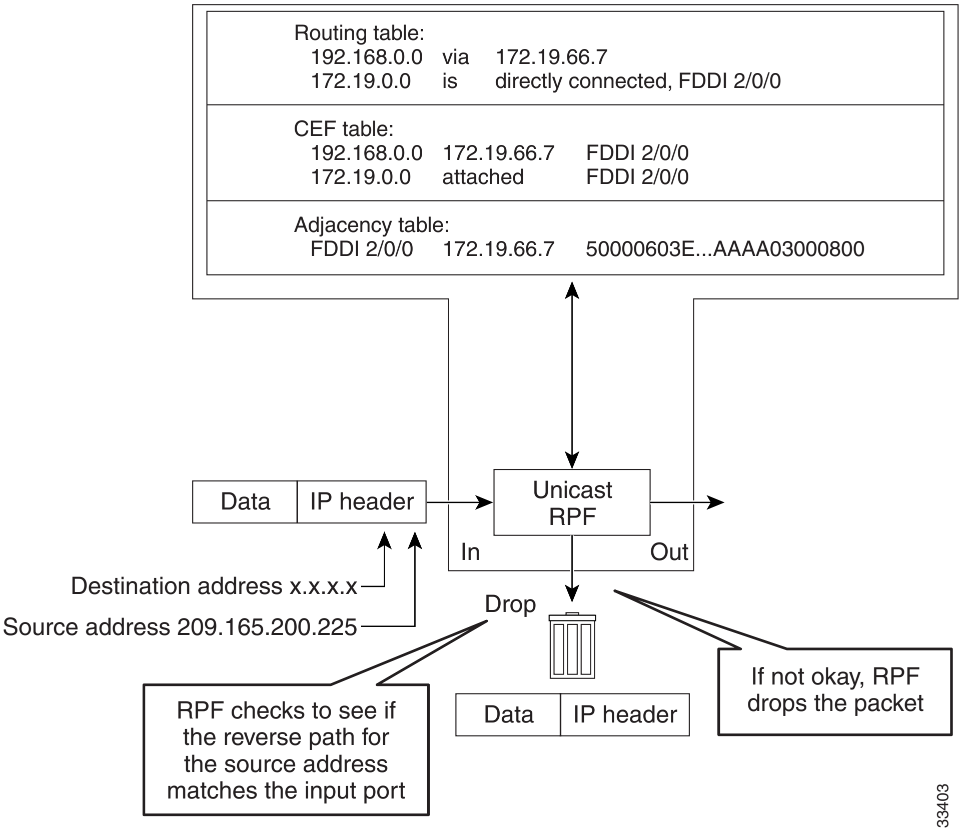

Routing table:
192.168.0.0 via 172.19.66.7
172.19.0.0 is directly connected, FDDI 2/0/0
CEF table:
192.168.0.0 172.19.66.7 FDDI 2/0/0
172.19.0.0 attached FDDI 2/0/0
Adjacency table:
FDDI 2/0/0 172.19.66.7 50000603E...AAAA03000800
If okay, RPF passes the packet to be forwarded by CEF
Data
IP header
Unicast RPF
In
Out
Data
IP header
Drop
Destination address x.x.x.x
Source address 192.168.1.1
RPF checks to see if the reverse path for the source address matches the input port
33402

Figure 28-2 illustrates how Unicast RPF drops packets that fail validation. In this example, a customer has sent a packet having a source address of 209.165.200.225, which is received at interface Gigabit Ethernet 1/1. Unicast RPF checks the FIB to see if 209.165.200.225 has a return path to Gigabit Ethernet 1/1. If there is a matching path, the packet is forwarded. In this case, there is no reverse entry in the routing table that routes the customer packet back to source address 209.165.200.225 on interface Gigabit Ethernet 1/1, and so the packet is dropped.

Figure 28-2 Unicast RPF Dropping Packets That Fail Verification


Routing table:
192.168.0.0 via 172.19.66.7
172.19.0.0 is directly connected, FDDI 2/0/0
CEF table:
192.168.0.0 172.19.66.7 FDDI 2/0/0
172.19.0.0 attached FDDI 2/0/0
Adjacency table:
FDDI 2/0/0 172.19.66.7 50000603E...AAAA03000800
Data
IP header
Unicast RPF
In
Out
Destination address x.x.x.x
Source address 209.165.200.225
Drop
RPF checks to see if the reverse path for the source address matches the input port
Data
IP header
If not okay, RPF drops the packet
33403

Implementing Unicast RPF

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB matching the route to the receiving interface. Adding a route in the FIB can be done via static route, network statement, or dynamic routing. (ACLs permit Unicast RPF to be used when packets are known to be arriving by specific, less optimal asymmetric input paths.)
- IP source addresses at the receiving interface must match the routing entry for the interface.
- Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

Given these implementation principles, Unicast RPF becomes a tool that network administrators can use not only for their customers but also for their downstream network or ISP, even if the downstream network or ISP has other connections to the Internet.

Caution Using optional BGP attributes such as weight and local preference, the best path back to the source address can be modified. Modification would affect the operation of Unicast RPF.

This section provides information about the implementation of Unicast RPF:

- Security Policy and Unicast RPF
- Where to Use Unicast RPF
- Routing Table Requirements
- Where Not to Use Unicast RPF
- Unicast RPF with BOOTP and DHCP

Security Policy and Unicast RPF

Consider the following points in determining your policy for deploying Unicast RPF:

- Unicast RPF must be applied at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF on an aggregation router helps mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying Unicast RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources, the better the chances of mitigating large-scale network disruptions throughout the Internet community, and the better the chances of tracing the source of an attack.
- Unicast RPF will not inspect IP packets encapsulated in tunnels, such as GRE, LT2P, or PPTP. Unicast RPF must be configured at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.

Where to Use Unicast RPF

Unicast RPF can be used in any “single-homed” environment where there is essentially only one access point out of the network; that is, one upstream connection. Networks having one access point offer the best example of symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet. Unicast RPF is best used at the network perimeter for Internet, intranet, or extranet environments, or in ISP environments for customer network terminations.

The following sections provide a look at implementing Unicast RPF in two network environments:

- Enterprise Networks with a Single Connection to an ISP
- Network Access Server Application (Applying Unicast RPF in PSTN/ISDN PoP Aggregation Routers)

Enterprise Networks with a Single Connection to an ISP

In enterprise networks, one objective of using Unicast RPF for filtering traffic at the input interface (a process called *ingress filtering*) is for protection from malformed packets arriving from the Internet. Traditionally, local networks that have one connection to the Internet would use ACLs at the receiving interface to prevent spoofed packets from the Internet from entering their local network.

ACLs work well for many single-homed customers; however, there are trade-offs when ACLs are used as ingress filters, including two commonly referenced limitations:

- Packet per second (PPS) performance at very high packet rates

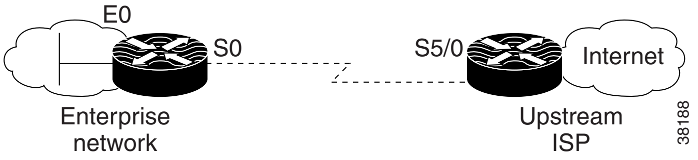
Note This restriction applies only to software packet forwarding. Hardware packet forwarding is the same on both ACL and uRPF.

- Maintenance of the ACL (whenever there are new addresses added to the network)

Unicast RPF is one tool that addresses both of these limitations. With Unicast RPF, ingress filtering is done at CEF PPS rates. This processing speed makes a difference when the link is more than 1 Mbps. Additionally, since Unicast RPF uses the FIB, no ACL maintenance is necessary, and thus the administration overhead of traditional ACLs is reduced. The following figure and example demonstrate how Unicast RPF is configured for ingress filtering.

Figure 28-3 illustrates an enterprise network that has a single link to an upstream ISP. In this example, Unicast RPF is applied at interface Gigabit Ethernet 1/1 on the Enterprise router for protection from malformed packets arriving from the Internet. Unicast RPF is also applied at interface Gigabit Ethernet 2/1 on the ISP router for protection from malformed packets arriving from the enterprise network.

Figure 28-3 Enterprise Network Using Unicast RPF for Ingress Filtering



Using the topography in Figure 28-3, a typical configuration (assuming that CEF is turned on) on the ISP router would be as follows:

```
interface Gigabit Ethernet 2/1
  description Link to Enterprise Network
  ip address 192.168.3.1 255.255.255.255
  no switchport
  ip address 10.1.1.2 255.255.255.0
  ip route 192.168.10.0 255.255.255.0 10.1.1.1
  ip verify unicast source reachable-via rx allow-default
```

The gateway router configuration of the enterprise network (assuming that CEF is turned on) would look similar to the following:

```
interface Gigabit Ethernet 1/2
 description ExampleCorp LAN
 ip address 192.168.10.1 255.255.252.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp

interface Gigabit Ethernet 1/1
 description Link to Internet
 no switchport
 ip address 10.1.1.1 255.255.255.0
 ip route 0.0.0.0 0.0.0.0 10.1.1.2
 ip verify unicast source reachable-via allow-default
 no ip proxy-arp
 no ip redirects
 no ip directed-broadcast
```

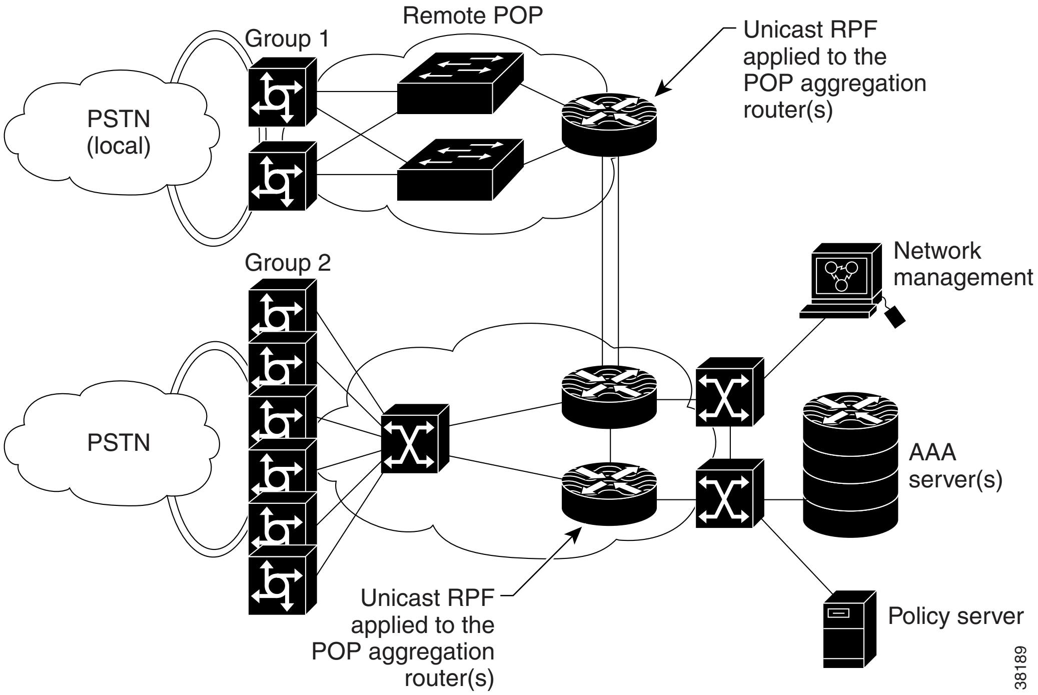
Notice that Unicast RPF works with a single default route. There are no additional routes or routing protocols. Network 192.168.10.0/22 is a connected network. Hence, packets coming from the Internet with a source address in the range 192.168.10.0/22 will be dropped by Unicast RPF.

Network Access Server Application (Applying Unicast RPF in PSTN/ISDN PoP Aggregation Routers)

Aggregation routers are ideal places to use Unicast RPF with single-homed clients. Unicast RPF works equally well on leased-line or PSTN/ISDN/xDSL customer connections into the Internet. In fact, dialup connections are reputed to be the greatest source of DoS attacks using forged IP addresses. As long as the network access server supports CEF, Unicast RPF will work. In this topology, the customer aggregation routers need not have the full Internet routing table. Aggregation routers need the routing prefixes information (IP address block); hence, information configured or redistributed in the Interior Gateway Protocol (IGP) or Internal Border Gateway Protocol (IBGP) (depending on the way that you add customer routes into your network) would be enough for Unicast RPF to do its job.

Figure 28-4 illustrates the application of Unicast RPF to the aggregation and access routers for an Internet service provider (ISP) point of presence (POP), with the ISP routers providing dialup customer connections. In this example, Unicast RPF is applied upstream from the customer dialup connection router on the receiving (input) interfaces of the ISP aggregation routers.

Figure 28-4 Unicast RPF Applied to PSTN/ISDN Customer Connections


Remote POP
Group 1
Unicast RPF applied to the POP aggregation router(s)
PSTN (local)
Group 2
Network management
PSTN
AAA server(s)
Unicast RPF applied to the POP aggregation router(s)
Policy server
38189

Routing Table Requirements

To work correctly, Unicast RPF needs proper information in the CEF tables. This requirement does not mean that the router must have the entire Internet routing table. The amount of routing information needed in the CEF tables depends on where Unicast RPF is configured and what functions the router performs in the network. For example, in an ISP environment, a router that is a leased-line aggregation router for customers needs only the information based on the static routes redistributed into the IGP or IBGP (depending on which technique is used in the network). Unicast RPF would be configured on the customer interfaces—hence the requirement for minimal routing information. In another scenario, a single-homed ISP could place Unicast RPF on the gateway link to the Internet. The full Internet routing table would be required. Requiring the full routing table would help protect the ISP from external DoS attacks that use addresses that are not in the Internet routing table.

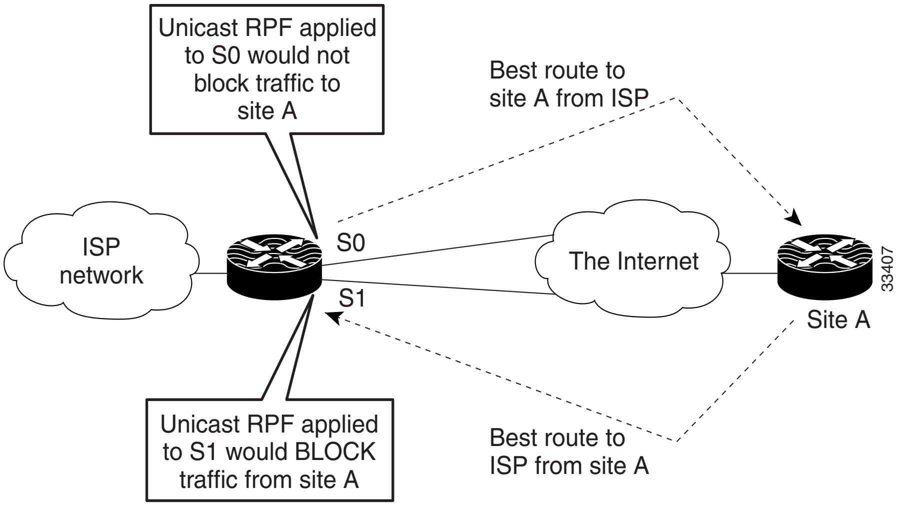
Where Not to Use Unicast RPF

Unicast RPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry (see Figure 28-5), meaning multiple routes to the source of a packet. Unicast RPF should be applied only where there is natural or configured symmetry. As long as administrators carefully plan which interfaces they activate Unicast RPF on, routing asymmetry is not a serious problem.

For example, routers at the edge of the network of an ISP are more likely to have symmetrical reverse paths than routers that are in the core of the ISP network. Routers that are in the core of the ISP network have no guarantee that the best forwarding path out of the router will be the path selected for packets returning to the router. Hence, it is not recommended that you apply Unicast RPF where there is a chance of asymmetric routing, unless you use ACLs to allow the router to accept incoming packets. ACLs permit Unicast RPF to be used when packets are known to be arriving by specific, less optimal asymmetric input paths. However, it is simplest to place Unicast RPF only at the edge of a network or, for an ISP, at the customer edge of the network.

Figure 28-5 illustrates how Unicast RPF can block legitimate traffic in an asymmetrical routing environment.

Figure 28-5 Unicast RPF Blocking Traffic in an Asymmetrical Routing Environment


Unicast RPF applied to S0 would not block traffic to site A
Best route to site A from ISP
ISP network
S0
The Internet
S1
Site A
33407
Unicast RPF applied to S1 would BLOCK traffic from site A
Best route to ISP from site A

Unicast RPF with BOOTP and DHCP

Unicast RPF will allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) functions work properly.

Restrictions

There are some basic restrictions to applying Unicast RPF to multihomed clients:

- Clients should not be multihomed to the same router because multihoming defeats the purpose of building a redundant service for the client.
- Customers must ensure that the packets flowing up the link (out to the Internet) match the route advertised out the link. Otherwise, Unicast RPF filters those packets as malformed packets.

Related Features and Technologies

For more information about Unicast RPF-related features and technologies, review the following:

- Unicast RPF requires Cisco express forwarding (CEF) to function properly on the router. For more information about CEF, refer to the *Cisco IOS Switching Services Configuration Guide*.
- Unicast RPF can be more effective at mitigating spoofing attacks when combined with a policy of *ingress* and *egress* filtering using Cisco IOS access control lists (ACLs).
  - Ingress filtering applies filters to traffic received at a network interface from either internal or external networks. With ingress filtering, packets that arrive from other networks or the Internet and that have a source address that matches a local network, private, or broadcast address are dropped. In ISP environments, for example, ingress filtering can apply to traffic received at the router from either the client (customer) or the Internet.
  - Egress filtering applies filters to traffic exiting a network interface (the sending interface). By filtering packets on routers that connect your network to the Internet or to other networks, you can permit only packets with valid source IP addresses to leave your network.

  For more information on network filtering, refer to RFC 2267 and to the *Cisco IOS IP Configuration Guide*.
- Cisco IOS software provides additional features that can help mitigate DoS attacks:
  - Committed Access Rate (CAR). CAR allows you to enforce a bandwidth policy against network traffic that matches an access list. For example, CAR allows you to rate-limit what should be low-volume traffic, such as ICMP traffic. To find out more about CAR, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide*.
  - Context-based Access Control (CBAC). CBAC selectively blocks any network traffic not originated by a protected network. CBAC uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. Setting timeout values for network sessions helps mitigate DoS attacks by freeing up system resources, dropping sessions after a specified amount of time. For more information on CBAC, refer to the *Cisco IOS Security Configuration Guide*.
  - TCP Intercept. The TCP Intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of DoS attack. A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Like CBAC, the TCP

Intercept feature also uses timeout and threshold values to manage session state information, helping to determine when to drop sessions that do not become fully established. For more information on TCP Intercept, refer to the *Cisco IOS Security Configuration Guide*.

Prerequisites to Configuring Unicast RPF

Prior to configuring Unicast RPF, configure ACLs:

- Configure standard or extended ACLs to mitigate transmission of invalid IP addresses (perform egress filtering). Permit only valid source addresses to leave your network and get onto the Internet. Prevent all other source addresses from leaving your network for the Internet.
- Configure standard or extended ACLs entries to drop (deny) packets that have invalid source IP addresses (perform ingress filtering). Invalid source IP addresses include the following types:
  - Reserved addresses
  - Loopback addresses
  - Private addresses (RFC 1918, *Address Allocation for Private Internets*)
  - Broadcast addresses (including multicast addresses)
  - Source addresses that fall outside the range of valid addresses associated with the protected network

Unicast RPF Configuration Task List

The following sections describe the configuration tasks for Unicast RPF. Each task in the list is identified as either optional or required.

- Configuring Unicast RPF (Required)
- Verifying Unicast RPF (Optional)

See the section “Unicast RPF Configuration Examples” at the end of this chapter.

Configuring Unicast RPF

Unicast RPF is an input-side function that is enabled on an interface or subinterface that supports any type of encapsulation and operates on IP packets received by the router.

To configure Unicast RPF, perform the followin task:

	Command	Purpose
Step 1	`Router(config-if)# interface type`	Selects the input interface on which you want to apply Unicast RPF. This is the receiving interface, which allows Unicast RPF to verify the best return path before forwarding the packet on to the next destination. The interface type is specific to your router and the types of interface cards installed on the router. To display a list of available interface types, enter the **interface ?** command.
Step 2	`Router(config-if)# ip verify unicast source reachable-via rx allow-default`	Enables Unicast RPF on the interface.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode. Repeat Steps 2 and 3 for each interface on which you want to apply Unicast RPF.

Verifying Unicast RPF

To verify that Unicast RPF is operational, use the **show cef interface** command. The following example shows that Unicast RPF is enabled at interface Gigabit Ethernet 3/1.

```
Switch# show cef interface gigabitEthernet 3/1
GigabitEthernet3/1 is up (if_number 79)
  Corresponding hwidb fast_if_number 79
  Corresponding hwidb firstsw->if_number 79
  Internet address is 10.1.1.1/24
  ICMP redirects are always sent
  IP unicast RPF check is enabled <======
  Input features: uRPF <=====
  Inbound access list is not set
  Outbound access list is not set
  IP policy routing is disabled
  BGP based policy accounting on input is disabled
  BGP based policy accounting on output is disabled
  Hardware idb is GigabitEthernet3/1
  Fast switching type 1, interface type 155
  IP CEF switching enabled
  IP CEF switching turbo vector
  IP Null turbo vector
  IP prefix lookup IPv4 mtrie 8-8-8-8 optimized
  Input fast flags 0x4000, Output fast flags 0x0
  ifindex 78(78)
  Slot 3 Slot unit 1 VC -1
  Transmit limit accumulator 0x0 (0x0)
  IP MTU 1500
```

Monitoring and Maintaining Unicast RPF

This section describes commands used to monitor and maintain Unicast RPF.

Command	Purpose
`Router# show ip traffic`	Displays global router statistics about Unicast RPF drops and suppressed drops.
`Router(config-if)# no ip verify unicast`	Disables Unicast RPF at the interface.

Unicast RPF counts the number of packets dropped or suppressed because of malformed or forged source addresses. Unicast RPF counts dropped or forwarded packets that include the following global and per-interface information:

- Global Unicast RPF drops
- Per-interface Unicast RPF drops
- Per-interface Unicast RPF suppressed drops

The **show ip traffic** command shows the total number (global count) of dropped or suppressed packets for all interfaces on the router. The Unicast RPF drop count is included in the IP statistics section.

```
Router# show ip traffic

IP statistics:
  Rcvd:  1471590 total, 887368 local destination
         0 format errors, 0 checksum errors, 301274 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 other
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 couldn't fragment
  Bcast: 205233 received, 0 sent
  Mcast: 463292 received, 462118 sent
  Sent:  990158 generated, 282938 forwarded
  ! The second line below ("0 unicast RPF") displays Unicast RPF packet dropping
  information.
  Drop:  3 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop
```

A nonzero value for the count of dropped or suppressed packets can mean one of two things:

- Unicast RPF is dropping or suppressing packets that have a bad source address (normal operation).
- Unicast RPF is dropping or suppressing legitimate packets because the route is misconfigured to use Unicast RPF in environments where asymmetric routing exists; that is, where multiple paths can exist as the best return path for a source address.

The **show ip interface** command shows the total of dropped or suppressed packets at a specific interface. If Unicast RPF is configured to use a specific ACL, that ACL information is displayed along with the drop statistics.

```
Router> show ip interface ethernet0/1/1

   Unicast RPF ACL 197
   1 unicast RPF drop
   1 unicast RPF suppressed drop
```

The **show access-lists** command displays the number of matches found for a specific entry in a specific access list.

```
Router> show access-lists

Extended IP access list 197
     deny ip 192.168.201.0 0.0.0.63 any log-input (1 match)
     permit ip 192.168.201.64 0.0.0.63 any log-input (1 match)
     deny ip 192.168.201.128 0.0.0.63 any log-input
     permit ip 192.168.201.192 0.0.0.63 any log-input
```

Unicast RPF Configuration Examples

This section provides the following configuration examples:

- Unicast RPF on a Leased-Line Aggregation Router Example
- Unicast RPF on the Cisco AS5800 Using Dialup Ports Example
- Unicast RPF with Inbound and Outbound Filters Example
- Unicast RPF with ACLs and Logging Example

Unicast RPF on a Leased-Line Aggregation Router Example

The following commands enable Unicast RPF on a serial interface:

```
ip cef
! or "ip cef distributed" for RSP+VIP based routers
!
interface serial 5/0/0
 ip verify unicast reverse-path
```

Unicast RPF on the Cisco AS5800 Using Dialup Ports Example

The following example enables Unicast RPF on a Cisco AS5800. The **interface Group-Async** command makes it easy to apply Unicast RPF on all the dialup ports.

```
ip cef
!
interface Group-Async1
 ip verify unicast reverse-path
```

Unicast RPF with Inbound and Outbound Filters Example

The following example uses a very simple single-homed ISP to demonstrate the concepts of ingress and egress filters used in conjunction with Unicast RPF. The example illustrates an ISP-allocated classless interdomain routing (CIDR) block 209.165.202.128/28 that has both inbound and outbound filters on the upstream interface. Be aware that ISPs are usually not single-homed. Hence, provisions for asymmetrical flows (when outbound traffic goes out one link and returns via a different link) need to be designed into the filters on the border routers of the ISP.

```
ip cef distributed
!
interface Serial 5/0/0
 description Connection to Upstream ISP
 ip address 209.165.200.225 255.255.255.252
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip verify unicast reverse-path
 ip access-group 111 in
 ip access-group 110 out
!
access-list 110 permit ip 209.165.202.128 0.0.0.31 any
access-list 110 deny ip any any log
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 209.165.202.128 0.0.0.31 any log
access-list 111 permit ip any any
```

Unicast RPF with ACLs and Logging Example

The following example demonstrates the use of ACLs and logging with Unicast RPF. In this example, extended ACL 197 provides entries that deny or permit network traffic for specific address ranges. Unicast RPF is configured on interface Ethernet0 to check packets arriving at that interface.

For example, packets with a source address of 192.168.201.10 arriving at interface Ethernet0 are dropped because of the deny statement in ACL 197. In this case, the ACL information is logged (the logging option is turned on for the ACL entry) and dropped packets are counted per interface and globally. Packets with a source address of 192.168.201.100 arriving at interface Ethernet0 are forwarded because of the permit statement in ACL 197. ACL information about dropped or suppressed packets is logged (logging option turned on for the ACL entry) to the log server.

```
ip cef distributed
!
int eth0/1/1
 ip address 192.168.200.1 255.255.255.0
 ip verify unicast reverse-path 197
!
int eth0/1/2
 ip address 192.168.201.1 255.255.255.0
!
access-list 197 deny   ip 192.168.201.0 0.0.0.63 any log-input
access-list 197 permit ip 192.168.201.64 0.0.0.63 any log-input
access-list 197 deny   ip 192.168.201.128 0.0.0.63 any log-input
access-list 197 permit ip 192.168.201.192 0.0.0.63 any log-input
access-list 197 deny ip host 0.0.0.0 any log
```

CHAPTER 29

Configuring IP Multicast

This chapter describes IP multicast routing on the Catalyst 4500 series switch. It also provides procedures and examples to configure IP multicast routing.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Note For more detailed information on IP multicast, refer to the discussion at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fipr_c/ipcpt3/

This chapter includes the following major sections:

- Overview of IP Multicast, page 29-1
- Configuring IP Multicast Routing, page 29-12
- Monitoring and Maintaining IP Multicast Routing, page 29-15
- Configuration Examples, page 29-21

For information on how to troubleshoot PVLANs, refer to the “Troubleshooting IP Multicast” section on page 51-19.

Overview of IP Multicast

This section includes these subsections:

- IP Multicast Protocols, page 29-2
- IP Multicast on the Catalyst 4500 Series Switch, page 29-4
- Unsupported Features, page 29-12

At one end of the IP communication spectrum is IP unicast, where a source IP host sends packets to a specific destination IP host. In IP unicast, the destination address in the IP packet is the address of a single, unique host in the IP network. These IP packets are forwarded across the network from the source to the destination host by routers. At each point on the path between source and destination, a router uses a unicast routing table to make unicast forwarding decisions, based on the IP destination address in the packet.

At the other end of the IP communication spectrum is an IP broadcast, where a source host sends packets to all hosts on a network segment. The destination address of an IP broadcast packet has the host portion of the destination IP address set to all ones and the network portion set to the address of the subnet. IP hosts, including routers, understand that packets, which contain an IP broadcast address as the destination address, are addressed to all IP hosts on the subnet. Unless specifically configured otherwise, routers do not forward IP broadcast packets, so IP broadcast communication is normally limited to a local subnet.

IP multicasting falls between IP unicast and IP broadcast communication. IP multicast communication enables a host to send IP packets to a *group* of hosts anywhere within the IP network. To send information to a specific group, IP multicast communication uses a special form of IP destination address called an IP *multicast group address*. The IP multicast group address is specified in the IP destination address field of the packet.

To multicast IP information, Layer 3 switches and routers must forward an incoming IP packet to all output interfaces that lead to *members* of the IP multicast group. In the multicasting process on the Catalyst 4000 family switch, a packet is replicated in the Integrated Switching Engine, forwarded to the appropriate output interfaces, and sent to each member of the multicast group.

It is not uncommon for people to think of IP multicasting and video conferencing as almost the same thing. Although the first application in a network to use IP multicast is often video conferencing, video is only one of many IP multicast applications that can add value to a company’s business model. Other IP multicast applications that have potential for improving productivity include multimedia conferencing, data replication, real-time data multicasts, and simulation applications.

This section contains the following subsections:

- IP Multicast Protocols, page 29-2
- IP Multicast on the Catalyst 4500 Series Switch, page 29-4
- Unsupported Features, page 29-12

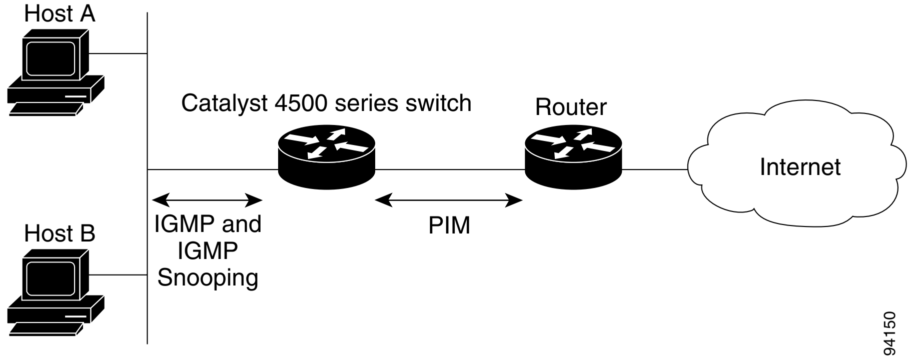
IP Multicast Protocols

The Catalyst 4000 family switch primarily uses these protocols to implement IP multicast routing:

- Internet Group Management Protocol (IGMP)
- Protocol Independent Multicast (PIM)
- IGMP snooping and Cisco Group Management Protocol

Figure 29-1 shows where these protocols operate within the IP multicast environment.

Figure 29-1 IP Multicast Routing Protocols



Internet Group Management Protocol

IGMP messages are used by IP multicast hosts to send their local Layer 3 switch or router a request to join a specific multicast group and begin receiving multicast traffic. With some extensions in IGMPv2, IP hosts can also send a request to a Layer 3 switch or router to leave an IP multicast group and not receive the multicast group traffic.

Using the information obtained via IGMP, a Layer 3 switch or router maintains a list of multicast group memberships on a per-interface basis. A multicast group membership is active on an interface if at least one host on the interface sends an IGMP request to receive multicast group traffic.

Protocol-Independent Multicast

PIM is *protocol independent* because it can leverage whichever unicast routing protocol is used to populate the unicast routing table, including EIGRP, OSPF, BGP, or static route, to support IP multicast. PIM also uses a unicast routing table to perform the reverse path forwarding (RPF) check function instead of building a completely independent multicast routing table. PIM does not send and receive multicast routing updates between routers like other routing protocols do.

PIM Dense Mode

PIM Dense Mode (PIM-DM) uses a *push* model to flood multicast traffic to every corner of the network. PIM-DM is intended for networks in which most LANs need to receive the multicast, such as LAN TV and corporate or financial information broadcasts. It can be an efficient delivery mechanism if there are active receivers on every subnet in the network.

PIM Sparse Mode

PIM Sparse Mode (PIM-SM) uses a *pull* model to deliver multicast traffic. Only networks with active receivers that have explicitly requested the data will be forwarded the traffic. PIM-SM is intended for networks with several different multicasts, such as desktop video conferencing and collaborative computing, that go to a small number of receivers and are typically in progress simultaneously.

For more detailed information on PIM Dense and Spare Mode, refer to this URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcpt3.

IGMP Snooping and CGMP

IGMP snooping is used for multicasting in a Layer 2 switching environment. With IGMP snooping, a Layer 3 switch or router examines Layer 3 information in the IGMP packets in transit between hosts and a router. When the switch receives the IGMP Host Report from a host for a particular multicast group, the switch adds the host's port number to the associated multicast table entry. When the switch receives the IGMP Leave Group message from a host, it removes the host's port from the table entry.

Because IGMP control messages are transmitted as multicast packets, they are indistinguishable from multicast data if only the Layer 2 header is examined. A switch running IGMP snooping examines every multicast data packet to determine whether it contains any pertinent IGMP control information. If IGMP snooping is implemented on a low end switch with a slow CPU, performance could be severely impacted when data is transmitted at high rates. On the Catalyst 4500 series switches, IGMP snooping is implemented in the forwarding ASIC, so it does not impact the forwarding rate.

Note A Catalyst 4000 family switch can act as a CGMP server for switches that do not support IGMP snooping, such as Catalyst 4500 family switches with Supervisor Engine I and Supervisor Engine II. You cannot configure the switch as a CGMP client. To configure a Catalyst 4000 family switch as a client, use IGMP snooping.

CGMP is a Cisco protocol that allows Catalyst switches to leverage IGMP information on Cisco routers to make Layer 2 forwarding decisions. CGMP is configured on the multicast routers and the Layer 2 switches. As a result, IP multicast traffic is delivered only to those Catalyst switchports with hosts that have requested the traffic. Switchports that have not explicitly requested the traffic will not receive it.

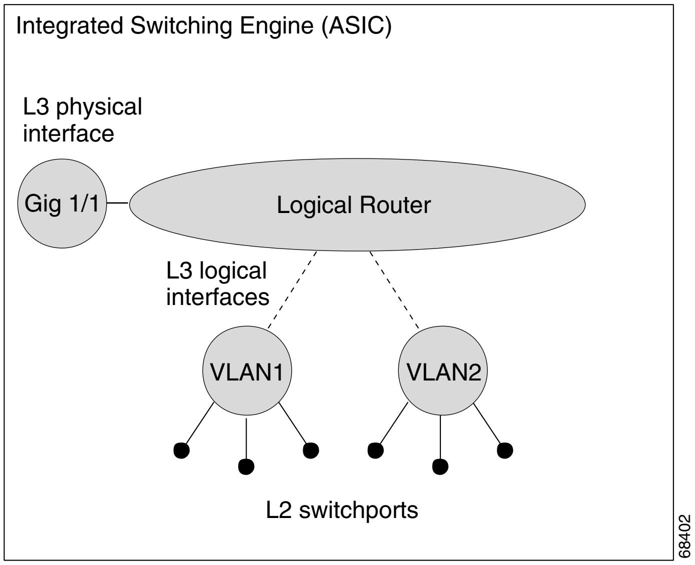
IP Multicast on the Catalyst 4500 Series Switch

The Catalyst 4000 family switch supports an ASIC-based Integrated Switching Engine that provides Ethernet bridging at Layer 2 and IP routing at Layer 3. Because the ASIC is specifically designed to forward packets, the Integrated Switching Engine hardware provides very high performance with ACLs and QoS enabled. At wire-speed, forwarding in hardware is significantly faster than the CPU subsystem software, which is designed to handle exception packets.

The Integrated Switching Engine hardware supports interfaces for inter-VLAN routing and switchports for Layer 2 bridging. It also provides a physical Layer 3 interface that can be configured to connect with a host, a switch, or a router.

Figure 29-2 shows a logical view of Layer 2 and Layer 3 forwarding in the Integrated Switching Engine hardware.

Figure 29-2 Logical View of Layer 2 and Layer 3 Forwarding in Hardware



This section contains the following subsections:

- CEF, MFIB, and Layer 2 Forwarding, page 29-5
- IP Multicast Tables, page 29-7
- Hardware and Software Forwarding, page 29-8
- Non-Reverse Path Forwarding Traffic, page 29-9
- Multicast Fast Drop, page 29-10
- Multicast Forwarding Information Base, page 29-11
- S/M, 224/4, page 29-12

CEF, MFIB, and Layer 2 Forwarding

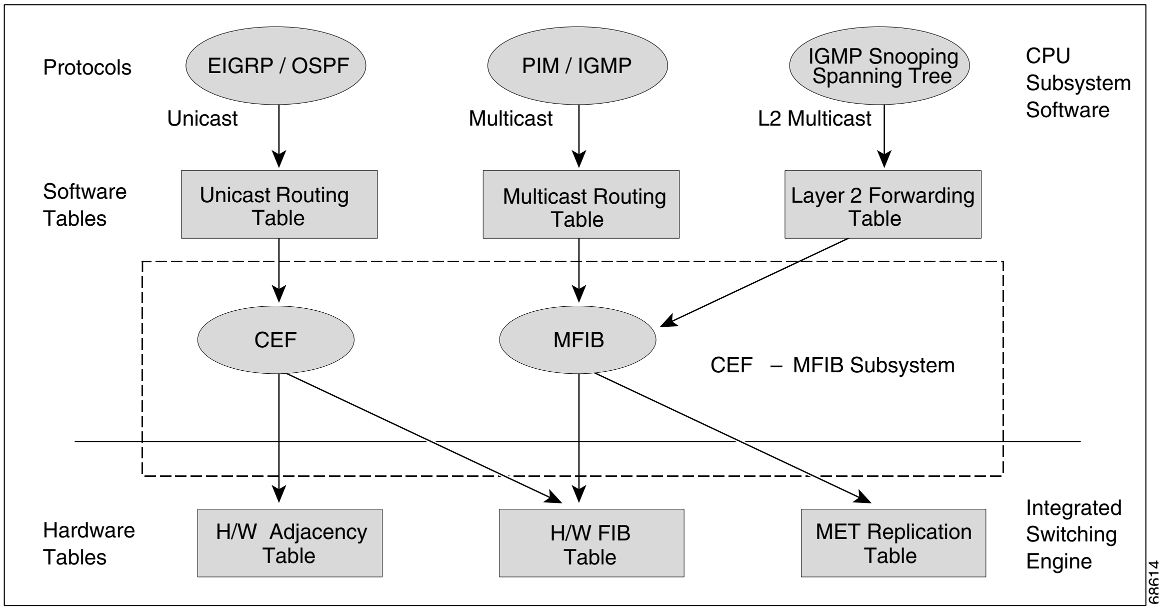
The implementation of IP multicast on the Catalyst 4000 family switch is an extension of centralized Cisco Express Forwarding (CEF). CEF extracts information from the unicast routing table, which is created by unicast routing protocols, such as BGP, OSPF, and EIGR and loads it into the hardware Forwarding Information Base (FIB). With the unicast routes in the FIB, when a route is changed in the upper-layer routing table, only one route needs to be changed in the hardware routing state. To forward unicast packets in hardware, the Integrated Switching Engine looks up source and destination routes in ternary content addressable memory (TCAM), takes the adjacency index from the hardware FIB, and gets the Layer 2 rewrite information and next-hop address from the hardware adjacency table.

The new Multicast Forwarding Information Base (MFIB) subsystem is the multicast analog of the unicast CEF. The MFIB subsystem extracts the multicast routes that PIM and IGMP create and refines them into a protocol-independent format for forwarding in hardware. The MFIB subsystem removes the protocol-specific information and leaves only the essential forwarding information. Each entry in the MFIB table consists of an (S,G) or (*,G) route, an input RPF VLAN, and a list of Layer 3 output interfaces. The MFIB subsystem, together with platform-dependent management software, loads this multicast routing information into the hardware FIB and hardware multicast expansion table (MET).

The Catalyst 4000 family switch performs Layer 3 routing and Layer 2 bridging at the same time. There can be multiple Layer 2 switchports on any VLAN interface. To determine the set of output switchports on which to forward a multicast packet, the Supervisor Engine III combines Layer 3 MFIB information with Layer 2 forwarding information and stores it in the hardware MET for packet replication.

Figure 29-3 shows a functional overview of how the Catalyst 4000 family switch combines unicast routing, multicast routing, and Layer 2 bridging information to forward in hardware.

Figure 29-3 Combining CEF, MFIB, and Layer 2 Forwarding Information in Hardware



Like the CEF unicast routes, the MFIB routes are Layer 3 and must be merged with the appropriate Layer 2 information. The following example shows an MFIB route:

```
(*,224.1.2.3)
RPF interface is Vlan3
Output Interfaces are:
Vlan 1
Vlan 2
```

The route (*,224.1.2.3) is loaded in the hardware FIB table and the list of output interfaces is loaded into the MET. A pointer to the list of output interfaces, the MET index, and the RPF interface are also loaded in the hardware FIB with the (*,224.1.2.3) route. With this information loaded in hardware, merging of the Layer 2 information can begin. For the output interfaces on VLAN1, the Integrated Switching Engine must send the packet to all switchports in VLAN1 that are in the spanning tree forwarding state. The same process applies to VLAN 2. To determine the set of switchports in VLAN 2, the Layer 2 Forwarding Table is used.

When the hardware routes a packet, in addition to sending it to all of the switchports on all output interfaces, the hardware also sends the packet to all switchports (other than the one it arrived on) in the input VLAN. For example, assume that VLAN 3 has two switchports in it, Gig 3/1 and Gig 3/2. If a host on Gig 3/1 sends a multicast packet, the host on Gig 3/2 might also need to receive the packet. To send a multicast packet to the host on Gig 3/2, all of the switchports in the ingress VLAN must be added to the portset that is loaded in the MET.

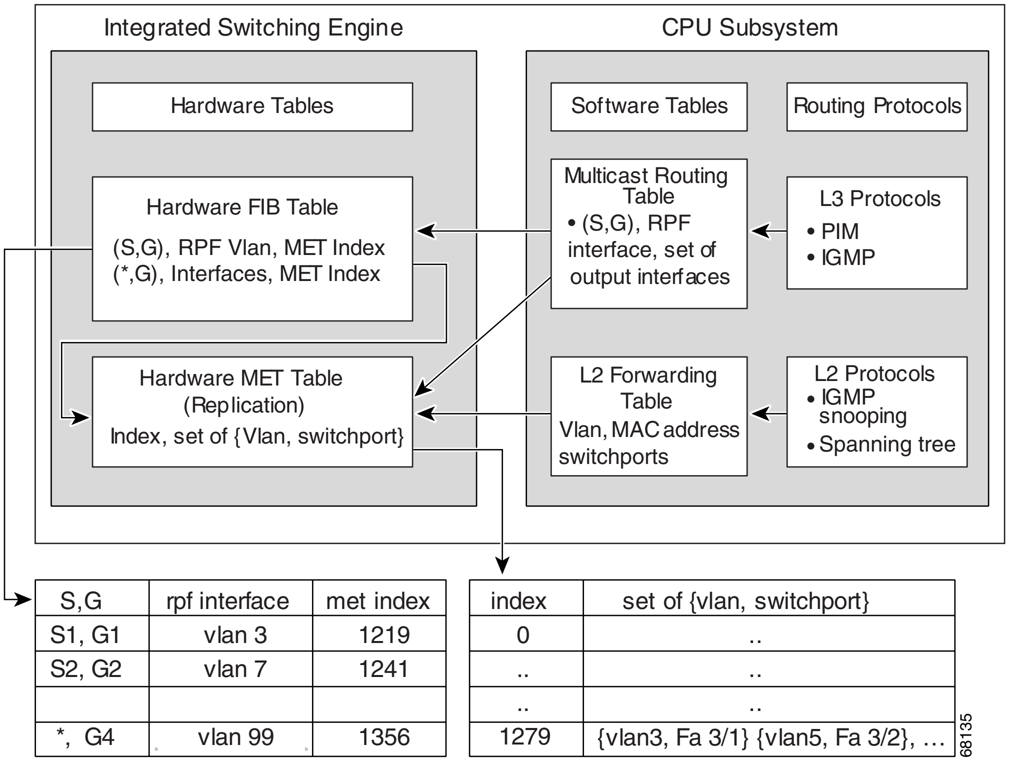
If VLAN 1 contains 1/1 and 1/2, VLAN 2 contains 2/1 and 2/2, and VLAN 3 contains 3/1 and 3/2, the MET chain for this route would contain these switchports: (1/1,1/2,2/1,2/2,3/1, and 3/2).

If IGMP snooping is on, the packet should not be forwarded to all output switchports on VLAN 2. The packet should be forwarded only to switchports where IGMP snooping has determined that there is either a group member or router. For example, if VLAN 1 had IGMP snooping enabled, and IGMP snooping determined that only port 1/2 had a group member on it, then the MET chain would contain these switchports: (1/1,1/2, 2/1, 2/2, 3/1, and 3/2).

IP Multicast Tables

Figure 29-4 shows some key data structures that the Catalyst 4000 family switch uses to forward IP multicast packets in hardware.

Figure 29-4 IP Multicast Tables and Protocols



S,G	rpf interface	met index
S1, G1	vlan 3	1219
S2, G2	vlan 7	1241
*, G4	vlan 99	1356

index	set of {vlan, switchport}
0	..
..	..
..	..
1279	{vlan3, Fa 3/1} {vlan5, Fa 3/2}, …

The Integrated Switching Engine maintains the hardware FIB table to identify individual IP multicast routes. Each entry consists of a destination group IP address and an optional source IP address. Multicast traffic flows on primarily two types of routes: (S,G) and (*,G). The (S,G) routes flow from a source to a group based on the IP address of the multicast source and the IP address of the multicast group destination. Traffic on a (*,G) route flows from the PIM RP to all receivers of group G. Only sparse-mode groups use (*,G) routes. The Integrated Switching Engine hardware contains space for a total of 128,000 routes, which are shared by unicast routes, multicast routes, and multicast fast-drop entries.

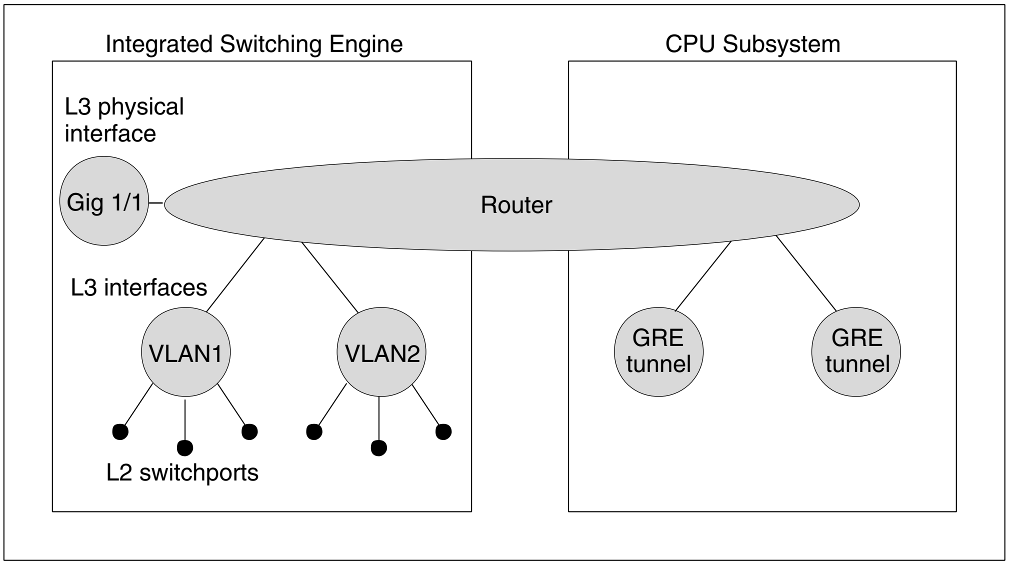
Output interface lists are stored in the multicast expansion table (MET). The MET has room for up to 32,000 output interface lists. The MET resources are shared by both Layer 3 multicast routes and by Layer 2 multicast entries. The actual number of output interface lists available in hardware depends on the specific configuration. If the total number of multicast routes exceed 32,000, multicast packets might not be switched by the Integrated Switching Engine. They would be forwarded by the CPU subsystem at much slower speeds.

Hardware and Software Forwarding

The Integrated Switching Engine forwards the majority of packets in hardware at very high rates of speed. The CPU subsystem forwards exception packets in software. Statistical reports should show that the Integrated Switching Engine is forwarding the vast majority of packets in hardware.

Figure 29-5 shows a logical view of the hardware and software forwarding components.

Figure 29-5 Hardware and Software Forwarding Components



In the normal mode of operation, the Integrated Switching Engine performs inter-VLAN routing in hardware. The CPU subsystem supports generic routing encapsulation (GRE) tunnels for forwarding in software.

Replication is a particular type of forwarding where, instead of sending out one copy of the packet, the packet is replicated and multiple copies of the packet are sent out. At Layer 3, replication occurs only for multicast packets; unicast packets are never replicated to multiple Layer 3 interfaces. In IP multicasting, for each incoming IP multicast packet that is received, many replicas of the packet are sent out.

IP multicast packets can be transmitted on the following types of routes:

- Hardware routes
- Software routes
- Partial routes

Hardware routes occur when the Integrated Switching Engine hardware forwards all replicas of a packet. Software routes occur when the CPU subsystem software forwards all replicas of a packet. Partial routes occur when the Integrated Switching Engine forwards some of the replicas in hardware and the CPU subsystem forwards some of the replicas in software.

Partial Routes

Note The conditions listed below cause the replicas to be forwarded by the CPU subsystem software, but the performance of the replicas that are forwarded in hardware is not affected.

The following conditions cause some replicas of a packet for a route to be forwarded by the CPU subsystem:

- The switch is configured with the **ip igmp join-group** command as a member of the IP multicast group on the RPF interface of the multicast source.
- The switch is the first-hop to the source in PIM sparse mode. In this case, the switch must send PIM-register messages to the RP.

Software Routes

Note If any one of the following conditions is configured on the RPF interface or the output interface, all replication of the output is performed in software.

The following conditions cause all replicas of a packet for a route to be forwarded by the CPU subsystem software:

- The interface is configured with multicast helper.
- The interface is a generic routing encapsulation (GRE) or Distance Vector Multicast Routing Protocol (DVMRP) tunnel.
- The interface uses non-Advanced Research Products Agency (ARPA) encapsulation.

The following packets are always forwarded in software:

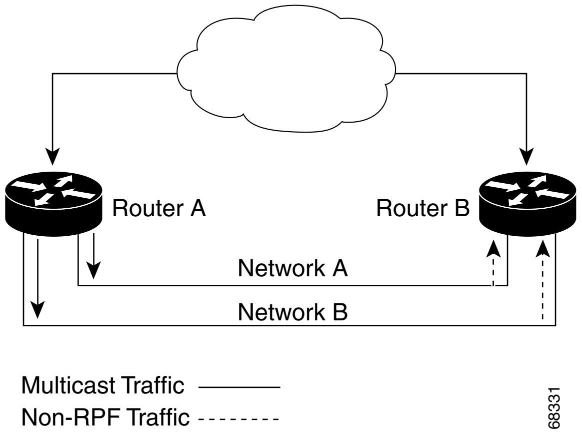
- Packets sent to multicast groups that fall into the range 224.0.0.* (where * is in the range from 0 to 255). This range is used by routing protocols. Layer 3 switching supports all other multicast group addresses.
- Packets with IP options.

Non-Reverse Path Forwarding Traffic

Traffic that fails an Reverse Path Forwarding (RPF) check is called non-RPF traffic. Non-RPF traffic is forwarded by the Integrated Switching Engine by filtering (persistently dropping) or rate limiting the non-RPF traffic.

In a redundant configuration where multiple Layer 3 switches or routers connect to the same LAN segment, only one device forwards the multicast traffic from the source to the receivers on the outgoing interfaces. Figure 29-6 shows how Non-RPF traffic can occur in a common network configuration.

Figure 29-6 Redundant Multicast Router Configuration in a Stub Network



In this kind of topology, only Router A, the PIM designated router (PIM DR), forwards data to the common VLAN. Router B receives the forwarded multicast traffic, but 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.

Multicast Fast Drop

In IP multicast protocols, such as PIM-SM and PIM-DM, every (S,G) or (*,G) route has an incoming interface associated with it. This interface is referred to as the reverse path forwarding interface. In some cases, when a packet arrives on an interface other than the expected RPF interface, the packet must be forwarded to the CPU subsystem software to allow PIM to perform special protocol processing on the packet. One example of this special protocol processing that PIM performs is the PIM Assert protocol.

By default, the Integrated Switching Engine hardware sends all packets that arrive on a non-RPF interface to the CPU subsystem software. However, processing in software is not necessary in many cases, because these non-RPF packets are often not needed by the multicast routing protocols. The problem is that if no action is taken, the non-RPF packets that are sent to the software can overwhelm the CPU.

Use the **ip mfib fastdrop** command to enable or disable MFIB fast drops.

To prevent this from happening, the CPU subsystem software loads fast-drop entries in the hardware when it receives an RPF failed packet that is not needed by the PIM protocols running on the switch. A fast-drop entry is keyed by (S,G, incoming interface). Any packet matching a fast-drop entry is bridged in the ingress VLAN, but is not sent to the software, so the CPU subsystem software is not overloaded by processing these RPF failures unnecessarily.

Protocol events, such as a link going down or a change in the unicast routing table, can impact the set of packets that can safely be fast dropped. A packet that was correctly fast dropped before might, after a topology change, need to be forwarded to the CPU subsystem software so that PIM can process it. The CPU subsystem software handles flushing fast-drop entries in response to protocol events so that the PIM code in IOS can process all the necessary RPF failures.

The use of fast-drop entries in the hardware is critical in some common topologies because it is possible to have persistent RPF failures. Without the fast-drop entries, the CPU would be exhausted by RPF failed packets that it did not need to process.

Multicast Forwarding Information Base

The Multicast Forwarding Information Base (MFIB) subsystem supports IP multicast routing in the Integrated Switching Engine hardware on the Catalyst 4000 family switch. The MFIB logically resides between the IP multicast routing protocols in the CPU subsystem software (PIM, IGMP, MSDP, MBGP, and DVMRP) and the platform-specific code that manages IP multicast routing in hardware. The MFIB translates the routing table information created by the multicast routing protocols into a simplified format that can be efficiently processed and used for forwarding by the Integrated Switching Engine hardware.

To display the information in the multicast routing table, use the **show ip mroute** command. To display the MFIB table information, use the **show ip mfib** command.

Note In Supervisor Engine 6-E systems, the output of the **show ip mfib** command does not display any hardware counters.

The MFIB table contains a set of IP multicast routes. There are several types of IP multicast routes, including (S,G) and (*,G) routes. Each route in the MFIB table can have one or more optional flags associated with it. The route flags indicate how a packet that matches a route should be forwarded. For example, the Internal Copy (IC) flag on an MFIB route indicates that a process on the switch needs to receive a copy of the packet. The following flags can be associated with MFIB routes:

- Internal Copy (IC) flag—set on a route when a process on the router needs to receive a copy of all packets matching the specified route
- Signalling (S) flag—set on a route when a process needs to be notified when a packet matching the route is received; the expected behavior is that the protocol code updates the MFIB state in response to receiving a packet on a signalling interface
- Connected (C) flag—–when set on an MFIB route, has the same meaning as the Signalling (S) flag, except that the C flag indicates that only packets sent by directly connected hosts to the route should be signalled to a protocol process

A route can also have a set of optional flags associated with one or more interfaces. For example, an (S,G) route with the flags on VLAN 1 indicates how packets arriving on VLAN 1 should be treated, and they also indicate whether packets matching the route should be forwarded onto VLAN 1. The per-interface flags supported in the MFIB include the following:

- Accepting (A)—set on the interface that is known in multicast routing as the RPF interface. A packet that arrives on an interface that is marked as Accepting (A) is forwarded to all Forwarding (F) interfaces.
- Forwarding (F)—used in conjunction with the Accepting (A) flag as described above. The set of Forwarding interfaces that form what is often referred to as the multicast "olist" or output interface list.
- Signalling (S)—set on an interface when some multicast routing protocol process in IOS needs to be notified of packets arriving on that interface.
- Not platform fast-switched (NP)—used in conjunction with the Forwarding (F) flag. A Forwarding interface is also marked as not platform fast-switched whenever that output interface cannot be fast switched by the platform. The NP flag is typically used when the Forwarding interface cannot be routed in hardware and requires software forwarding. For example, Catalyst 4000 family switch tunnel interfaces are not hardware switched, so they are marked with the NP flag. If there are any NP interfaces associated with a route, then for every packet arriving on an Accepting interface, one copy of that packet is sent to the software forwarding path for software replication to those interfaces that were not switched in hardware.

Note When PIM-SM routing is in use, the MFIB route might include an interface like in this example: PimTunnel [1.2.3.4]. This is a virtual interface that the MFIB subsystem creates to indicate that packets are being tunnelled to the specified destination address. A PimTunnel interface cannot be displayed with the normal **show interface** command.

S/M, 224/4

An (S/M, 224/4) entry is created in the MFIB for every multicast-enabled interface. This entry ensures that all packets sent by directly connected neighbors can be Register-encapsulated to the PIM-SM RP. Typically, only a small number of packets would be forwarded using the (S/M,224/4) route, until the (S,G) route is established by PIM-SM.

For example, on an interface with IP address 10.0.0.1 and netmask 255.0.0.0, a route would be created matching all IP multicast packets in which the source address is anything in the class A network 10. This route can be written in conventional subnet/masklength notation as (10/8,224/4). If an interface has multiple assigned IP addresses, then one route is created for each such IP address.

Unsupported Features

The following IP multicast features are not supported in this release:

- Controlling the transmission rate to a multicast group
- Load splitting IP multicast traffic across equal-cost paths

Configuring IP Multicast Routing

The following sections describe IP multicast routing configuration tasks:

- Default Configuration in IP MUlticast Routing, page 29-13
- Enabling IP Multicast Routing, page 29-13
- Enabling PIM on an Interface, page 29-13

For more detailed information on IP multicast routing, such as Auto-RP, PIM Version 2, and IP multicast static routes, refer to the *Cisco IOS IP and IP Routing Configuration Guide, Cisco IOS Release 12.3.*

Default Configuration in IP MUlticast Routing

Table 29-1 shows the IP multicast default configuration.

Table 29-1 Default IP Multicast Configuration

Feature	Default Value
Rate limiting of RPF	Enabled globally
IP multicast routing	Disabled globally **Note** When IP multicast routing is disabled, IP multicast traffic data packets are not forwarded by the Catalyst 4000 family switch. However, IP multicast control traffic will continue to be processed and forwarded. Therefore, IP multicast routes can remain in the routing table even if IP multicast routing is disabled.
PIM	Disabled on all interfaces
IGMP snooping	Enabled on all VLAN interfaces **Note** If you disable IGMP snooping on an interface, all output ports are forwarded by the Integrated Switching Engine. When IGMP snooping is disabled on an input VLAN interface, multicast packets related to that interface are sent to all forwarding switchports in the VLAN.

Note Source-specific multicast and IGMP v3 are supported.

For more information about source-specific multicast with IGMPv3 and IGMP, see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcpt3/1cfssm.htm

Enabling IP Multicast Routing

Enabling IP multicast routing allows the Catalyst 4000 family switch to forward multicast packets. To enable IP multicast routing on the router, perform this task in global configuration mode:

Command	Purpose
`Switch(config)# ip multicast-routing`	Enables IP multicast routing.

Enabling PIM on an Interface

Enabling PIM on an interface also enables IGMP operation on that interface. An interface can be configured to be in dense mode, sparse mode, or sparse-dense mode. The mode determines how the Layer 3 switch or router populates its multicast routing table and how the Layer 3 switch or router forwards multicast packets it receives from its directly connected LANs. You must enable PIM in one of these modes for an interface to perform IP multicast routing.

When the switch populates the multicast routing table, dense-mode interfaces are always added to the table. Sparse-mode interfaces are added to the table only when periodic join messages are received from downstream routers, or when there is a directly connected member on the interface. When forwarding from a LAN, sparse-mode operation occurs if there is an RP known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense-mode fashion. If the multicast traffic from a specific source is sufficient, the receiver's first-hop router can send join messages toward the source to build a source-based distribution tree.

There is no default mode setting. By default, multicast routing is disabled on an interface.

Enabling Dense Mode

To configure PIM on an interface to be in dense mode, perform this task:

Command	Purpose
`Switch(config-if)# ip pim dense-mode`	Enables dense-mode PIM on the interface.

See the "PIM Dense Mode Example" section at the end of this chapter for an example of how to configure a PIM interface in dense mode.

Enabling Sparse Mode

To configure PIM on an interface to be in sparse mode, perform this task:

Command	Purpose
`Switch(config-if)# ip pim sparse-mode`	Enables sparse-mode PIM on the interface.

See the "PIM Sparse Mode Example" section at the end of this chapter for an example of how to configure a PIM interface in sparse mode.

Enabling Sparse-Dense Mode

When you enter either the **ip pim sparse-mode** or **ip pim dense-mode** command, sparseness or denseness is applied to the interface as a whole. However, some environments might require PIM to run in a single region in sparse mode for some groups and in dense mode for other groups.

An alternative to enabling only dense mode or only sparse mode is to enable sparse-dense mode. In this case, the interface is treated as dense mode if the group is in dense mode; the interface is treated in sparse mode if the group is in sparse mode. If you want to treat the group as a sparse group, and the interface is in sparse-dense mode, you must have an RP.

If you configure sparse-dense mode, the idea of sparseness or denseness is applied to the group on the switch, and the network manager should apply the same concept throughout the network.

Another benefit of sparse-dense mode is that Auto-RP information can be distributed in a dense-mode manner; yet, multicast groups for user groups can be used in a sparse-mode manner. Thus, there is no need to configure a default RP at the leaf routers.

When an interface is treated in dense mode, it is populated in a multicast routing table's outgoing interface list when either of the following is true:

- When there are members or DVMRP neighbors on the interface
- When there are PIM neighbors and the group has not been pruned

When an interface is treated in sparse mode, it is populated in a multicast routing table's outgoing interface list when either of the following is true:

- When there are members or DVMRP neighbors on the interface
- When an explicit join has been received by a PIM neighbor on the interface

To enable PIM to operate in the same mode as the group, perform this task:

Command	Purpose
`Switch(config-if)# ip pim sparse-dense-mode`	Enables PIM to operate in sparse or dense mode, depending on the group.

Monitoring and Maintaining IP Multicast Routing

You can remove all contents of a particular cache, table, or database. You also can display specific statistics. The following sections describe how to monitor and maintain IP multicast:

- Displaying System and Network Statistics, page 29-15
- Displaying the Multicast Routing Table, page 29-16
- Displaying IP MFIB, page 29-18
- Displaying IP MFIB Fast Drop, page 29-19
- Displaying PIM Statistics, page 29-20
- Clearing Tables and Databases, page 29-20

Displaying System and Network Statistics

You can display specific statistics, such as the contents of IP routing tables and databases. Information provided can be used to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path your device's packets are taking through the network.

To display various routing statistics, you can perform any of these tasks:

Command	Purpose
Switch# **ping** [*group-name* | *group-address*]	Sends an ICMP Echo Request to a multicast group address.
Switch# **show ip mroute** [*hostname* | *group_number*]	Displays the contents of the IP multicast routing table.
Switch# **show ip pim interface** [*type number*] [**count**]	Displays information about interfaces configured for PIM.
Switch# **show ip interface**	Displays PIM information for all interfaces.

Displaying the Multicast Routing Table

The following is sample output from the **show ip mroute** command for a router operating in dense mode. This command displays the contents of the IP multicast FIB table for the multicast group named cbone-audio.

```
Switch# show ip mroute cbone-audio

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
R - RP-bit set, F - Register flag, T - SPT-bit set
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(*, 224.0.255.1), uptime 0:57:31, expires 0:02:59, RP is 0.0.0.0, flags: DC
    Incoming interface: Null, RPF neighbor 0.0.0.0, Dvmrp
    Outgoing interface list:
    Ethernet0, Forward/Dense, 0:57:31/0:02:52
    Tunnel0, Forward/Dense, 0:56:55/0:01:28

(198.92.37.100/32, 224.0.255.1), uptime 20:20:00, expires 0:02:55, flags: C
    Incoming interface: Tunnel0, RPF neighbor 10.20.37.33, Dvmrp
    Outgoing interface list:
    Ethernet0, Forward/Dense, 20:20:00/0:02:52
```

The following is sample output from the **show ip mroute** command for a router operating in sparse mode:

```
Switch# show ip mroute

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
R - RP-bit set, F - Register flag, T - SPT-bit set
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(*, 224.0.255.3), uptime 5:29:15, RP is 198.92.37.2, flags: SC
  Incoming interface: Tunnel0, RPF neighbor 10.3.35.1, Dvmrp
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 5:29:15/0:02:57

(198.92.46.0/24, 224.0.255.3), uptime 5:29:15, expires 0:02:59, flags: C
  Incoming interface: Tunnel0, RPF neighbor 10.3.35.1
  Outgoing interface list:
    Ethernet0, Forward/Sparse, 5:29:15/0:02:57
```

Note Interface timers are not updated for hardware-forwarded packets. Entry timers are updated approximately every five seconds.

The following is sample output from the **show ip mroute** command with the **summary** keyword:

```
Switch# show ip mroute summary

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(*, 224.255.255.255), 2d16h/00:02:30, RP 171.69.10.13, flags: SJPC
(*, 224.2.127.253), 00:58:18/00:02:00, RP 171.69.10.13, flags: SJC

(*, 224.1.127.255), 00:58:21/00:02:03, RP 171.69.10.13, flags: SJC

(*, 224.2.127.254), 2d16h/00:00:00, RP 171.69.10.13, flags: SJCL
  (128.9.160.67/32, 224.2.127.254), 00:02:46/00:00:12, flags: CLJT
  (129.48.244.217/32, 224.2.127.254), 00:02:15/00:00:40, flags: CLJT
  (130.207.8.33/32, 224.2.127.254), 00:00:25/00:02:32, flags: CLJT
  (131.243.2.62/32, 224.2.127.254), 00:00:51/00:02:03, flags: CLJT
  (140.173.8.3/32, 224.2.127.254), 00:00:26/00:02:33, flags: CLJT
  (171.69.60.189/32, 224.2.127.254), 00:03:47/00:00:46, flags: CLJT
```

The following is sample output from the **show ip mroute** command with the **active** keyword:

```
Switch# show ip mroute active

Active IP Multicast Sources - sending >= 4 kbps

Group: 224.2.127.254, (sdr.cisco.com)
   Source: 146.137.28.69 (mbone.ipd.anl.gov)
     Rate: 1 pps/4 kbps(1sec), 4 kbps(last 1 secs), 4 kbps(life avg)

Group: 224.2.201.241, ACM 97
   Source: 130.129.52.160 (webcast3-e1.acm97.interop.net)
     Rate: 9 pps/93 kbps(1sec), 145 kbps(last 20 secs), 85 kbps(life avg)

Group: 224.2.207.215, ACM 97
   Source: 130.129.52.160 (webcast3-e1.acm97.interop.net)
     Rate: 3 pps/31 kbps(1sec), 63 kbps(last 19 secs), 65 kbps(life avg)
```

The following is sample output from the **show ip mroute** command with the **count** keyword:

```
Switch# show ip mroute count

IP Multicast Statistics - Group count: 8, Average sources per group: 9.87
Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second

Group: 224.255.255.255, Source count: 0, Group pkt count: 0
  RP-tree: 0/0/0/0

Group: 224.2.127.253, Source count: 0, Group pkt count: 0
  RP-tree: 0/0/0/0

Group: 224.1.127.255, Source count: 0, Group pkt count: 0
  RP-tree: 0/0/0/0

Group: 224.2.127.254, Source count: 9, Group pkt count: 14
  RP-tree: 0/0/0/0
  Source: 128.2.6.9/32, 2/0/796/0
  Source: 128.32.131.87/32, 1/0/616/0
  Source: 128.125.51.58/32, 1/0/412/0
  Source: 130.207.8.33/32, 1/0/936/0
  Source: 131.243.2.62/32, 1/0/750/0
  Source: 140.173.8.3/32, 1/0/660/0
  Source: 146.137.28.69/32, 1/0/584/0
  Source: 171.69.60.189/32, 4/0/447/0
  Source: 204.162.119.8/32, 2/0/834/0

Group: 224.0.1.40, Source count: 1, Group pkt count: 3606
  RP-tree: 0/0/0/0
  Source: 171.69.214.50/32, 3606/0/48/0, RPF Failed: 1203
```

```
Group: 224.2.201.241, Source count: 36, Group pkt count: 54152
  RP-tree: 7/0/108/0
  Source: 13.242.36.83/32, 99/0/123/0
  Source: 36.29.1.3/32, 71/0/110/0
  Source: 128.9.160.96/32, 505/1/106/0
  Source: 128.32.163.170/32, 661/1/88/0
  Source: 128.115.31.26/32, 192/0/118/0
  Source: 128.146.111.45/32, 500/0/87/0
  Source: 128.183.33.134/32, 248/0/119/0
  Source: 128.195.7.62/32, 527/0/118/0
  Source: 128.223.32.25/32, 554/0/105/0
  Source: 128.223.32.151/32, 551/1/125/0
  Source: 128.223.156.117/32, 535/1/114/0
  Source: 128.223.225.21/32, 582/0/114/0
  Source: 129.89.142.50/32, 78/0/127/0
  Source: 129.99.50.14/32, 526/0/118/0
  Source: 130.129.0.13/32, 522/0/95/0
  Source: 130.129.52.160/32, 40839/16/920/161
  Source: 130.129.52.161/32, 476/0/97/0
  Source: 130.221.224.10/32, 456/0/113/0
  Source: 132.146.32.108/32, 9/1/112/0
```

Note Multicast route byte and packet statistics are supported only for the first 1024 multicast routes. Output interface statistics are not maintained.

Displaying IP MFIB

You can display all routes in the MFIB, including routes that might not exist directly in the upper-layer routing protocol database but that are used to accelerate fast switching. These routes appear in the MFIB, even if dense-mode forwarding is in use.

To display various MFIB routing routes, perform one of these tasks:

Command	Purpose
`Switch# show ip mfib`	Displays the (S,G) and (*,G) routes that are used for packet forwarding. Displays counts for fast, slow, and partially-switched packets for every multicast route.
`Switch# show ip mfib all`	Displays all routes in the MFIB, including routes that may not exist directly in the upper-layer routing protocol database, but that are used to accelerate fast switching.These routes include the (S/M,224/4) routes.
`Switch# show ip mfib log [n]`	Displays a log of the most recent n MFIB related events, most recent first.
`Switch# show ip mfib counters`	Displays counts of MFIB related events. Only non-zero counters are shown.

The following is sample output from the **show ip mfib** command.

```
IP Multicast Forwarding Information Base
Entry Flags: C - Directly Connected, S - Signal,
             IC - Internal Copy
Interface Flags: A - Accept, F - Forward, S - Signal,
             NP - Not platform switched
Packets: Fast/Partial/Slow Bytes: Fast/Partial/Slow:
(171.69.10.13, 224.0.1.40), flags (IC)
   Packets: 2292/2292/0, Bytes: 518803/0/518803
   Vlan7 (A)
   Vlan100 (F NS)
   Vlan105 (F NS)
(*, 224.0.1.60), flags ()
   Packets: 2292/0/0, Bytes: 518803/0/0
   Vlan7 (A NS)
(*, 224.0.1.75), flags ()
   Vlan7 (A NS)
(10.34.2.92, 239.192.128.80), flags ()
   Packets: 24579/100/0, 2113788/15000/0 bytes
   Vlan7 (F NS)
   Vlan100 (A)
(*, 239.193.100.70), flags ()
   Packets: 1/0/0, 1500/0/0 bytes
   Vlan7 (A)
..
```

The fast-switched packet count represents the number of packets that were switched in hardware on the corresponding route.

The partially switched packet counter represents the number of times that a fast-switched packet was also copied to the CPU for software processing or for forwarding to one or more non-platform switched interfaces (such as a PimTunnel interface).

The slow-switched packet count represents the number of packets that were switched completely in software on the corresponding route.

Displaying IP MFIB Fast Drop

Note Supervisor Engine 6-E does *not* support the **show ip mfib fastdrop** command.

To display fast-drop entries, perform this task:

Command	Purpose
Switch# **show ip mfib fastdrop**	Displays all currently active fast-drop entries and indicates whether **fastdrop** is enabled.

The following is sample output from the **show ip mfib fastdrop** command.

```
Switch> show ip mfib fastdrop
MFIB fastdrop is enabled.
MFIB fast-dropped flows:
(10.0.0.1, 224.1.2.3, Vlan9 ) 00:01:32
(10.1.0.2, 224.1.2.3, Vlan9 ) 00:02:30
(1.2.3.4, 225.6.7.8, Vlan3) 00:01:50
```

The full (S,G) flow and the ingress interface on which incoming packets are dropped is shown. The timestamp indicates the age of the entry.

Displaying PIM Statistics

The following is sample output from the **show ip pim interface** command:

```
Switch# show ip pim interface

Address          Interface          Mode    Neighbor  Query     DR
                                            Count     Interval
198.92.37.6      Ethernet0          Dense   2         30        198.92.37.33
198.92.36.129    Ethernet1          Dense   2         30        198.92.36.131
10.1.37.2        Tunnel0            Dense   1         30        0.0.0.0
```

The following is sample output from the **show ip pim interface** command with a **count**:

```
Switch# show ip pim interface count

Address          Interface          FS  Mpackets In/Out
171.69.121.35    Ethernet0          *   548305239/13744856
171.69.121.35    Serial0.33         *   8256/67052912
198.92.12.73     Serial0.1719       *   219444/862191
```

The following is sample output from the **show ip pim interface** command with a **count** when IP multicast is enabled. The example lists the PIM interfaces that are fast-switched and process-switched, and the packet counts for these. The H is added to interfaces where IP multicast is enabled.

```
Switch# show ip pim interface count

States: FS - Fast Switched, H - Hardware Switched
Address          Interface          FS  Mpackets In/Out
192.1.10.2       Vlan10             * H 40886/0
192.1.11.2       Vlan11             * H 0/40554
192.1.12.2       Vlan12             * H 0/40554
192.1.23.2       Vlan23             *   0/0
192.1.24.2       Vlan24             *   0/0
```

Clearing Tables and Databases

You can remove all contents of a particular cache, table, or database. Clearing a cache, table, or database might be necessary when the contents of the particular structure have become, or are suspected to be, invalid.

To clear IP multicast caches, tables, and databases, perform one of these tasks:

Command	Purpose
`Switch# clear ip mroute`	Deletes entries from the IP routing table.
`Switch# clear ip mfib counters`	Deletes all per-route and global MFIB counters.
`Switch# clear ip mfib fastdrop`	Deletes all fast-drop entries.

Note IP multicast routes can be regenerated in response to protocol events and as data packets arrive.

Configuration Examples

The following sections provide IP multicast routing configuration examples:

- PIM Dense Mode Example, page 29-21
- PIM Sparse Mode Example, page 29-21
- BSR Configuration Example, page 29-21

PIM Dense Mode Example

This example is a configuration of dense-mode PIM on an Ethernet interface:

```
ip multicast-routing
interface ethernet 0
 ip pim dense-mode
```

PIM Sparse Mode Example

This example is a configuration of sparse-mode PIM. The RP router is the router with the address 10.8.0.20.

```
ip multicast-routing
 ip pim rp-address 10.8.0.20 1
interface ethernet 1
 ip pim sparse-mode
```

BSR Configuration Example

This example is a configuration of a candidate BSR, which also happens to be a candidate RP:

```
version 11.3
!
ip multicast-routing
!
interface Ethernet0
 ip address 171.69.62.35 255.255.255.240
!
interface Ethernet1
 ip address 172.21.24.18 255.255.255.248
 ip pim sparse-dense-mode
!
interface Ethernet2
 ip address 172.21.24.12 255.255.255.248
 ip pim sparse-dense-mode
!
router ospf 1
 network 172.21.24.8 0.0.0.7 area 1
 network 172.21.24.16 0.0.0.7 area 1
!
ip pim bsr-candidate Ethernet2 30 10
ip pim rp-candidate Ethernet2 group-list 5
access-list 5 permit 239.255.2.0 0.0.0.255
```

CHAPTER 30

Configuring Policy-Based Routing

Note PBR is *not* supported on Supervisor Engine 6-E.

This chapter describes the tasks for configuring policy-based routing (PBR) on a router and includes these major sections:

- Overview of Policy-Based Routing, page 30-1
- Policy-Based Routing Configuration Task List, page 30-3
- Policy-Based Routing Configuration Examples, page 30-5

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Note To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Overview of Policy-Based Routing

This section contains the following subsections:

- Understanding PBR, page 30-2
- Understanding PBR Flow Switching, page 30-2
- Using Policy-Based Routing, page 30-2

PBR gives you a flexible means of routing packets by allowing you to configure a defined policy for traffic flows, lessening reliance on routes derived from routing protocols. To this end, PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.

You can set up PBR as a way to route packets based on configured policies. For example, you can implement routing policies to allow or deny paths based on the identity of a particular end system, an application protocol, or the size of packets.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists, then establish the match criteria.
- Route packets to specific traffic-engineered paths.

Policies can be based on IP address, port numbers, or protocols. For a simple policy, you can use any one of these descriptors; for a complicated policy, you can use all of them.

Understanding PBR

All packets received on an interface with PBR enabled are passed through enhanced packet filters known as route maps. The route maps used by PBR dictate the policy, determining to where the packets are forwarded.

Route maps are composed of statements. The route map statements can be marked as permit or deny, and they are interpreted in the following ways:

- If a statement is marked as deny, the packets meeting the match criteria are sent back through the normal forwarding channels and destination-based routing is performed.
- If the statement is marked as permit and a packet matches the access-lists, then the first valid set clause is applied to that packet.

You specify PBR on the incoming interface (the interface on which packets are received), not outgoing interface.

Understanding PBR Flow Switching

The Catalyst 4500 switching engine supports matching a “set next-hop” route-map action with a packet on a permit ACL. All other route-map actions, as well as matches of deny ACLs, are supported by a flow switching model. In this model, the first packet on a flow that matches a route-map is delivered to the software for forwarding. Software determines the correct destination for the packet and installs an entry into the TCAM so that future packets on that flow are switched in hardware. The Catalyst 4500 switching engine supports a maximum of 4096 flows.

Using Policy-Based Routing

You can enable PBR to change the routing path of certain packets from the obvious shortest path. For example, PBR can be used to provide the following functionality:

- equal access
- protocol-sensitive routing
- source-sensitive routing
- routing based on interactive versus batch traffic
- routing based on dedicated links

Some applications or traffic can benefit from source-specific routing; for example, you can transfer stock records to a corporate office on a higher-bandwidth, higher-cost link for a short time while sending routine application data, such as e-mail, over a lower-bandwidth, lower-cost link.

Policy-Based Routing Configuration Task List

To configure PBR, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional. See the end of this chapter for the section "Policy-Based Routing Configuration Examples."

- Enabling PBR (Required)
- Enabling Local PBR (Optional)

Enabling PBR

To enable PBR, you must create a route map that specifies the match criteria and the resulting action if all of the match clauses are met. Then you must enable PBR for that route map on a particular interface. All packets arriving on the specified interface matching the match clauses are subject to PBR.

To enable PBR on an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`route-map`** *`map-tag`* `[`**`permit`** `|` **`deny`**`]` `[`*`sequence-number`*`]`	Defines a route map to control where packets are output. This command puts the router into route-map configuration mode.
Step 2	`Switch(config-route-map)#` **`match ip address`** `{`*`access-list-number`* `|` *`name`*`}` `[`*`...access-list-number`* `|` *`name`*`]`	Specifies the match criteria. Matches the source and destination IP address that is permitted by one or more standard or extended access lists.
Perform Step 3, 4, 5, or 6		

Step 3	`Switch(config-route-map)#` **`set ip next-hop`** *`ip-address`* `[`*`... ip-address`*`]`
	`Or`
		Specifies the action or actions to take on the packets that match the criteria.
		Specifies the next hop for which to route the packet (the next hop must be adjacent). This behavior is identical to a next hop specified in the normal routing table.

Step 4	`Switch(config-route-map)#` **`set interface`** *`interface-type interface-number`* `[`*`... type number`*`]`
	`Or`
		Specifies the action or actions to take on the packets that match the criteria.
		Sets output interface for the packet. This action specifies that the packet is forwarded out of the local interface. The interface must be a Layer 3 interface (no switchports), and the destination address in the packet must lie within the IP network assigned to that interface. If the destination address for the packet does not lie within that network, the packet is dropped.

	Command	Purpose
Step 5	`Switch(config-route-map)#` **`set ip default next-hop`** *`ip-address`* `[...` *`ip-address`*`]` Or	Specifies the action or actions to take on the packets that match the criteria. Sets next hop to which to route the packet if there is no explicit route for this destination. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by way of the routing table. If no match is found, the packet is forwarded to the specified next hop.
Step 6	`Switch(config-route-map)#` **`set default interface`** *`interface-type interface-number`* `[...`*`type`* `...`*`number`*`]`	Specifies the action or actions to take on the packets that match the criteria. Sets output interface for the packet if there is no explicit route for this destination. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded via the routing table. If no match is found, the packet is forwarded to the specified output interface. If the destination address for the packet does not lie within that network, the packet is dropped.
Step 7	`Switch(config-route-map)#` **`interface`** *`interface-type interface-number`*	Specifies the interface. This command puts the router into interface configuration mode.
Step 8	`Switch(config-if)#` **`ip policy route-map`** *`map-tag`*	Identifies the route map to use for PBR. One interface can only have one route map tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If there is no match, packets are routed as usual.

The **set** commands can be used in conjunction with each other. These commands are evaluated in the order shown in Step 3 in the previous task table. A usable next hop implies an interface. Once the local router finds a next hop and a usable interface, it routes the packet.

Enabling Local PBR

Packets that are generated by the router are not normally policy-routed. To enable local PBR for such packets, indicate which route map the router should use by performing this task:

Command	Purpose
`Switch(config)# ip local policy route-map map-tag`	Identifies the route map to use for local PBR.

All packets originating on the router are then be subject to local PBR.

Use the **show ip local policy** command to display the route map used for local PBR, if one exists.

Unsupported Commands

The following PBR commands in config-route-map mode are in the CLI but not supported in Cisco IOS for the Catalyst 4500 series switches. If you attempt to use these commands, an error message displays.

- **match-length**
- **set ip qos**
- **set ip tos**
- **set ip precedence**

Policy-Based Routing Configuration Examples

The following sections provide PBR configuration examples:

- Equal Access Example, page 30-5
- Differing Next Hops Example, page 30-6
- Deny ACE Example, page 30-6

For information on how to configure policy-based routing, see the section "Policy-Based Routing Configuration Task List" in this chapter.

Equal Access Example

The following example provides two sources with equal access to two different service providers. Packets arriving on interface fastethernet 3/1 from the source 1.1.1.1 are sent to the router at 6.6.6.6 if the router has no explicit route for the destination of the packet. Packets arriving from the source 2.2.2.2 are sent to the router at 7.7.7.7 if the router has no explicit route for the destination of the packet. All other packets for which the router has no explicit route to the destination are discarded.

```
Switch (config)# access-list 1 permit ip 1.1.1.1
access-list 1 permit ip 1.1.1.1
!
interface fastethernet 3/1
 ip policy route-map equal-access
```

```
!

route-map equal-access permit 10
 match ip address 1
 set ip default next-hop 6.6.6.6
route-map equal-access permit 20
 match ip address 2
 set ip default next-hop 7.7.7.7
route-map equal-access permit 30
 set default interface null0
```

Note If the packets you want to drop do not match either of the first two route-map clauses, then change **set default interface null0** to **set interface null0**.

Differing Next Hops Example

The following example illustrates how to route traffic from different sources to different places (next hops). Packets arriving from source 1.1.1.1 are sent to the next hop at 3.3.3.3; packets arriving from source 2.2.2.2 are sent to the next hop at 3.3.3.5.

```
access-list 1 permit ip 1.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
 ip policy route-map Texas
!
route-map Texas permit 10
 match ip address 1
 set ip next-hop 3.3.3.3
!
route-map Texas permit 20
 match ip address 2
 set ip next-hop 3.3.3.5
```

Deny ACE Example

The following example illustrates how to stop processing a given route map sequence, and to jump to the next sequence. Packets arriving from source 1.1.1.1 skip sequence 10 and jump to sequence 20. All other packets from subnet 1.1.1.0 follow the set statement in sequence 10.

```
access-list 1 deny ip 1.1.1.1
access-list 1 permit ip 1.1.1.0 0.0.0.255
access-list 2 permit ip 1.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
ip policy route-map Texas
!
route-map Texas permit 10
match ip address 1
set ip next-hop 3.3.3.3
!
route-map Texas permit 20
match ip address 2
set ip next-hop 3.3.3.5
```

CHAPTER 31

Configuring VRF-lite

Virtual Private Networks (VPNs) provide a secure way for customers to share bandwidth over an ISP backbone network. A VPN is a collection of sites sharing a common routing table. A customer site is connected to the service provider network by one or more interfaces, and the service provider associates each interface with a VPN routing table. A VPN routing table is called a VPN routing/forwarding (VRF) table.

With the VRF-lite feature, the Catalyst 4500 series switch supports multiple VPN routing/forwarding instances in customer edge devices. (VRF-lite is also termed multi-VRF CE, or multi-VRF Customer Edge Device). VRF-lite allows a service provider to support two or more VPNs with overlapping IP addresses using one interface.

Note The switch does not use Multiprotocol Label Switching (MPLS) to support VPNs. For information about MPLS VRF, refer to the *Cisco IOS Switching Services Configuration Guide for Release 12.3* at this location:

http://www.cisco.com/univerd/cc/td/doc/product/software/ios123/123cgcr/swit_vcg.htm

This chapter includes these topics:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cr/index.htm.

Understanding VRF-lite

VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but a Layer 3 interface cannot belong to more than one VRF at any time.

Note VRF-lite interfaces must be Layer 3 interfaces.

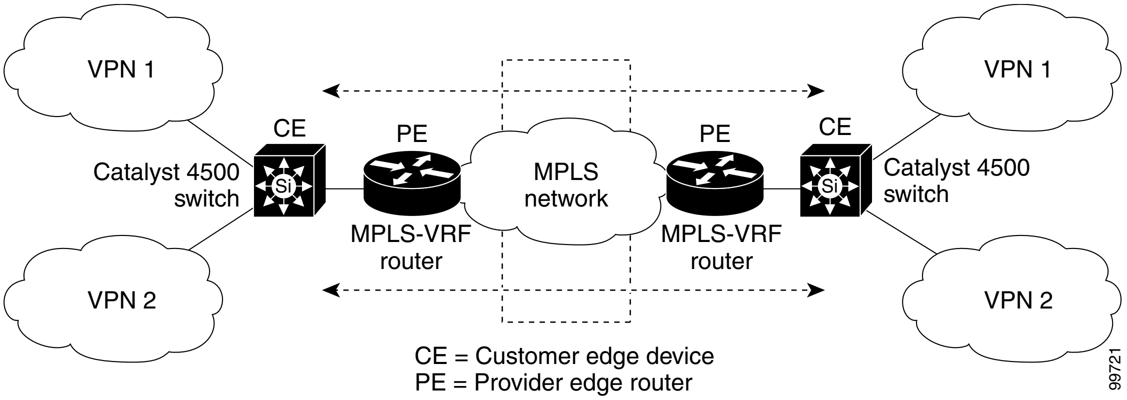
VRF-lite includes these devices:

- Customer edge (CE) devices provide customer access to the service provider network over a data link to one or more provider edge routers. The CE device advertises the site's local routes to the provider edge router and learns the remote VPN routes from it. A Catalyst 4500 series switch can be a CE.
- Provider edge (PE) routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.
- The PE is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE to maintain all of the service provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CEs, a PE router exchanges VPN routing information with other PE routers by using internal BGP (IBPG).
- Provider routers (or core routers) are any routers in the service provider network that do not attach to CE devices.

With VRF-lite, multiple customers can share one CE, and only one physical link is used between the CE and the PE. The shared CE maintains separate VRF tables for each customer and switches or routes packets for each customer based on its own routing table. VRF-lite extends limited PE functionality to a CE device, giving it the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

Figure 31-1 shows a configuration where each Catalyst 4500 series switch acts as multiple virtual CEs. Because VRF-lite is a Layer 3 feature, each interface in a VRF must be a Layer 3 interface.

Figure 31-1 Catalyst 4500 Series Switches Acting as Multiple Virtual CEs



This is the packet-forwarding process in a VRF-lite CE-enabled network as shown in Figure 31-1:

- When the CE receives a packet from a VPN, it looks up the routing table based on the input interface. When a route is found, the CE forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it performs a VRF lookup. When a route is found, the router adds a corresponding MPLS label to the packet and sends it to the MPLS network.
- When an egress PE receives a packet from the network, it strips the label and uses the label to identify the correct VPN routing table. Then the egress PE performs the normal route lookup. When a route is found, it forwards the packet to the correct adjacency.
- When a CE receives a packet from an egress PE, it uses the input interface to look up the correct VPN routing table. If a route is found, the CE forwards the packet within the VPN.

To configure VRF, create a VRF table and specify the Layer 3 interface associated with the VRF. Then configure the routing protocols in the VPN and between the CE and the PE. BGP is the preferred routing protocol used to distribute VPN routing information across the provider's backbone. The VRF-lite network has three major components:

- VPN route target communities—Lists of all other members of a VPN community. You need to configure VPN route targets for each VPN community member.
- Multiprotocol BGP peering of VPN community PE routers—Propagates VRF reachability information to all members of a VPN community. You need to configure BGP peering in all PE routers within a VPN community.
- VPN forwarding—Transports all traffic between all VPN community members across a VPN service-provider network.

Default VRF-lite Configuration

Table 31-1 shows the default VRF configuration.

Table 31-1 Default VRF Configuration

Feature	Default Setting
VRF	Disabled. No VRFs are defined.
Maps	No import maps, export maps, or route maps are defined.
VRF maximum routes	None.
Forwarding table	The default for an interface is the global routing table.

VRF-lite Configuration Guidelines

Consider these points when configuring VRF in your network:

- A switch with VRF-lite is shared by multiple customers, and all customers have their own routing tables.
- Because customers use different VRF tables, the same IP addresses can be reused. Overlapped IP addresses are allowed in different VPNs.
- VRF-lite lets multiple customers share the same physical link between the PE and the CE. Trunk ports with multiple VLANs separate packets among customers. All customers have their own VLANs.
- VRF-lite does not support all MPLS-VRF functionality: label exchange, LDP adjacency, or labeled packets.
- For the PE router, there is no difference between using VRF-lite or using multiple CEs. In Figure 31-1, multiple virtual Layer 3 interfaces are connected to the VRF-lite device.
- The Catalyst 4500 series switch supports configuring VRF by using physical ports, VLAN SVIs, or a combination of both. The SVIs can be connected through an access port or a trunk port.
- A customer can use multiple VLANs as long as they do not overlap with those of other customers. A customer’s VLANs are mapped to a specific routing table ID that is used to identify the appropriate routing tables stored on the switch.
- The Layer 3 TCAM resource is shared between all VRFs. To ensure that any one VRF has sufficient CAM space, use the **maximum routes** command.
- A Catalyst 4500 series switch using VRF can support one global network and up to 64 VRFs. The total number of routes supported is limited by the size of the TCAM.
- Most routing protocols (BGP, OSPF, EIGRP, RIP and static routing) can be used between the CE and the PE. However, we recommend using external BGP (EBGP) for these reasons:
  - BGP does not require multiple algorithms to communicate with multiple CEs.
  - BGP is designed for passing routing information between systems run by different administrations.
  - BGP makes it easy to pass attributes of the routes to the CE.
- VRF-lite does not support IGRP and ISIS.
- VRF-lite does not affect the packet switching rate.
- Multicast cannot be configured on the same Layer 3 interface at the same time.
- The **capability vrf-lite** subcommand under **router ospf** should be used when configuring OSPF as the routing protocol between the PE and the CE.

Configuring VRFs

To configure one or more VRFs, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`ip routing`**	Enables IP routing.
Step 3	`Switch(config)#` **`ip vrf`** *`vrf-name`*	Names the VRF, and enter VRF configuration mode.
Step 4	`Switch(config-vrf)#` **`rd`** *`route-distinguisher`*	Creates a VRF table by specifying a route distinguisher. Enter either an AS number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 5	`Switch(config-vrf)#` **`route-target`** {**`export`** | **`import`** | **`both`**} *`route-target-ext-community`*	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). **Note** This command is effective only if BGP is running.
Step 6	`Switch(config-vrf)#` **`import map`** *`route-map`*	(Optional) Associates a route map with the VRF.
Step 7	`Switch(config-vrf)#` **`interface`** *`interface-id`*	Enters interface configuration mode and specify the Layer 3 interface to be associated with the VRF. The interface can be a routed port or SVI.
Step 8	`Switch(config-if)#` **`ip vrf forwarding`** *`vrf-name`*	Associates the VRF with the Layer 3 interface.
Step 9	`Switch(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 10	`Switch#` **`show ip vrf`** [**`brief`** | **`detail`** | **`interfaces`**] [*`vrf-name`*]	Verifies the configuration. Display information about the configured VRFs.
Step 11	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Note For complete syntax and usage information for the commands, refer to the switch command reference for this release and the *Cisco IOS Switching Services Command Reference for Release 12.2*.

Use the **no ip vrf** *vrf-name* global configuration command to delete a VRF and to remove all interfaces from it. Use the **no ip vrf forwarding** interface configuration command to remove an interface from the VRF.

Configuring a VPN Routing Session

Routing within the VPN can be configured with any supported routing protocol (RIP, OSPF, or BGP) or with static routing. The configuration shown here is for OSPF, but the process is the same for other protocols.

To configure OSPF in the VPN, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **router ospf** *process-id* **vrf** *vrf-name*	Enables OSPF routing, specifies a VPN forwarding table, and enters router configuration mode.
Step 3	Switch(config-router)# **log-adjacency-changes**	(Optional) Logs changes in the adjacency state. This is the default state.
Step 4	Switch(config-router)# **redistribute bgp** *autonomous-system-number* subnets	Sets the switch to redistribute information from the BGP network to the OSPF network.
Step 5	Switch(config-router)# **network** *network-number* **area** *area-id*	Defines a network address and mask on which OSPF runs and the area ID for that network address.
Step 6	Switch(config-router)# **end**	Returns to privileged EXEC mode.
Step 7	Switch# **show ip ospf** *process-id*	Verifies the configuration of the OSPF network.
Step 8	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no router ospf** *process-id* **vrf** *vrf-name* global configuration command to disassociate the VPN forwarding table from the OSPF routing process.

Configuring BGP PE to CE Routing Sessions

To configure a BGP PE to CE routing session, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **router bgp** *autonomous-system-number*	Configures the BGP routing process with the AS number passed to other BGP routers and enters router configuration mode.
Step 3	Switch(config-router)# **network** *network-number* **mask** *network-mask*	Specifies a network and mask to announce using BGP.
Step 4	Switch(config-router)# **redistribute ospf** *process-id* match internal	Sets the switch to redistribute OSPF internal routes.
Step 5	Switch(config-router)# **network** *network-number* **area** *area-id*	Defines a network address and mask on which OSPF runs and the area ID for that network address.
Step 6	Switch(config-router-af)# **address-family ipv4 vrf** *vrf-name*	Defines BGP parameters for PE to CE routing sessions and enters VRF address-family mode.
Step 7	Switch(config-router-af)# **neighbor** *address* **remote-as** *as-number*	Defines a BGP session between PE and CE routers.
Step 8	Switch(config-router-af)# **neighbor** *address* **activate**	Activates the advertisement of the IPv4 address family.
Step 9	Switch(config-router-af)# **end**	Returns to privileged EXEC mode.

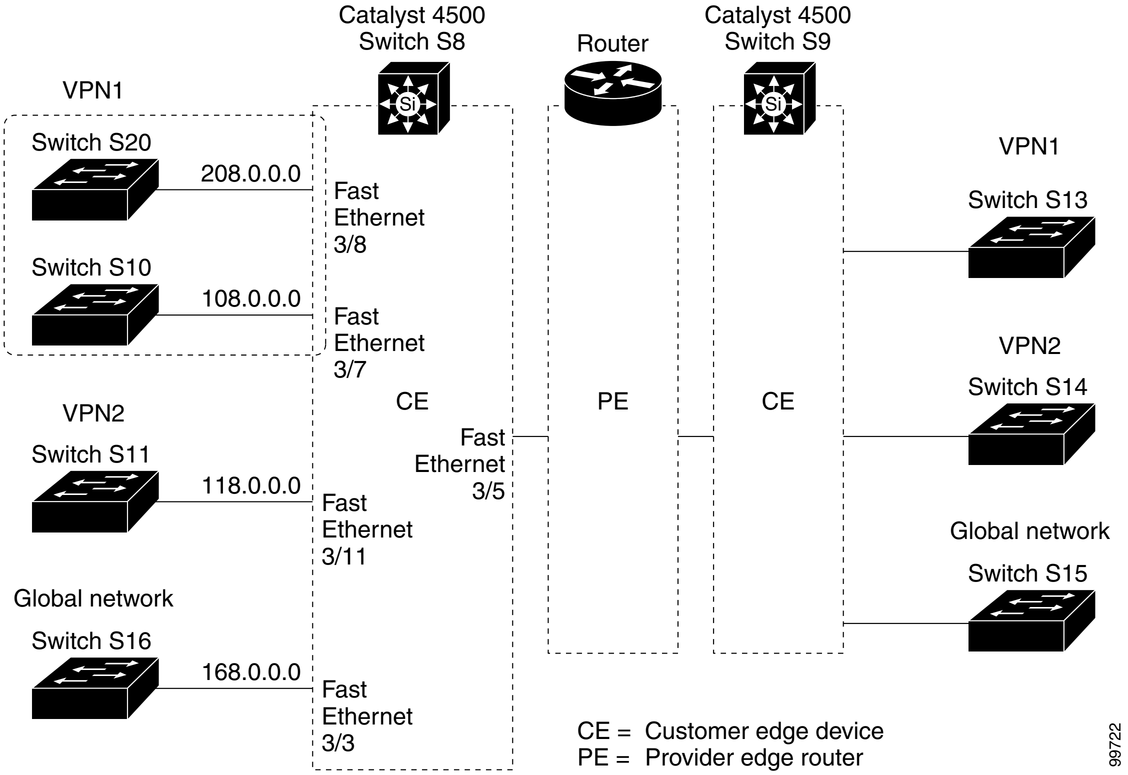
	Command	Purpose
Step 10	`Switch# show ip bgp [ipv4] [neighbors]`	Verifies BGP configuration.
Step 11	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Use the **no router bgp** *autonomous-system-number* global configuration command to delete the BGP routing process. Use the command with keywords to delete routing characteristics.

VRF-lite Configuration Example

Figure 31-2 is a simplified example of the physical connections in a network similar to that in Figure 31-1. OSPF is the protocol used in VPN1, VPN2, and the global network. BGP is used in the CE to PE connections. The example commands show how to configure the CE switch S8 and include the VRF configuration for switches S20 and S11 and the PE router commands related to traffic with switch S8. Commands for configuring the other switches are not included but would be similar.

Figure 31-2 VRF-lite Configuration Example



Configuring Switch S8

On switch S8, enable routing and configure VRF.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# ip vrf v11
Switch(config-vrf)# rd 800:1
Switch(config-vrf)# route-target export 800:1
Switch(config-vrf)# route-target import 800:1
Switch(config-vrf)# exit
Switch(config)# ip vrf v12
Switch(config-vrf)# rd 800:2
Switch(config-vrf)# route-target export 800:2
Switch(config-vrf)# route-target import 800:2
Switch(config-vrf)# exit
```

Configure the loopback and physical interfaces on switch S8. Fast Ethernet interface 3/5 is a trunk connection to the PE. Interfaces 3/7 and 3/11 connect to VPNs:

```
Switch(config)# interface loopback1
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 8.8.1.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface loopback2
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 8.8.2.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/5
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/8
Switch(config-if)# switchport access vlan 208
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/11
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit
```

Configure the VLANs used on switch S8. VLAN 10 is used by VRF 11 between the CE and the PE. VLAN 20 is used by VRF 12 between the CE and the PE. VLANs 118 and 208 are used for VRF for the VPNs that include switch S11 and switch S20, respectively:

```
Switch(config)# interface Vlan10
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 38.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan20
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 83.0.0.8 255.255.255.0
Switch(config-if)# exit
Switch(config)# interface Vlan118
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 118.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan208
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 208.0.0.8 255.255.255.0
Switch(config-if)# exit
```

Configure OSPF routing in VPN1 and VPN2:

```
Switch(config)# router ospf 1 vrf vl1
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
Switch(config)# router ospf 2 vrf vl2
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
```

Configure BGP for CE to PE routing:

```
Switch(config)# router bgp 800
Switch(config-router)# address-family ipv4 vrf vl2
Switch(config-router-af)# redistribute ospf 2 match internal
Switch(config-router-af)# neighbor 83.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 83.0.0.3 activate
Switch(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Switch(config-router-af)# exit

Switch(config-router)# address-family ipv4 vrf vl1
Switch(config-router-af)# redistribute ospf 1 match internal
Switch(config-router-af)# neighbor 38.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 38.0.0.3 activate
Switch(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Switch(config-router-af)# end
```

Configuring Switch S20

Configure S20 to connect to CE:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Fast Ethernet 0/7
Switch(config-if)# no switchport
Switch(config-if)# ip address 208.0.0.20 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring Switch S11

Configure S11 to connect to CE:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Gigabit Ethernet 0/3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# ip address 118.0.0.11 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring the PE Switch S3

On switch S3 (the router), these commands configure only the connections to switch S8:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit

Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Loopback2
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf v1
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

Displaying VRF-lite Status

To display information about VRF-lite configuration and status, perform one of the following tasks:

Command	Purpose
Switch# **show ip protocols vrf** *vrf-name*	Displays routing protocol information associated with a VRF.
Switch# **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**] [**supernets-only**]	Displays IP routing table information associated with a VRF.
Switch# **show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]	Displays information about the defined VRF instances.

Note For more information about the information in the displays, refer to the *Cisco IOS Switching Services Command Reference for Release 12.2* at:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fswtch_r

CHAPTER 32

Configuring Quality of Service

This chapter describes how to configure quality of service (QoS) by using automatic QoS (auto-QoS) commands or by using standard QoS commands on a Catalyst 4500 series switch. It describes how to specify QoS configuration on different types of interfaces (access, Layer 2 trunk, Layer 3 routed, Etherchannel) as well as VLANs. It also describes how to specify different QoS configurations on different VLANs on a given interface (per-port per-VLAN QoS). This chapter describes QoS support on Supervisor Engines II-Plus to V-10GE and on Supervisor Engine 6-E.

This chapter consists of these sections:

- Overview of QoS on Catalyst 4500 Series Switch, page 32-1
- Configuring Auto-QoS on Supervisor Engines II-Plus, II+10GE, VI, V, V-10GE, 4924, 4948, and 4948-10GE, page 32-17
- Configuring QoS on Supervisor Engines II-Plus, II+10GE, VI, V, V-10GE, 4924, 4948, and 4948-10GE, page 32-23
- Configuring Auto-QoS on Supervisor Engine 6-E, page 32-65
- Configuring QoS on Supervisor Engine 6-E, page 32-67

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of QoS on Catalyst 4500 Series Switch

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 can make network performance more predictable and bandwidth utilization more effective.

This section contains the following subsections:

- Prioritization, page 32-2
- QoS Terminology, page 32-3

Prioritization

QoS implementation is based on the DiffServ architecture. This architecture specifies that each packet is classified upon entry into the network. The classification is carried in the IP packet header, using 6 bits from the deprecated IP type of service (TOS) field to carry the classification (*class*) information. Classification can also be carried in the Layer 2 frame. These special bits in the Layer 2 frame or a Layer 3 packet are described here and shown in Figure 32-1:

- Prioritization values in Layer 2 frames:

  Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p class of service (CoS) value in the three least-significant bits. On interfaces configured as Layer 2 ISL trunks, all traffic is in ISL frames.

  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. On interfaces configured as Layer 2 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.

  Other frame types cannot carry Layer 2 CoS values.

  Layer 2 CoS values range from 0 for low priority to 7 for high priority.

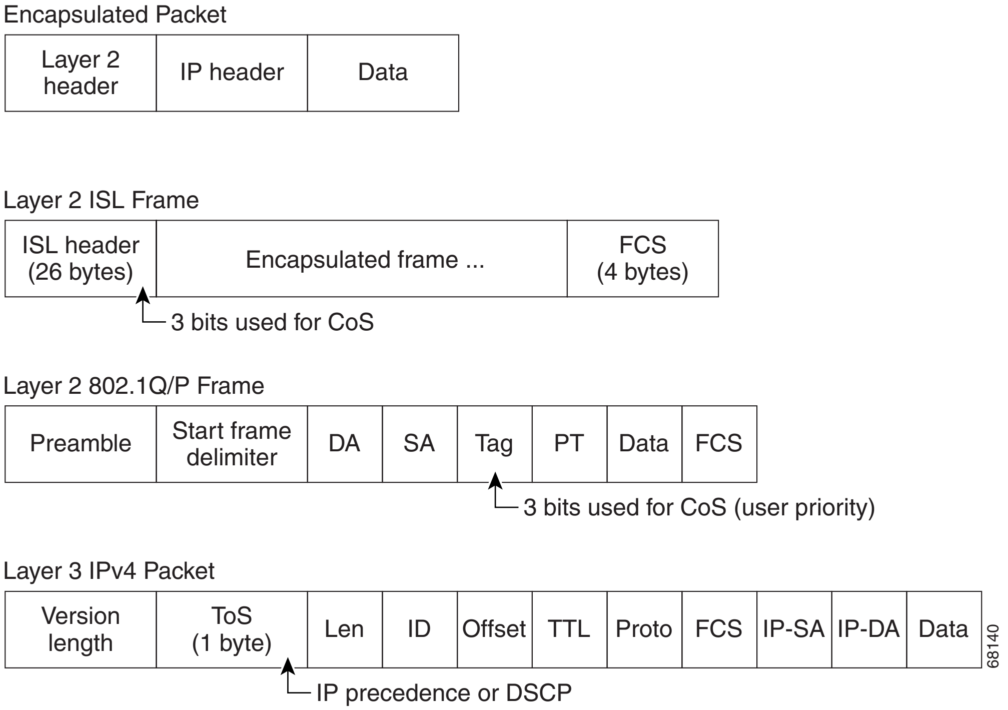
- Prioritization bits in Layer 3 packets:

  Layer 3 IP packets can carry either an IP precedence value or a Differentiated Services Code Point (DSCP) value. QoS supports the use of either value because DSCP values are backward-compatible with IP precedence values.

  IP precedence values range from 0 to 7.

  DSCP values range from 0 to 63.

Figure 32-1 QoS Classification Layers in Frames and Packets



All switches and routers across the Internet rely on the class information to provide the same forwarding treatment to packets with the same class information and different treatment to packets with different class information. The class information in the packet can be assigned by end hosts or by switches or routers along the way, based on a configured policy, detailed examination of the packet, or both. Detailed examination of the packet is expected to happen closer to the edge of the network so that the core switches and routers are not overloaded.

Switches and routers along the path can use the class information to limit the amount of resources allocated per traffic class. The behavior of an individual device when handling traffic in the DiffServ architecture is called per-hop behavior. If all devices along a path provide a consistent per-hop behavior, you can construct an end-to-end QoS solution.

Implementing QoS in your network can be a simple or complex task and depends on the QoS features offered by your internetworking devices, the traffic types and patterns in your network, and the granularity of control you need over incoming and outgoing traffic.

QoS Terminology

The following terms are used when discussing QoS features:

- *Packets* carry traffic at Layer 3.
- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *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 interfaces configured as Layer 2 ISL trunks, all traffic is in ISL frames. On interfaces 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 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 Maps" section on page 32-57.

Note Layer 3 IP packets can carry either an IP precedence value or a DSCP value. QoS supports the use of either value, since DSCP values are backwards compatible with IP precedence values. See Table 32-1.

Table 32-1 IP Precedence and DSCP Values

3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP		3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP
	8	7	6	5	4	3				8	7	6	5	4	3	
0	0	0	0	0	0	0	0		4	1	0	0	0	0	0	32
	0	0	0	0	0	1	1			1	0	0	0	0	1	33
	0	0	0	0	1	0	2			1	0	0	0	1	0	34
	0	0	0	0	1	1	3			1	0	0	0	1	1	35
	0	0	0	1	0	0	4			1	0	0	1	0	0	36
	0	0	0	1	0	1	5			1	0	0	1	0	1	37
	0	0	0	1	1	0	6			1	0	0	1	1	0	38
	0	0	0	1	1	1	7			1	0	0	1	1	1	39
1	0	0	1	0	0	0	8		5	1	0	1	0	0	0	40
	0	0	1	0	0	1	9			1	0	1	0	0	1	41
	0	0	1	0	1	0	10			1	0	1	0	1	0	42
	0	0	1	0	1	1	11			1	0	1	0	1	1	43
	0	0	1	1	0	0	12			1	0	1	1	0	0	44
	0	0	1	1	0	1	13			1	0	1	1	0	1	45
	0	0	1	1	1	0	14			1	0	1	1	1	0	46
	0	0	1	1	1	1	15			1	0	1	1	1	1	47

Table 32-1 IP Precedence and DSCP Values (continued)

3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP		3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP
	8	7	6	5	4	3				8	7	6	5	4	3	
2	0	1	0	0	0	0	16		6	1	1	0	0	0	0	48
	0	1	0	0	0	1	17			1	1	0	0	0	1	49
	0	1	0	0	1	0	18			1	1	0	0	1	0	50
	0	1	0	0	1	1	19			1	1	0	0	1	1	51
	0	1	0	1	0	0	20			1	1	0	1	0	0	52
	0	1	0	1	0	1	21			1	1	0	1	0	1	53
	0	1	0	1	1	0	22			1	1	0	1	1	0	54
	0	1	0	1	1	1	23			1	1	0	1	1	1	55
3	0	1	1	0	0	0	24		7	1	1	1	0	0	0	56
	0	1	1	0	0	1	25			1	1	1	0	0	1	57
	0	1	1	0	1	0	26			1	1	1	0	1	0	58
	0	1	1	0	1	1	27			1	1	1	0	1	1	59
	0	1	1	1	0	0	28			1	1	1	1	0	0	60
	0	1	1	1	0	1	29			1	1	1	1	0	1	61
	0	1	1	1	1	0	30			1	1	1	1	1	0	62
	0	1	1	1	1	1	31			1	1	1	1	1	1	63

[1] MSb = most significant bit

- *Classification* is the selection of traffic to be marked.
- *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. QoS assigns frames to a queue based on internal DSCP values as shown in Internal DSCP Values, page 32-13.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.

Basic QoS Model

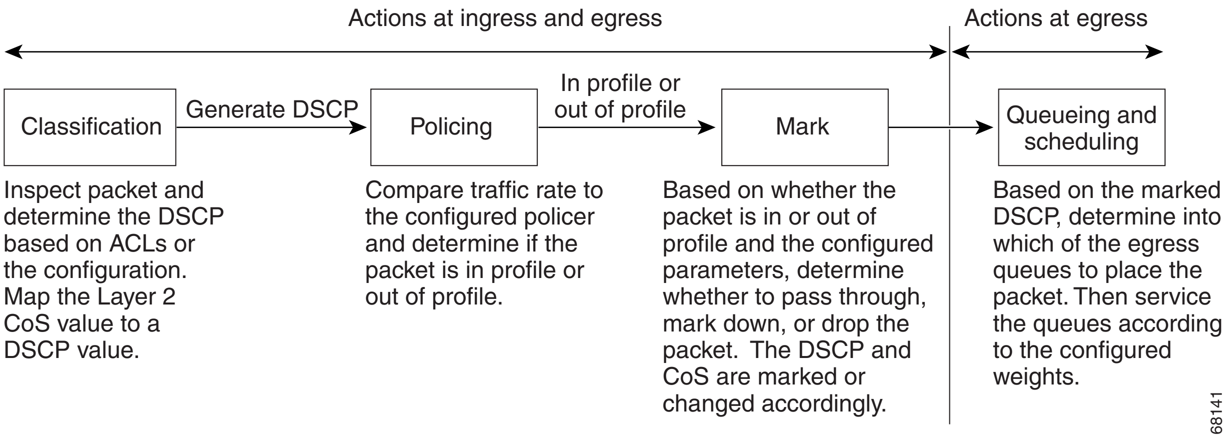
Figure 32-2 shows the basic QoS model (also referred to as Switch QoS model; it is not MQC compliant). Actions at the ingress and egress interfaces include classifying traffic, policing, and marking:

- Classifying distinguishes one kind of traffic from another. The process generates an internal DSCP for a packet, which identifies all the future QoS actions to be performed on this packet. For more information, see the “Classification” section on page 32-6.
- Policing determines whether a packet is in or out of profile by comparing the traffic rate to the configured policer, which limits the bandwidth consumed by a flow of traffic. The result of this determination is passed to the marker. For more information, see the “Policing and Marking” section on page 32-10.
- Marking evaluates the policer configuration information regarding the action to be taken when a packet is out of profile and decides what to do with the packet (pass through a packet without modification, mark down the DSCP value in the packet, or drop the packet). For more information, see the “Policing and Marking” section on page 32-10.

Actions at the egress interface include queueing and scheduling:

- Queueing evaluates the internal DSCP and determines which of the four egress queues in which to place the packet.
- Scheduling services the four egress (transmit) queues based on the sharing and shaping configuration of the egress (transmit) port. Sharing and shaping configurations are described in the "Queueing and Scheduling" section on page 32-14.

Figure 32-2 Basic QoS Model


Actions at ingress and egress
Actions at egress
Classification
Generate DSCP
Policing
In profile or out of profile
Mark
Queueing and scheduling
Inspect packet and determine the DSCP based on ACLs or the configuration. Map the Layer 2 CoS value to a DSCP value.
Compare traffic rate to the configured policer and determine if the packet is in profile or out of profile.
Based on whether the packet is in or out of profile and the configured parameters, determine whether to pass through, mark down, or drop the packet. The DSCP and CoS are marked or changed accordingly.
Based on the marked DSCP, determine into which of the egress queues to place the packet. Then service the queues according to the configured weights.
68141

Classification

Classification is the process of distinguishing one kind of traffic from another by examining the fields in the packet. Classification is enabled only if QoS is globally enabled on the switch. By default, QoS is globally disabled, so no classification occurs.

You specify which fields in the frame or packet that you want to use to classify incoming traffic.

Classification options are shown in Figure 32-3.

For non-IP traffic, you have the following classification options:

- Use the port default. If the packet is a non-IP packet, assign the default port DSCP value to the incoming packet.
- Trust the CoS value in the incoming frame (configure the port to trust CoS). Then use the configurable CoS-to-DSCP map to generate the internal DSCP value. Layer 2 ISL frame headers carry the CoS value in the three least-significant bits of the 1-byte User field. Layer 2 802.1Q frame headers carry the CoS value in the three most-significant bits of the Tag Control Information field. CoS values range from 0 for low priority to 7 for high priority. If the frame does not contain a CoS value, assign the default port CoS to the incoming frame.

  The trust DSCP configuration is meaningless for non-IP traffic. If you configure a port with trust DSCP and non-IP traffic is received, the switch assigns the default port DSCP.

For IP traffic, you have the following classification options:

- Trust the IP DSCP in the incoming packet (configure the port to trust DSCP), and assign the same DSCP to the packet for internal use. The IETF defines the six most-significant bits of the 1-byte Type of Service (ToS) field as the DSCP. The priority represented by a particular DSCP value is configurable. DSCP values range from 0 to 63.

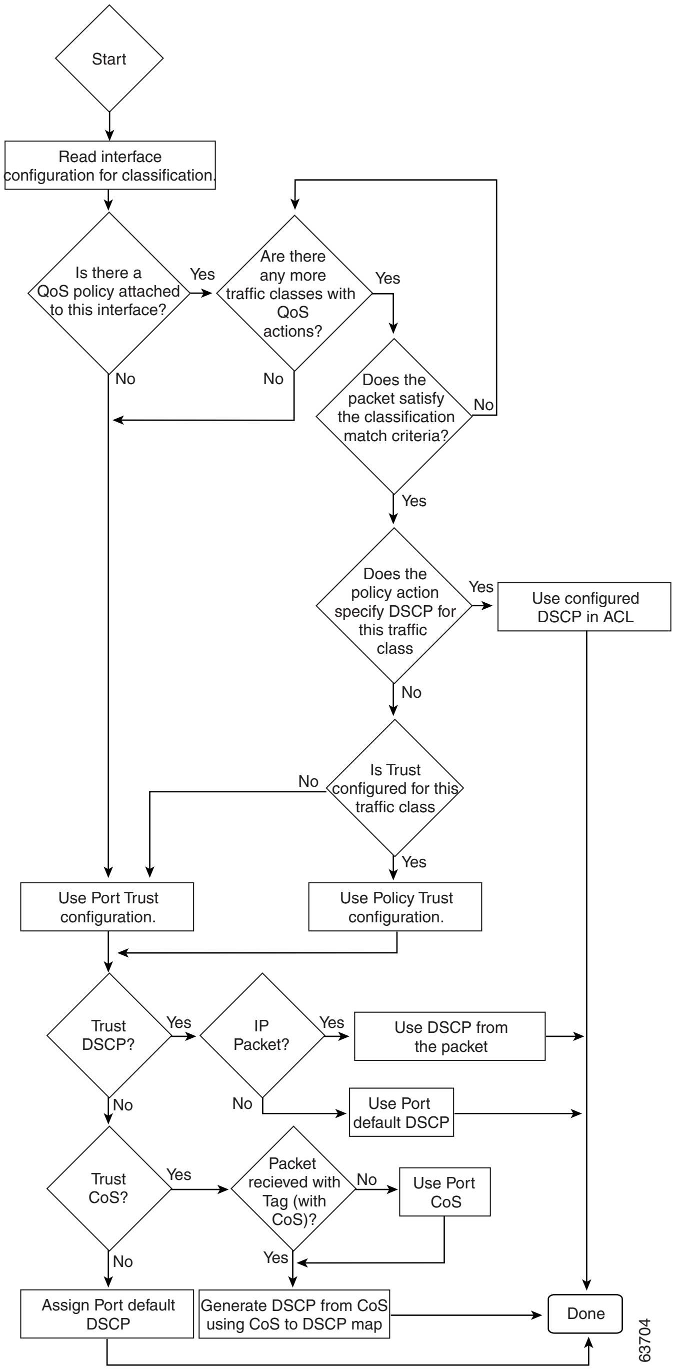
- Trust the CoS value (if present) in the incoming packet, and generate the DSCP by using the CoS-to-DSCP map.
- Perform the classification based on a configured IP standard or extended ACL, which examines various fields in the IP header. If no ACL is configured, the packet is assigned the default DSCP based on the trust state of the ingress port; otherwise, the policy map specifies the DSCP to assign to the incoming frame.

Note It is not possible to classify traffic based on the markings performed by an input QoS policy. In the Catalyst 4500 platform, the input and output QoS lookup happen in parallel, and therefore, input marked DSCP value cannot be used to classify traffic in the output QoS policy.

Note It is not possible to classify traffic based on *internal DSCP*. The *internal DSCP* is purely an internal classification mechanism used for all packets to determine transmit queue and transmit CoS values only.

For information on the maps described in this section, see the “Mapping Tables” section on page 32-14. For configuration information on port trust states, see the “Configuring the Trust State of Interfaces” section on page 32-52.

Figure 32-3 Classification Flowchart


Start
Read interface configuration for classification.
Is there a QoS policy attached to this interface?
Yes
No
Are there any more traffic classes with QoS actions?
Yes
No
Does the packet satisfy the classification match criteria?
No
Yes
Does the policy action specify DSCP for this traffic class
Yes
No
Use configured DSCP in ACL
Is Trust configured for this traffic class
No
Yes
Use Port Trust configuration.
Use Policy Trust configuration.
Trust DSCP?
Yes
No
IP Packet?
Yes
No
Use DSCP from the packet
Use Port default DSCP
Trust CoS?
Yes
No
Packet recieved with Tag (with CoS)?
No
Yes
Use Port CoS
Assign Port default DSCP
Generate DSCP from CoS using CoS to DSCP map
Done
63704

Classification Based on QoS ACLs

A packet can be classified for QoS using multiple match criteria, and the classification can specify whether the packet should match all of the specified match criteria or at least one of the match criteria. To define a QoS classifier, you can provide the match criteria using the *match* statements in a class map. In the 'match' statements, you can specify the fields in the packet to match on, or you can use IP standard or IP extended ACLs. For more information, see the "Classification Based on Class Maps and Policy Maps" section on page 32-9.

If the class map is configured to match all the match criteria, then a packet must satisfy all the match statements in the class map before the QoS action is taken. The QoS action for the packet is not taken if the packet does not match even one match criterion in the class map.

If the class map is configured to match at least one match criterion, then a packet must satisfy at least one of the match statements in the class map before the QoS action is taken. The QoS action for the packet is not taken if the packet does not match any match criteria in the class map.

Note When you use the IP standard and IP extended ACLs, the permit and deny ACEs in the ACL have a slightly different meaning in the QoS context.

- If a packet encounters (and satisfies) an ACE with a "permit," then the packet "matches" the match criterion in the QoS classification.
- If a packet encounters (and satisfies) an ACE with a "deny," then the packet "does not match" the match criterion in the QoS classification.
- If no match with a permit action is encountered and all the ACEs have been examined, then the packet "does not match" the criterion in the QoS classification.

Note When creating an access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.

After a traffic class has been defined with the class map, you can create a policy that defines the QoS actions for a traffic class. A policy might contain multiple classes with actions specified for each one of them. A policy might include commands to classify the class as a particular aggregate (for example, assign a DSCP) or rate limit the class. This policy is then attached to a particular port on which it becomes effective.

You implement IP ACLs to classify IP traffic by using the **access-list** global configuration command. For configuration information, see the "Configuring a QoS Policy" section on page 32-32.

Classification Based on Class Maps and Policy Maps

A class map is a mechanism that you use to isolate and name a specific traffic flow (or class) from all other traffic. The class map defines the criterion used to match against a specific traffic flow to further classify it; the criteria can include matching the access group defined by the ACL or matching a specific list of DSCP, IP precedence, or L2 CoS values. If you have more than one type of traffic that you want to classify, you can create another class map and use a different name. After a packet is matched against the class-map criteria, you can specify the QoS actions via a policy map.

A policy map specifies the QoS actions for the traffic classes. Actions can include trusting the CoS or DSCP values in the traffic class; setting a specific DSCP or IP precedence value in the traffic class; or specifying the traffic bandwidth limitations and the action to take when the traffic is out of profile. Before a policy map can be effective, you must attach it to an interface.

You create a class map by using the **class-map** global configuration command. When you enter the **class-map** command, the switch enters the class-map configuration mode. In this mode, you define the match criteria for the traffic by using the **match** class-map configuration command.

You create and name a policy map by using the **policy-map** global configuration command. When you enter this command, the switch enters the policy-map configuration mode. In this mode, you specify the actions to take on a specific traffic class by using the **trust** or **set** policy-map configuration and policy-map class configuration commands. To make the policy map effective, you attach it to an interface by using the **service-policy** interface configuration command.

The policy map can also contain commands that define the policer, (the bandwidth limitations of the traffic) and the action to take if the limits are exceeded. For more information, see the "Policing and Marking" section on page 32-10.

A policy map also has these characteristics:

- A policy map can contain up to 255 class statements.
- You can have different classes within a policy map.
- A policy-map trust state supersedes an interface trust state.

For configuration information, see the "Configuring a QoS Policy" section on page 32-32.

Policing and Marking

After a packet is classified and has an internal DSCP value assigned to it, the policing and marking process can begin as shown in Figure 32-4.

Policing involves creating a policer that specifies the bandwidth limits for the traffic. Packets that exceed the limits are *out of profile* or *nonconforming*. Each policer specifies the action to take for packets that are in or out of profile. These actions, carried out by the marker, include passing through the packet without modification, dropping the packet, or marking down the packet with a new DSCP value that is obtained from the configurable policed-DSCP map. For information on the policed-DSCP map, see the "Mapping Tables" section on page 32-14.

You can create these types of policers:

- Individual

  QoS applies the bandwidth limits specified in the policer separately to each matched traffic class for each port/VLAN to which the policy map is attached to. You configure this type of policer within a policy map by using the **police** command under policy-map class configuration mode.

- Aggregate

  QoS applies the bandwidth limits specified in an aggregate policer cumulatively to all matched traffic flows. You configure this type of policer by specifying the aggregate policer name within a policy map by using the **police aggregate** policy-map configuration command. You specify the bandwidth limits of the policer by using the **qos aggregate-policer** global configuration command. In this way, the aggregate policer is shared by multiple classes of traffic within a policy map.

- Flow or Microflow

  With flow-based policing, all the identified flows are policed to the specified rate individually. Because the flows are dynamic, key distinguishing fields must be configured in class maps. Two flow-matching options are provided: *source ip based* (each flow with unique source IP address is treated as a new flow) and *destination ip based* (each flow with unique destination IP address is treated as new flow). For information on flow-based policer configuration, see "Configuring User Based Rate Limiting" on page 42.

When configuring policing and policers, keep these items in mind:

- For IP packets, only the length of the IP payload (the total length field in the IP header) is used by the policer for policing computation. The Layer 2 header and trailer length are not taken into account. For example, for a 64-byte Ethernet II IP packet, only 46 bytes are taken into account for policing (64 bytes - 14 byte Ethernet Header - 4 bytes Ethernet CRC).

  For non-IP packets, the Layer 2 length as specified in the Layer 2 Header is used by the policer for policing computation. To specify additional Layer 2 encapsulation length when policing IP packets, use the **qos account layer2 encapsulation** command.
- By default, no policers are configured.
- Only the average rate and committed burst parameters are configurable.
- Policing for individual and aggregate policers can occur in ingress and egress interfaces.
  - With the Supervisor Engine V-10GE (WS-X4516-10GE), 8192 policers are supported on ingress and on egress.
  - With all other supervisor engines, 1024 policers are supported on ingress and on egress.

Note Four policers in ingress and egress direction are reserved.

- Policers can be of individual or aggregate type. On the Supervisor Engine V-10GE, flow based policers are supported.
- Policing for flow policers can occur on ingress Layer 3 interfaces only.
  - 512 unique flow policers can be configured on the Supervisor Engine V-10GE.

Note Because one flow policer is reserved by software, 511 unique flow policers can be defined.

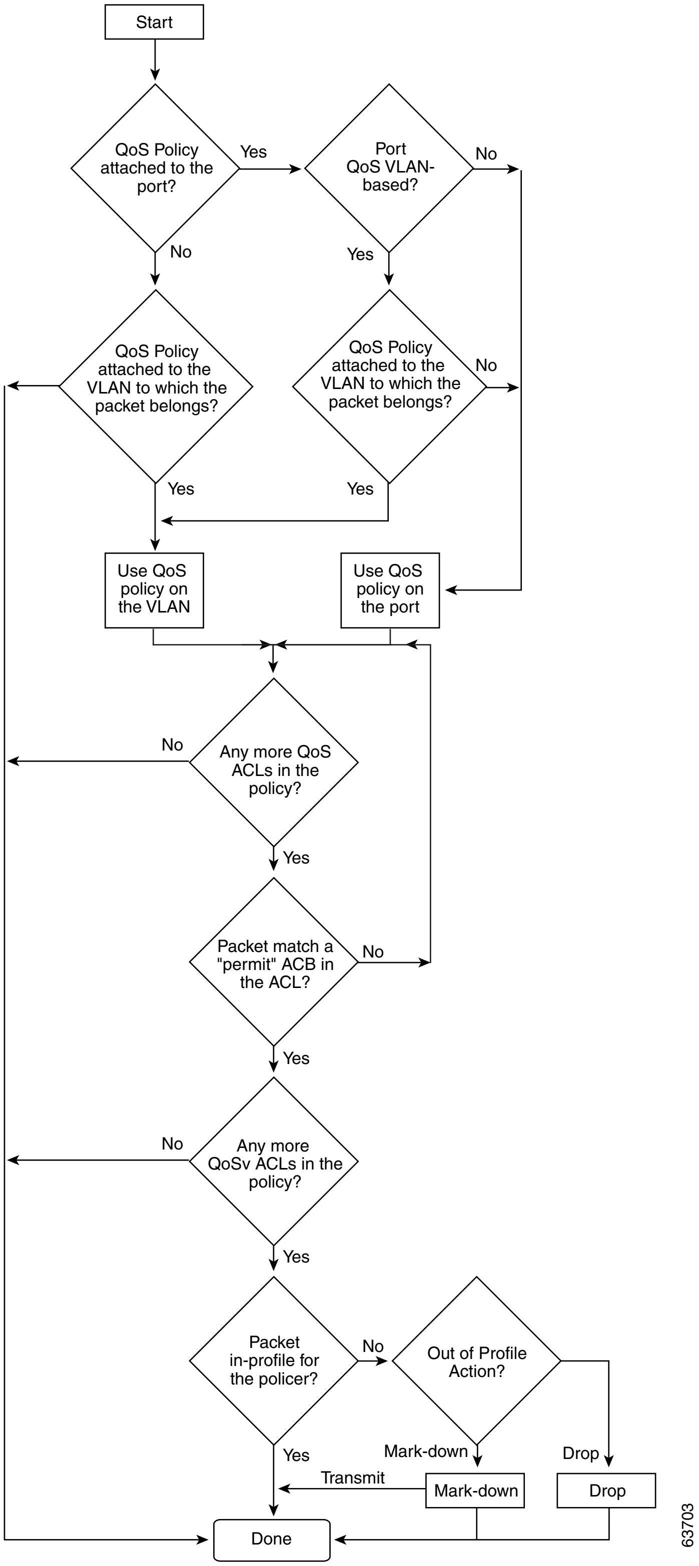
  - Greater than 100,000 flows can be microflow policed.

Note Microflow currently supports two flow matching options (source IP address based and destination IP address based). When microflow policing is used together with Netflow Statistics Collection, full flow statistics for the flows matching the source IP address or destination IP address are not available. For information on configuring Netflow Statistics, refer to “Enabling NetFlow Statistics Collection” section on page 46-7.

- On an interface configured for QoS, all traffic received or sent through the interface is classified, policed, and marked according to the policy map attached to the interface. However, if the interface is configured to use VLAN-based QoS (using the **qos vlan-based** command), the traffic received or sent through the interface is classified, policed, and marked according to the policy map attached to the VLAN (configured on the VLAN interface) to which the packet belongs. If there is no policy map attached to the VLAN to which the packet belongs, the policy map attached to the interface is used.

After you configure the policy map and policing actions, attach the policy to an ingress or egress interface by using the **service-policy** interface configuration command. For configuration information, see the “Configuring a QoS Policy” section on page 32-32 and the “Creating Named Aggregate Policers” section on page 32-31.

Figure 32-4   Policing and Marking Flowchart


Start
QoS Policy attached to the port?
Yes
Port QoS VLAN-based?
No
No
Yes
QoS Policy attached to the VLAN to which the packet belongs?
QoS Policy attached to the VLAN to which the packet belongs?
No
Yes
Yes
Use QoS policy on the VLAN
Use QoS policy on the port
No
Any more QoS ACLs in the policy?
Yes
Packet match a "permit" ACB in the ACL?
No
Yes
No
Any more QoSv ACLs in the policy?
Yes
Packet in-profile for the policer?
No
Out of Profile Action?
Yes
Mark-down
Drop
Transmit
Mark-down
Drop
Done
63703

Internal DSCP Values

The following sections describe the internal DSCP values:

- Internal DSCP Sources, page 32-13
- Egress ToS and CoS Sources, page 32-13

Internal DSCP Sources

During processing, QoS represents the priority of all traffic (including non-IP traffic) with an internal DSCP value. QoS derives the internal DSCP value from the following:

- For trust-CoS traffic, from received or ingress interface Layer 2 CoS values
- For trust-DSCP traffic, from received or ingress interface DSCP values
- For untrusted traffic, from ingress interface DSCP value

The trust state of traffic is the trust state of the ingress interface unless set otherwise by a policy action for this traffic class.

QoS uses configurable mapping tables to derive the internal 6-bit DSCP value from CoS, which are 3-bit values (see the“Configuring DSCP Maps” section on page 32-57).

Egress ToS and CoS Sources

For egress IP traffic, QoS creates a ToS byte from the internal DSCP value and sends it to the egress interface 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 ToS value can mimic an IP precedence value (see Table 32-1 on page 32-4).

For all egress traffic, QoS uses a configurable mapping table to derive a CoS value from the internal ToS value associated with traffic (see the “Configuring the DSCP-to-CoS Map” section on page 32-59). QoS sends the CoS value to be written into ISL and 802.1Q frames.

For traffic received on an ingress interface configured to *trust CoS* using the **qos trust cos** command, the transmit CoS is always the incoming packet CoS (or the ingress interface default CoS if the packet is received untagged).

When the interface trust state is not configured to *trust dscp* using the **qos trust dscp** command, the security and QoS ACL classification always use the interface DSCP and not the incoming packet DSCP.

Mapping Tables

During QoS processing, the switch represents the priority of all traffic (including non-IP traffic) with an internal DSCP value:

- During classification, QoS uses configurable mapping tables to derive the internal DSCP (a 6-bit value) from received CoS. These maps include the CoS-to-DSCP map.
- During policing, QoS can assign another DSCP value to an IP or non-IP packet (if the packet is out of profile and the policer specifies a marked down DSCP value). This configurable map is called the policed-DSCP map.
- Before the traffic reaches the scheduling stage, QoS uses the internal DSCP to select one of the four egress queues for output processing. The DSCP-to-egress queue mapping can be configured using the **qos map dscp to tx-queue** command.

The CoS-to-DSCP and DSCP-to-CoS map have default values that might or might not be appropriate for your network.

For configuration information, see the "Configuring DSCP Maps" section on page 32-57.

Queueing and Scheduling

Each physical port has four transmit queues (egress queues). Each packet that needs to be transmitted is enqueued to one of the transmit queues. The transmit queues are then serviced based on the transmit queue scheduling algorithm.

Once the final transmit DSCP is computed (including any markdown of DSCP), the transmit DSCP to transmit queue mapping configuration determines the transmit queue. The packet is placed in the transmit queue of the transmit port, determined from the transmit DSCP. Use the **qos map dscp to tx-queue** command to configure the transmit DSCP to transmit queue mapping. The transmit DSCP is the internal DSCP value if the packet is a non-IP packet as determined by the QoS policies and trust configuration on the ingress and egress ports.

For configuration information, see the "Configuring Transmit Queues" section on page 32-54.

Active Queue Management

Active queue management (AQM) is the pro-active approach of informing you about congestion before a buffer overflow occurs. AQM is done using Dynamic buffer limiting (DBL). DBL tracks the queue length for each traffic flow in the switch. When the queue length of a flow exceeds its limit, DBL drop packets or set the Explicit Congestion Notification (ECN) bits in the packet headers.

DBL classifies flows in two categories, adaptive and aggressive. Adaptive flows reduce the rate of packet transmission once it receives congestion notification. Aggressive flows do not take any corrective action in response to congestion notification. For every active flow the switch maintains two parameters, "buffersUsed" and "credits". All flows start with "max-credits", a global parameter. When a flow with credits less than "aggressive-credits" (another global parameter) it is considered an aggressive flow and is given a small buffer limit called "aggressiveBufferLimit".

Queue length is measured by the number of packets. The number of packets in the queue determines the amount of buffer space that a flow is given. When a flow has a high queue length the computed value is lowered. This allows new incoming flows to receive buffer space in the queue. This allows all flows to get a proportional share of packets through the queue.

Because 4 transmit queues exist per interface and DBL is a per-queue mechanism, DSCP values can make DBL application more complex.

The following table provides the default DSCP-to-transmit queue mapping:

DSCP	txQueue
0-15	1
16-31	2
32-48	3
49-63	4

For example, if you are sending two streams, one with a DSCP of 16 and other with a value of 0, they will transmit from different queues. Even though an aggressive flow in txQueue 2 (packets with DSCP of 16) can saturate the link, packets with a DSCP of 0 will not be blocked by the aggressive flow, as they will transmit from txQueue 1. Thus, even without DBL, packets whose DSCP value places them in txQueue 1, 3, or 4 will not be dropped due to the aggressive flow.

Sharing Link Bandwidth Among Transmit Queues

The four transmit queues for a transmit port share the available link bandwidth of that transmit port. You can set the link bandwidth to be shared differently among the transmit queues using **bandwidth** command in interface transmit queue configuration mode. With this command, you assign the minimum guaranteed bandwidth for each transmit queue.

By default, all queues are scheduled in a round robin manner.

For systems using Supervisor Engine II-Plus, Supervisor Engine II-Plus TS, Supervisor Engine III, and Supervisor Engine IV, bandwidth can be configured on these ports only:

- Uplink ports on supervisor engines
- Ports on the WS-X4306-GB GBIC module
- Ports on the WS-X4506-GB-T CSFP module
- The 2 1000BASE-X ports on the WS-X4232-GB-RJ module
- The first 2 ports on the WS-X4418-GB module
- The two 1000BASE-X ports on the WS-X4412-2GB-TX module

For systems using Supervisor Engine V, bandwidth can be configured on all ports (10/100 Fast Ethernet, 10/100/1000BASE-T, and 1000BASE-X).

Strict Priority / Low Latency Queueing

You can configure transmit queue 3 on each port with higher priority using the **priority high** tx-queue configuration command in the interface configuration mode. When transmit queue 3 is configured with higher priority, packets in transmit queue 3 are scheduled ahead of packets in other queues.

When transmit queue 3 is configured at a higher priority, the packets are scheduled for transmission before the other transmit queues only if it has not met the allocated bandwidth sharing configuration. Any traffic that exceeds the configured shape rate is queued and transmitted at the configured rate. If the burst of traffic, exceeds the size of the queue, packets are dropped to maintain transmission at the configured shape rate.

Traffic Shaping

Traffic Shaping provides the ability to control the rate of outgoing traffic in order to make sure that the traffic conforms to the maximum rate of transmission contracted for it. Traffic that meets certain profile can be shaped to meet the downstream traffic rate requirements to handle any data rate mismatches.

Each transmit queue can be configured to transmit a maximum rate using the **shape** command. The configuration allows you to specify the maximum rate of traffic. Any traffic that exceeds the configured shape rate is queued and transmitted at the configured rate. If the burst of traffic exceeds the size of the queue, packets are dropped to maintain transmission at the configured shape rate.

Packet Modification

A packet is classified, policed, and queued to provide QoS. Packet modifications can occur during this process:

- For IP packets, classification involves assigning a DSCP to the packet. However, the packet is not modified at this stage; only an indication of the assigned DSCP is carried along. The reason for this is that QoS classification and ACL lookup occur in parallel, and it is possible that the ACL specifies that the packet should be denied and logged. In this situation, the packet is forwarded with its original DSCP to the CPU, where it is again processed through ACL software.
- For non-IP packets, classification involves assigning an internal DSCP to the packet, but because there is no DSCP in the non-IP packet, no overwrite occurs. Instead, the internal DSCP is used both for queueing and scheduling decisions and for writing the CoS priority value in the tag if the packet is being transmitted on either an ISL or 802.1Q trunk port.
- During policing, IP and non-IP packets can have another DSCP assigned to them (if they are out of profile and the policer specifies a markdown DSCP). Once again, the DSCP in the packet is not modified, but an indication of the marked-down value is carried along. For IP packets, the packet modification occurs at a later stage.

Per Port Per VLAN QoS

Per-port per-VLAN QoS (PVQoS) offers differentiated quality-of-services to individual VLANs on a trunk port. It enables service providers to rate limit individual VLAN-based services on each trunk port to a business or a residence. In an enterprise Voice-over-IP environment, it can be used to rate limit voice VLAN even if an attacker impersonates an IP phone. A per-port per-VLAN service policy can be separately applied to either ingress or egress traffic.

QoS and Software Processed Packets

The Catalyst 4500 platform does not apply the QoS marking or policing configuration for any packets that are forwarded or generated by the Cisco IOS software. This means that any input or output QoS policy configured on the port or VLAN is not applied to packets if the Cisco IOS is forwarding or generating packets.

However, Cisco IOS marks all the generated control packets appropriately and uses the internal IP DSCP to determine the transmit queue on the output transmission interface. For IP packets, the internal IP DSCP is the IP DSCP field in the IP packet. For non-IP packets, Cisco IOS assigns a packet priority internally and maps it to an internal IP DSCP value.

Cisco IOS assigns an IP precedence of 6 to routing protocol packets on the control plane. As noted in RFC 791, "The Internetwork Control designation is intended for use by gateway control originators only." Specifically, Cisco IOS marks the following IP-based control packets: Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP) hellos, and keepalives. Telnet packets to and from the router also receive an IP precedence value of 6. The assigned value remains with the packets when the output interface transmits them into the network.

For Layer 2 control protocols, the software assigns an internal IP DSCP. Typically, Layer 2 control protocol packets are assigned an internal DSCP value of 48 (corresponding to an IP precedence value of 6).

The internal IP DSCP is used to determine the transmit queue to which the packet is enqueued on the transmission interface. See “Configuring Transmit Queues” on page 54 for details on how to configure the DSCP to transmit queues.

The internal IP DSCP is also used to determine the transmit CoS marking if the packet is transmitted with a IEEE 802.1q or ISL tag on a trunk interface. See “Configuring the DSCP-to-CoS Map” on page 59 for details on how to configure the DSCP to CoS mapping.

Configuring Auto-QoS on Supervisor Engines II-Plus, II+10GE, VI, V, V-10GE, 4924, 4948, and 4948-10GE

You can use the auto-QoS feature to simplify the deployment of existing QoS features. Auto-QoS makes assumptions about the network design, and as a result, the switch can prioritize different traffic flows and appropriately use the egress queues instead of using the default QoS behavior. (The default is that QoS is disabled. The switch then offers best-effort service to each packet, regardless of the packet content or size, and sends it from a single queue.)

When you enable auto-QoS, it automatically classifies traffic based on ingress packet label. The switch uses the resulting classification to choose the appropriate egress queue.

You use auto-QoS commands to identify ports connected to Cisco IP phones and to identify ports that receive trusted voice over IP (VoIP) traffic through an uplink. Auto-QoS then performs these functions:

- Detects the presence or absence of IP phones
- Configures QoS classification
- Configures egress queues

These sections describe how to configure auto-QoS on your switch:

- Generated Auto-QoS Configuration, page 32-17
- Effects of Auto-QoS on the Configuration, page 32-19
- Configuration Guidelines, page 32-19
- Enabling Auto-QoS for VoIP, page 32-19

Generated Auto-QoS Configuration

By default, auto-QoS is disabled on all interfaces.

When you enable the auto-QoS feature on the first interface, these automatic actions occur:

- QoS is globally enabled (**qos** global configuration command).

- DBL is enabled globally (**qos dbl** global configuration command)
- When you enter the **auto qos voip trust** interface configuration command, the ingress classification on the specified interface is set to trust the CoS label received in the packet if the specified interface is configured as Layer 2 (and is set to trust DSCP if the interface is configured as Layer 3). (See Table 32-2.)
- When you enter the **auto qos voip cisco-phone** interface configuration command, the trusted boundary feature is enabled. It uses the Cisco Discovery Protocol (CDP) to detect the presence or absence of a Cisco IP phone. When a Cisco IP phone is detected, the ingress classification on the interface is set to trust the CoS label received in the packet, if the interface is configured as Layer 2. (The classification is set to trust DSCP if the interface is configured as Layer 3.) When a Cisco IP phone is absent, the ingress classification is set to not trust the CoS label in the packet.

  For information about the trusted boundary feature, see the "Configuring a Trusted Boundary to Ensure Port Security" section on page 32-26.

When you enable auto-QoS by using the **auto qos voip cisco-phone** or the **auto qos voip trust** interface configuration commands, the switch automatically generates a QoS configuration based on the traffic type and ingress packet label and applies the commands listed in Table 32-2 to the interface.

Table 32-2 Generated Auto-QoS Configuration

Description	Automatically Generated Command

The switch automatically enables standard QoS and DBL configures the CoS-to-DSCP map (maps CoS values in incoming packets to a DSCP value).	`Switch(config)# qos`
	`Switch(config)# qos map cos 3 to 26`
	`Switch(config)# qos dbl`
	`Switch(config)# qos map cos 5 to 46`

The switch automatically configures the DSCP-to-Tx-queue mapping.	`Switch(config)# qos map dscp 24 25 26 27 b28 29 30 31 to tx-queue 4`
	`Switch(config)# qos map dscp 32 33 34 35 36 37 38 39 to tx-queue 4`

The switch automatically sets the ingress classification on the interface to trust the CoS/DSCP value received in the packet.	`Switch(config-if)# qos trust cos`
	or
	`Switch(config-if)# qos trust dscp`

The switch automatically creates a QoS service policy, enables DBL on the policy, and attaches it to the interface.	`Switch(config)# policy-map autoqos-voip-policy`
	`Switch(config-pmap)# class class-default`
	`Switch(config-pmap-c)# dbl`

If you entered the **auto qos voip cisco-phone** command, the switch automatically enables the trusted boundary feature, which uses the CDP to detect the presence or absence of a Cisco IP phone.	`Switch(config-if)# qos trust device cisco-phone`

The switch assigns a higher priority for queue 3. Limit for shaping on queue 3 is selected so that it is 33 percent of the link speed. Configure shaping as 33 percent on those ports where sharing is supported.
This procedure ensures that the higher-priority queue does not starve other queues.
	`Switch(config-if)# tx-queue 3`
	`Switch(config-if-tx-queue)# priority high`
	`Switch(config-if-tx-queue)# shape percent 33`
	`Switch(config-if-tx-queue)# bandwidth percent 33`

Effects of Auto-QoS on the Configuration

When auto-QoS is enabled, the **auto qos voip** interface configuration command and the generated configuration are added to the running configuration.

Configuration Guidelines

Before configuring auto-QoS, you should be aware of this information:

- In this release, auto-QoS configures the switch only for VoIP with Cisco IP phones.
- To take advantage of the auto-QoS defaults, do not configure any standard-QoS commands before entering the auto-QoS commands. If necessary, you can fine-tune the QoS configuration, but we recommend that you do so only after the auto-QoS configuration is completed.
- You can enable auto-QoS on static, dynamic-access, voice VLAN access, and trunk ports.
- By default, the CDP is enabled on all interfaces. For auto-QoS to function properly, do not disable the CDP.
- To enable **auto qos voip trust** on Layer 3 interfaces, change the port to Layer 3, then apply auto-QoS to make it trust DSCP.

Enabling Auto-QoS for VoIP

To enable auto-QoS for VoIP within a QoS domain, perform this task:

	Command	Purpose
Step 1	`Switch# debug auto qos`	(Optional) Enables debugging for auto-QoS. When debugging is enabled, the switch displays the QoS commands that are automatically generated and applied when auto-QoS is enabled or disabled.
Step 2	`Switch# configure terminal`	Enters global configuration mode.
Step 3	`Switch(config)# interface interface-id`	Enters interface configuration mode, and specify the interface that is connected to a Cisco IP phone or the uplink interface that is connected to another switch or router in the interior of the network.
Step 4	`Switch(config-if)# auto qos voip {cisco-phone | trust}`	Enables auto-QoS. The keywords have these meanings: • **cisco-phone**—If the interface is connected to a Cisco IP phone, the CoS labels of incoming packets are trusted only when the telephone is detected. • **trust**—The uplink interface is connected to a trusted switch or router, and the VoIP traffic classification in the ingress packet is trusted.

	Command	Purpose
Step 5	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 6	`Switch# show auto qos interface interface-id`	Verifies your entries. This command displays the auto-QoS configuration that was initially applied; it does not display any user changes to the configuration that might be in effect.

To disable auto-QoS on an interface, use the **no auto qos voip** interface configuration command. When you enter this command, the switch changes the auto-QoS settings to the standard-QoS default settings for that interface. It does not change any global configuration performed by auto-QoS. Global configuration remains the same.

This example shows how to enable auto-QoS and to trust the CoS labels in incoming packets when the device connected to Fast Ethernet interface 1/1 is detected as a Cisco IP phone:

```
Switch(config)# interface fastethernet1/1
Switch(config-if)# auto qos voip cisco-phone
```

This example shows how to enable auto-QoS and to trust the CoS/DSCP labels in incoming packets when the switch or router connected to Gigabit Ethernet interface 1/1 is a trusted device:

```
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# auto qos voip trust
```

This example shows how to display the QoS commands that are automatically generated when auto-QoS is enabled:

```
Switch# debug auto qos
AutoQoS debugging is on
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# auto qos voip cisco-phone
```

Displaying Auto-QoS Information

To display the initial auto-QoS configuration, use the **show auto qos** [**interface** [*interface-id*]] privileged EXEC command. To display any user changes to that configuration, use the **show running-config** privileged EXEC command. You can compare the **show auto qos** and the **show running-config** command output to identify the user-defined QoS settings.

To display information about the QoS configuration that might be affected by auto-QoS, use one of these commands:

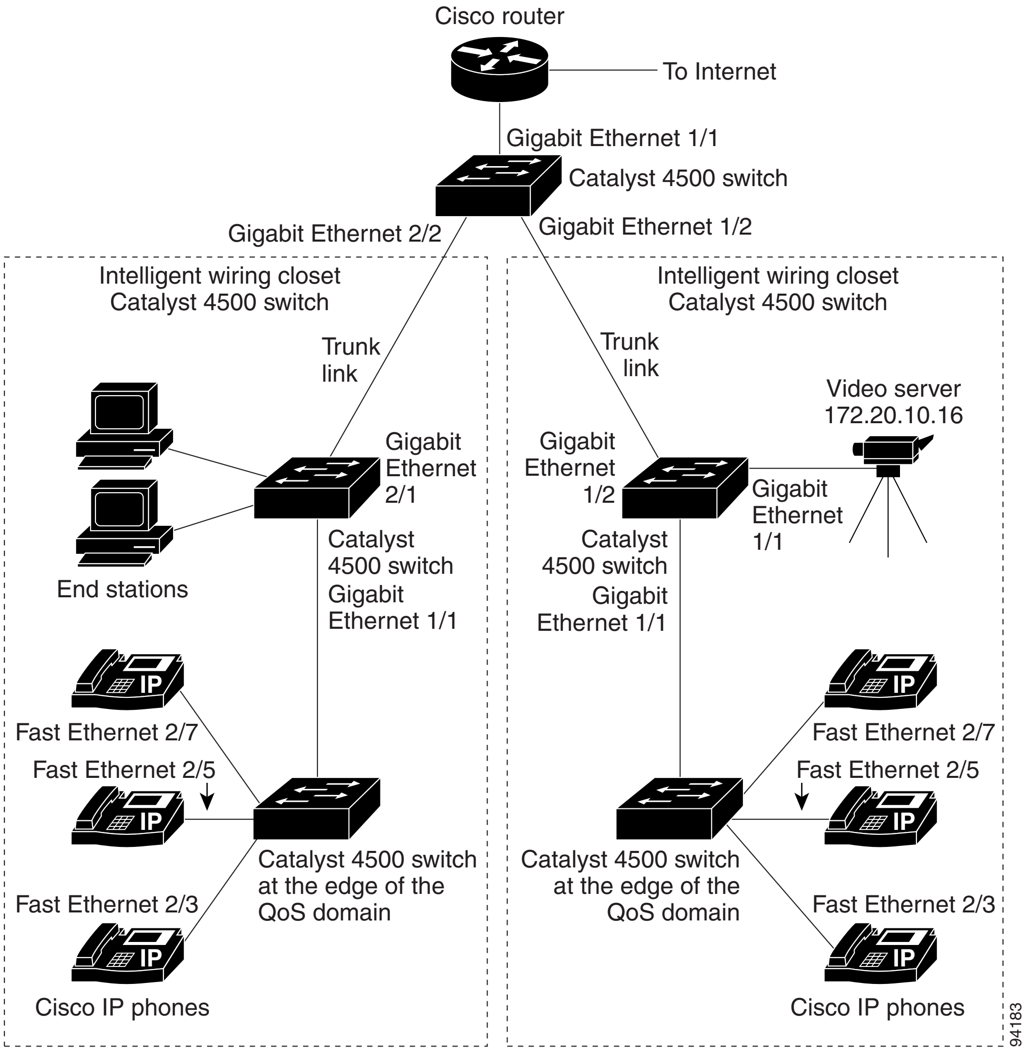
- **show qos**
- **show qos map**
- **show qos interface** [*interface-id*]

For more information about these commands, refer to the command reference for this release.

Auto-QoS Configuration Example

This section describes how you could implement auto-QoS in a network, as shown in Figure 32-5.

Figure 32-5 Auto-QoS Configuration Example Network



The intelligent wiring closets in Figure 32-5 are composed of Catalyst 4500 switches. The object of this example is to prioritize the VoIP traffic over all other traffic. To do so, enable auto-QoS on the switches at the edge of the QoS domains in the wiring closets.

Note You should not configure any standard QoS commands before entering the auto-QoS commands. You can fine-tune the QoS configuration, but we recommend that you do so only after the auto-QoS configuration is completed.

To configure the switch at the edge of the QoS domain to prioritize the VoIP traffic over all other traffic, perform this task:

	Command	Purpose
Step 1	`Switch# debug auto qos`	Enables debugging for auto-QoS. When debugging is enabled, the switch displays the QoS configuration that is automatically generated when auto-QoS is enabled.
Step 2	`Switch# configure terminal`	Enters global configuration mode.
Step 3	`Switch(config)# cdp enable`	Enables CDP globally. By default, CDP is enabled.
Step 4	`Switch(config)# interface fastethernet2/3`	Enters interface configuration mode.
Step 5	`Switch(config-if)# auto qos voip cisco-phone`	Enables auto-QoS on the interface, and specifies that the interface is connected to a Cisco IP phone. The CoS labels of incoming packets are trusted only when the IP phone is detected.
Step 6	`Switch(config)# interface fastethernet2/5`	Enters interface configuration mode.
Step 7	`Switch(config)# auto qos voip cisco-phone`	Enables auto-QoS on the interface, and specifies that the interface is connected to a Cisco IP phone.
Step 8	`Switch(config)# interface fastethernet2/7`	Enters interface configuration mode.
Step 9	`Switch(config)# auto qos voip cisco-phone`	Enables auto-QoS on the interface, and specifies that the interface is connected to a Cisco IP phone.
Step 10	`Switch(config)# interface gigabit1/1`	Enters interface configuration mode.
Step 11	`Switch(config)# auto qos voip trust`	Enables auto-QoS on the interface, and specifies that the interface is connected to a trusted router or switch.
Step 12	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 13	`Switch# show auto qos`	Verifies your entries. This command displays the auto-QoS configuration that is initially applied; it does not display any user changes to the configuration that might be in effect. For information about the QoS configuration that might be affected by auto-QoS, see the "Displaying Auto-QoS Information" section on page 32-20.
Step 14	`Switch# show auto qos interface` *`interface-id`*	Verifies your entries. This command displays the auto-QoS configuration that was initially applied; it does not display any user changes to the configuration that might be in effect.
Step 15	`Switch# copy running-config startup-config`	Saves the **auto qos voip** interface configuration commands and the generated auto-QoS configuration in the configuration file.

Configuring QoS on Supervisor Engines II-Plus, II+10GE, VI, V, V-10GE, 4924, 4948, and 4948-10GE

Before configuring QoS, you must have a thorough understanding of these items:

- The types of applications used and the traffic patterns on your network.
- Traffic characteristics and needs of your network. Is the traffic bursty? Do you need to reserve bandwidth for voice and video streams?
- Bandwidth requirements and speed of the network.
- Location of congestion points in the network.

These sections describe how to configure QoS on the Catalyst 4000 family switch:

- Default QoS Configuration, page 32-23
- Configuration Guidelines, page 32-25
- Enabling QoS Globally, page 32-25
- Configuring a Trusted Boundary to Ensure Port Security, page 32-26
- Enabling Dynamic Buffer Limiting, page 32-27
- Creating Named Aggregate Policers, page 32-31
- Configuring a QoS Policy, page 32-32
- Configuring CoS Mutation, page 32-40
- Configuring User Based Rate Limiting, page 32-42
- Enabling Per-Port Per-VLAN QoS, page 32-48
- Enabling or Disabling QoS on an Interface, page 32-50
- Configuring VLAN-Based QoS on Layer 2 Interfaces, page 32-51
- Configuring the Trust State of Interfaces, page 32-52
- Configuring the CoS Value for an Interface, page 32-53
- Configuring DSCP Values for an Interface, page 32-54
- Configuring Transmit Queues, page 32-54
- Configuring DSCP Maps, page 32-57
- Enabling Layer 2 Control Packet QoS, page 32-60

Default QoS Configuration

Table 32-3 shows the QoS default configuration.

Table 32-3 QoS Default Configuration

Feature	Default Value
Global QoS configuration	Disabled
Interface QoS configuration (port based)	Enabled when QoS is globally enabled
Interface CoS value	0

Table 32-3 QoS Default Configuration (continued)

Feature	Default Value
Interface DSCP value	0
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
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 (Policed-DSCP)	Marked-down DSCP value equals original DSCP value (no markdown)
Policers	None
Policy maps	None
Transmit queue sharing	1/4 of the link bandwidth
Transmit queue size	1/4 of the transmit queue entries for the port. The transmit queue size of a port depends on the type of port, ranging from 240 packets per transmit queue to 1920 packets per transmit queue.
Transmit queue shaping	None
DCSP-to-Transmit queue map	DSCP 0–15 Queue 1 DSCP 16–31 Queue 2 DSCP 32–47 Queue 3 DSCP 48–63 Queue 4
High priority transmit queue	Disabled
With QoS disabled	
Interface trust state	Trust DSCP
With QoS enabled	With QoS enabled and all other QoS parameters at default values, QoS sets IP DSCP to zero and Layer 2 CoS to zero in all traffic transmitted.
Interface trust state	Untrusted

Configuration Guidelines

Before beginning the QoS configuration, you should be aware of this information:

- If you have EtherChannel ports configured on your switch, you must configure QoS classification and policing on the EtherChannel. The transmit queue configuration must be configured on the individual physical ports that comprise the EtherChannel.
- If the ip fragments match the source and destination configured in the ACL used to classify the traffic for quality of service, but do not match the layer 4 port numbers in the ACL, they are still matched with the ACL and may get prioritized. If the desired behavior is to give best effort service to ip fragments, following two ACEs should be added to the ACL used to classify the traffic.

  ```
  access-list xxx deny udp any any fragments
  access-list xxx deny tcp any any fragments
  ```

- It is not possible to match IP options against configured IP extended ACLs to enforce QoS. These packets are sent to the CPU and processed by software. IP options are denoted by fields in the IP header.
- Control traffic (such as spanning-tree BPDUs and routing update packets) received by the switch are subject to all ingress QoS processing.
- You cannot use **set** commands in policy maps if ip routing is disabled (enabled by default).
- On a dot1q tunnel port, only Layer 2 match criteria can be applied to tagged packets. However, all match criteria can be applied for untagged packets.
- On a trunk port, only Layer 2 match criteria can be applied to packets with multiple 802.1q tags.

Note QoS processes both unicast and multicast traffic.

Enabling QoS Globally

To enable QoS globally, perform this task:

	Command	Purpose
Step 1	`Switch# conf terminal`	Enter configuration mode.
Step 2	`Switch(config)# qos`	Enables QoS on the switch. Use the **no qos** command to globally disable QoS.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show qos`	Verifies the configuration.

This example shows how to enable QoS globally and verify the configuration:

```
Switch# config terminal
Switch(config)# qos
Switch(config)# end
Switch#
Switch# show qos
  QoS is enabled globally

Switch#
```

Configuring a Trusted Boundary to Ensure Port Security

In a typical network, you connect a Cisco IP phone to a switch port as discussed in Chapter 33, “Configuring Voice Interfaces.” Traffic sent from the telephone to the switch is typically marked with a tag that uses the 802.1Q header. The header contains the VLAN information and the class of service (CoS) 3-bit field, which determines the priority of the packet. For most Cisco IP phone configurations, the traffic sent from the telephone to the switch is trusted to ensure that voice traffic is properly prioritized over other types of traffic in the network. By using the **qos trust cos** interface configuration command, you can configure the switch port to which the telephone is connected to trust the CoS labels of all traffic received on that port.

Note Starting with Cisco IOS Release 12.2(31)SG, Supervisor Engine V-10GE enables you to classify traffic based on packet's IP DSCP value irrespective of the port trust state. Because of this, even when a Cisco IP phone is not detected, data traffic can be classified based on IP DSCP values. Output queue selection is not impacted by this new behavior. It is still based on the incoming port trust configuration. For information on configuring transmit queues, refer to the “Configuring Transmit Queues” section on page 32-54".

In some situations, you also might connect a PC or workstation to the IP phone. In this case, you can use the **switchport priority extend cos** interface configuration command to configure the telephone through the switch CLI to override the priority of the traffic received from the PC. With this command, you can prevent a PC from taking advantage of a high-priority data queue.

However, if a user bypasses the telephone and connects the PC directly to the switch, the CoS labels generated by the PC are trusted by the switch (because of the trusted CoS setting) and can allow misuse of high-priority queues. The trusted boundary feature solves this problem by using the CDP to detect the presence of a Cisco IP phone (such as the Cisco IP Phone 7910, 7935, 7940, and 7960) on a switch port.

Note If CDP is not running on the switch globally or on the port in question, trusted boundary does not work.

When you configure trusted boundary on a port, trust is disabled. Then, when a phone is plugged in and detected, trust is enabled. (It may take a few minutes to detect the phone.) Now, when a phone is unplugged (and not detected), the trusted boundary feature disables the trusted setting on the switch port and prevents misuse of a high-priority queue.

To enable trusted boundary on a port, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode, and specifies the interface connected to the IP phone. Valid interfaces include physical interfaces.
Step 3	`Switch(config)#` **`qos trust [cos	dscp]`**	Configures the interface to trust the CoS value in received traffic. By default, the port is not trusted.
Step 4	`Switch(config)#` **`qos trust device cisco-phone`**	Specifies that the Cisco IP phone is a trusted device. You cannot enable both trusted boundary and auto-QoS (**auto qos voip** interface configuration command) at the same time; they are mutually exclusive.

	Command	Purpose
Step 5	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 6	`Switch#` **`show qos interface`** *`interface-id`*	Verifies your entries.
Step 7	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To disable the trusted boundary feature, use the **no qos trust device cisco-phone** interface configuration command.

Enabling Dynamic Buffer Limiting

Note Supervisor Engine 6-E does *not* support this feature.

Dynamic Buffer Limiting (DBL) provides active queue management on Cat4500 platforms. (Refer to "Active Queue Management" section on page 32-14 for details.)

Through "selective" DBL, you can select the flows that would be subjected (or would not be subjected) to the DBL algorithm. You ca n enable DBL globally, on specific IP DSCP values, or on specific CoS values.

The following tasks are discussed:

- Enabling DBL Globally, page 32-27
- Selectively Enable DBL, page 32-28

Enabling DBL Globally

To enable DBL globally on the switch, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`qos dbl`**	Enables DBL on the switch. Use the **no qos dbl** command to disable AQM.
Step 2	`Switch(config)#` **`end`**	Exits configuration mode.
Step 3	`Switch#` **`show qos dbl`**	Verifies the configuration.

This example shows how to enable DBL globally and verify the configuration:

```
Switch# configure terminal
Switch(config)# qos dbl
Global DBL enabled
Switch(config)# end
Switch# show qos dbl
    QOS is enabled globally
    DBL is enabled globally on DSCP values:
          0-63
    DBL flow includes vlan
    DBL flow includes layer4-ports
    DBL does not use ecn to indicate congestion DBL exceed-action probability: 15% DBL max
    credits: 15 DBL aggressive credit limit: 10 DBL aggressive buffer limit: 2 packets
Switch#
```

You can enable DBL on the egress interface direction by applying a service-policy:

```
Switch# conf terminal
Switch(config)# policy-map dbl
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# dbl
Switch(config-pmap-c)# end
Switch#
00:08:12: %SYS-5-CONFIG_I: Configured from console by console
Switch# conf terminal
Switch(config)# int gig 1/2
Switch(config-if)# service-policy output dbl
Switch(config-if)# end
Switch#
```

Selectively Enable DBL

DSCP values enable you to selectively apply DBL for IP Packets only (single or untagged). (Refer to the “Enable DBL on Specific IP DSCP Values” section on page 32-28.) To selectively apply DBL for non-IP packets or double-tagged packets (like Q-in-Q), you must use COS values as in the following section. (Refer to the “Enable DBL on Specific CoS Values” section on page 32-29.)

You can do the following:

- Enable DBL on Specific IP DSCP Values, page 32-28
- Enable DBL on Specific CoS Values, page 32-29

Enable DBL on Specific IP DSCP Values

DBL action is performed on transmit queues (4 per interface). You govern the mapping from IP DSCP to transmit queues with the **qos map dscp dscp-values to tx-queue queue-id** command. (Refer to “Configuring Transmit Queues” section on page 32-54 for details on how to do this.)

To enable DBL on specific IP DSCP values, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [`**`no`**`]` **`qos dbl dscp-based`** `<`*`value, value_range`*`>`	Enables DBL on specific IP DSCP values.
Step 2	`Switch(config)#` **`end`**	Exits configuration mode.
Step 3	`Switch#` **`show qos dbl`**	Verifies the configuration.

This example shows how to selectively enable DBL on the DSCP values 1 through 10:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos dbl dscp-based 1-10
Switch(config)# end
Switch# show qos dbl
    QOS is enabled globally
    DBL is enabled globally on DSCP values:
          1-10
    DBL flow includes vlan
    DBL flow includes layer4-ports
    DBL does not use ecn to indicate congestion DBL exceed-action probability: 15%
    DBL max credits: 15
    DBL aggressive credit limit: 10
    DBL aggressive buffer limit: 2 packets
Switch#
```

This example shows how to selectively disable DBL on DSCP values 1 through 10 and to verify the configuration:

```
Switch# configure terminal
Switch(config)# no qos dbl dscp-based 1-5, 7
Switch(config)# end
Switch# show qos dbl
    QOS is enabled globally
    DBL is enabled globally on DSCP values:
          6,8-10
    DBL flow includes vlan
    DBL flow includes layer4-ports
    DBL does not use ecn to indicate congestion DBL exceed-action probability: 15% DBL max
    credits: 15 DBL aggressive credit limit: 10 DBL aggressive buffer limit: 2 packets
Switch#
```

Although you apply DBL based on class attributes other than DSCP, you still need to attach a policy-map to an egress interface (“Configuring Policy-Map Class Actions” section on page 32-36).

Provided the value has been set according to your network policies, you must configure “trust DSCP” on the ingress interface of the aggressive flow that DBL will throttle:

```
Interface <ingress>
  qos trust dscp
```

Enable DBL on Specific CoS Values

You might need to use COS values to selectively applying DBL if you intend to use non-IP packets or double-tagged packets (for example, Q-in-Q).

For single-tagged IP packets, use the following approach. Specify the global **qos dbl** dscp-based command as shown in the “Enable DBL on Specific IP DSCP Values” section on page 32-28).

```
Interface <ingress>
  switchport mode trunk
  qos trust cos
```

For non-IP packets or double-tagged packets, use the following method:

	Command	Purpose
Step 1	Switch(config)# **qos dbl**	Enables DBL globally.
Step 2	Switch(config)# **end**	Exits configuration mode.
Step 3	Switch(config)# **class-map cos**	Defines a traffic class.
Step 4	Switch(config-cmap)# **match cos x y**	Specifies CoS values used as match criteria.
Step 5	Switch(config-cmap)# **exit**	Returns to global configuration mode.
Step 6	Switch(config)# **policy-map cos**	Creates a policy map with a user-specified name.
Step 7	Switch(config-pmap)# **class cos**	Specifies the class map to be used by the policy map.
Step 8	Switch(config-pmap-c)# **dbl**	Enables DBL on the policy.
Step 9	Switch(config-pmap-c)# **end**	Returns to EXEC mode.

	Command	Purpose
Step 10	`Switch# show policy-map cos`	Verifies configuration.
Step 11	`Switch# configure terminal`	Enters global configuration mode.
Step 12	`Switch(config)# interface gigabitEthernet 1/20`	Applies the configuration to an interface.
Step 13	`Switch(config-if)# service-policy output cos`	Attaches the policy map to the interface.
Step 14	`Switch# show policy-map interface`	Verifies the configuration.

Note For more details on using CoS Mutation, refer to the “Configuring CoS Mutation” section on page 32-40.

To selectively enable DBL on CoS values 2 and 3:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos dbl
Switch(config)# end
Switch# configure terminal
Switch(config)# class-map cos
Switch(config-cmap)# match cos 2 3
Switch(config-cmap)# exit
Switch(config)# policy-map cos
Switch(config-pmap)# class cos
Switch(config-pmap-c)# dbl
Switch(config-pmap-c)# end
Switch# show policy-map cos
    Policy Map cos
      Class cos
        dbl
Switch# configure terminal
Switch(config)# interface gigabitEthernet 1/20
Switch(config-if)# service-policy output cos
Switch# show policy-map interface
 GigabitEthernet1/20

  Service-policy output: cos

    Class-map: cos (match-all)
      0 packets
      Match: cos  2  3
      dbl

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
```

Creating Named Aggregate Policers

To create a named aggregate policer, perform this task:

Command	Purpose
`Switch(config)#` **`qos aggregate-policer`** ***`policer_name`*** *`rate burst`* `[[`**`conform-action`** `{`**`transmit`** `|` **`drop`**`}]` `[`**`exceed-action`** `{`**`transmit`** `|` **`drop`** `|` **`policed-dscp-transmit`**`}]]`	Creates a named aggregate policer.

An aggregate policer can be applied to one or more interfaces. However, if you apply the same policer to the input direction on one interface and to the output direction on a different interface, then you have created the equivalent of two different aggregate policers in the switching engine. Each policer has the same policing parameters, with one policing the ingress traffic on one interface and the other policing the egress traffic on another interface. If an aggregate policer is applied to multiple interfaces in the same direction, then only one instance of the policer is created in the switching engine.

Similarly, an aggregate policer can be applied to a port or to a VLAN. If you apply the same aggregate policer to a port and to a VLAN, then you have created the equivalent of two different aggregate policers in the switching engine. Each policer has the same policing parameters, with one policing the traffic on the configured port and the other policing the traffic on the configured VLAN. If an aggregate policer is applied to only ports or only VLANs, then only one instance of the policer is created in the switching engine.

In effect, if you apply a single aggregate policer to ports and VLANs in different directions, then you have created the equivalent of four aggregate policers; one for all ports sharing the policer in input direction, one for all ports sharing the policer in output direction, one for all VLANs sharing the policer in input direction and one for all VLANs sharing the policer in output direction.

When creating a named aggregate policer, note the following:

- The valid range of values for the *rate* parameter is as follows:
  - Minimum—32 kilobits per second
  - Maximum—32 gigabits per second

  See the “Configuration Guidelines” section on page 32-25.
- Rates can be entered in bits-per-second, or you can use the following abbreviations:
  - k to denote 1000 bps
  - m to denote 1000000 bps
  - g to denote 1000000000 bps

  Note You can also use a decimal point. For example, a rate of 1,100,000 bps can be entered as 1.1m.

- The valid range of values for the *burst* parameter is as follows:
  - Minimum—1 kilobyte
  - Maximum—512 megabytes

- Bursts can be entered in bytes, or you can use the following abbreviation:
  - k to denote 1000 bytes
  - m to denote 1000000 bytes
  - g to denote 1000000000 bytes

Note You can also use a decimal point. For example, a burst of 1,100,000 bytes can be entered as 1.1m.

- Optionally, you can specify a conform action for matched in-profile traffic as follows:
  - The default conform action is **transmit**.
  - Enter the **drop** keyword to drop all matched traffic.

Note When you configure **drop** as the conform action, QoS configures **drop** as the exceed action.

- Optionally, for traffic that exceeds the CIR, you can specify an exceed action as follows:
  - The default exceed action is **drop**.
  - Enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map.
  - For no policing, enter the **transmit** keyword to transmit all matched out-of-profile traffic.
- You can enter the **no qos aggregate-policer** *policer_name* command to delete a named aggregate policer.

This example shows how to create a named aggregate policer with a 10 Mbps rate limit and a 1-MB burst size that transmits conforming traffic and marks down out-of-profile traffic.

```
Switch# config terminal
Switch(config)# qos aggregate-policer aggr-1 10000000 1000000 conform-action transmit
exceed-action policed-dscp-transmit
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos aggregate-policer aggr-1
Policer aggr-1
  Rate(bps):10000000 Normal-Burst(bytes):1000000
  conform-action:transmit exceed-action:policed-dscp-transmit
  Policymaps using this policer:
Switch#
```

Configuring a QoS Policy

The following subsections describe QoS policy configuration:

- Overview of QoS Policy Configuration, page 32-33
- Configuring a Class Map (Optional), page 32-33
- Configuring a Policy Map, page 32-35
- Attaching a Policy Map to an Interface, page 32-40

Note QoS policies process both unicast and multicast traffic.

Overview of QoS Policy Configuration

Configuring a QoS policy requires you to configure traffic classes and the policies that will be applied to those traffic classes, and to attach the policies to interfaces using these commands:

- **access-list** (optional for IP traffic—you can filter IP traffic with **class-map** commands):
  - QoS supports these access list types:

Protocol	Numbered Access Lists?	Extended Access Lists?	Named Access Lists?
IP	Yes: 1 to 99 1300 to 1999	Yes: 100 to 199 2000 to 2699	Yes

  - See Chapter 39, "Configuring Network Security with ACLs," for information about ACLs on the Catalyst 4500 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 (Optional)" section on page 32-33.)
- **policy-map**—Enter the **policy-map** command to define the following for each class of traffic:
  - Internal DSCP source
  - Aggregate or individual policing and marking
- **service-policy**—Enter the **service-policy** command to attach a policy map to an interface.

Configuring a Class Map (Optional)

The following subsections describe class map configuration:

- Creating a Class Map, page 32-33
- Configuring Filtering in a Class Map, page 32-34
- Verifying Class Map Configuration, page 32-35

Enter the **class-map** configuration command to define a traffic class and the match criteria that will be used to identify traffic as belonging to that class. Match statements can include criteria such as an ACL, an IP precedence value, or a DSCP value. The match criteria are defined with one match statement entered within the class-map configuration mode.

Creating a Class Map

To create a class map, perform this task:

Command	Purpose
`Switch(config)# [no] class-map [match-all | match-any] class_name`	Creates a named class map. Use the **no** keyword to delete a class map.

Configuring Filtering in a Class Map

To configure filtering in a class map, perform one of these tasks:

Command	Purpose
Switch(config-cmap)# [**no**] **match access-group** **{***acl_index* | **name** *acl_name*}	(Optional) Specifies the name of the ACL used to filter traffic. Use the **no** keyword to remove the statement from a class map. **Note** Access lists are not documented in this publication. See the reference under **access-list** in the "Configuring a QoS Policy" section on page 32-32.
Switch (config-cmap)# [**no**] **match ip precedence** *ipp_value1* [*ipp_value2* [*ipp_valueN*]]	(Optional—for IP traffic only) Specifies up to eight IP precedence values used as match criteria. Use the **no** keyword to remove the statement from a class map.
Switch (config-cmap)# [**no**] **match ip dscp** *dscp_value1* [*dscp_value2* [*dscp_valueN*]]	(Optional—for IP traffic only) Specifies up to eight DSCP values used as match criteria. Use the **no** keyword to remove the statement from a class map.
Switch (config-cmap)# [**no**] **match cos** *value1* [*value2*] [*value3*} [*value4*}	(Optional—for non-IPV4 traffic only) Specifies up to eight CoS values used as match criteria. Use the **no** keyword to remove the statement from a class map. For information on non-IPV4 traffic, see "Configuration Guidelines" section on page 32-19.
Switch (config-cmap)# [**no**] **match any**	(Optional) Matches any IP traffic or non-IP traffic.
Switch (config-cmap)# **match flow ip** {**source-address** | **destination-address**	(Optional) Treats each flow with a unique IP source address or destination address as a new flow.

Note Any Input or Output policy that uses a class map with the **match ip precedence** or **match ip dscp** class-map commands, requires that you configure the port on which the packet is received to **trust dscp**. If not, the IP packet DSCP/IP-precedence is not used for matching the traffic; instead, the receiving port's default DSCP is used. Starting with Cisco IOS Release 12.2(31)SG, the Supervisor Engine V-10GE enables you to classify traffic based on packet's IP DSCP value irrespective of port trust state.

Note With Cisco IOS Release 12.2(31), the Catalyst 4500 series switch supports Match CoS.

Note The interfaces on the Catalyst 4000 family switch do not support the **match classmap**, **match destination-address**, **match input-interface**, **match mpls**, **match not**, **match protocol**, **match qos-group**, and **match source-address** keywords.

Verifying Class Map Configuration

To verify class-map configuration, perform this task:

	Command	Purpose
Step 1	`Switch (config-cmap)# end`	Exits configuration mode.
Step 2	`Switch# show class-map class_name`	Verifies the configuration.

This example shows how to create a class map named *ipp5* and how to configure filtering to match traffic with IP precedence 5:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map ipp5
Switch(config-cmap)# match ip precedence 5
Switch(config-cmap)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show class-map ipp5
 Class Map match-all ipp5 (id 1)
   Match ip precedence 5

Switch#
```

This example shows how to configure match CoS for non-IPV4 traffic and how to configure filtering to match traffic with CoS value of 5:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map maptwo
Switch(config-cmap)# match cos 5
Switch(config-cmap)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show class-map maptwo
 Class Map match-all maptwo (id 1)
   Match cos 5

Switch#
```

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 match criteria and policers.

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. QoS does not attempt to apply commands from more than one policy-map class to matched traffic.

The following sections describe policy-map configuration:

- Creating a Policy Map, page 32-36
- Configuring Policy-Map Class Actions, page 32-36

Creating a Policy Map

To create a policy map, perform this task:

Command	Purpose
`Switch(config)# [no] policy-map policy_name`	Creates a policy map with a user-specified name. Use the **no** keyword to delete the policy map.

Configuring Policy-Map Class Actions

These sections describe policy-map class action configuration:

- Configuring the Policy-Map Marking State, page 32-36
- Configuring the Policy-Map Class Trust State, page 32-36
- Configuring the Policy Map Class DBL State, page 32-37
- Configuring Policy-Map Class Policing, page 32-37
- Using a Named Aggregate Policer, page 32-37
- Configuring a Per-Interface Policer, page 32-37

Configuring the Policy-Map Marking State

To configure the policy map to mark the IP precedence or dscp of a packet, perform this task:

Command	Purpose
`Switch(config-pmap-c)# [no] set ip [precedence prec_value | dscp dscp_value]`	Configures the policy-map marking state, which decides the internal DSCP of the packet for subsequent processing. Use the **no** keyword to clear a configured value and return to the default.

Configuring the Policy-Map Class Trust State

To configure the policy-map class trust state, perform this task:

Command	Purpose
`Switch(config-pmap-c)# [no] trust {cos | dscp]`	Configures the policy-map class trust state, which selects the value that QoS uses as the source of the internal DSCP value (see the “Internal DSCP Values” section on page 32-13). Use the **no** keyword to clear a configured value and return to the default.

When configuring the policy-map class trust state, note the following:

- You can enter the **no trust** command to use the trust state configured on the ingress interface (this is the default).

- With the **cos** keyword, QoS sets the internal DSCP value from received or interface CoS.
- With the **dscp** keyword, QoS uses received DSCP.

Configuring the Policy Map Class DBL State

To configure the policy map class DBL state, perform this task:

Command	Purpose
`Switch(config-pmap-c)# [`**`no`**`] `**`dbl`**	Configures the policy-map class DBL state, which tracks the queue length of traffic flows (see the "Active Queue Management" section on page 32-14). Use the **no** keyword to clear an DBL value and return to the default.

When configuring the policy-map class DBL state, note the following:

- Any class that uses a named aggregate policer must have the same DBL configuration to work.

Configuring Policy-Map Class Policing

These sections describe configuration of policy-map class policing:

- Using a Named Aggregate Policer, page 32-37
- Configuring a Per-Interface Policer, page 32-37

Using a Named Aggregate Policer

To use a named aggregate policer (see the "Creating Named Aggregate Policers" section on page 32-31), perform this task:

Command	Purpose
`Switch(config-pmap-c)# [`**`no`**`] `**`police aggregate`** *`aggregate_name`*	Uses a previously defined aggregate policer. Use the **no** keyword to delete the policer from the policy map class.

Configuring a Per-Interface Policer

To configure a per-interface policer (see the "Policing and Marking" section on page 32-10), perform this task:

Command	Purpose
`Switch(config-pmap-c)# [`**`no`**`] `**`police`** *`rate burst`* `[[`**`conform-action`**` {`**`transmit`**` | `**`drop`**`}] [`**`exceed-action`**` {`**`transmit`**` | `**`drop`**` | `**`policed-dscp-transmit`**`}]]`	Configures a per-interface policer. Use the **no** keyword to delete a policer from the policy map class.

When configuring a per-interface policer, note the following:

- The valid range of values for the *rate* parameter is as follows:
  - Minimum—32 kilobits per second, entered as 32000
  - Maximum—32 gigabits per second, entered as 32000000000

Note See the “Configuration Guidelines” section on page 32-25.

- Rates can be entered in bits-per-second, or you can use the following abbreviations:
  - k to denote 1000 bps
  - m to denote 1000000 bps
  - g to denote 1000000000 bps

Note You can also use a decimal point. For example, a rate of 1,100,000 bps can be entered as 1.1m.

- The valid range of values for the *burst* parameter is as follows:
  - Minimum—1 kilobyte
  - Maximum—512 megabytes
- Bursts can be entered in bytes, or you can use the following abbreviation:
  - k to denote 1000 bytes
  - m to denote 1000000 bytes
  - g to denote 1000000000 bytes

Note You can also use a decimal point. For example, a burst of 1,100,000 bytes can be entered as 1.1m.

- Optionally, you can specify a conform action for matched in-profile traffic as follows:
  - The default conform action is **transmit**.
  - You can enter the **drop** keyword to drop all matched traffic.
- Optionally, for traffic that exceeds the CIR, 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 “Configuring the Policed-DSCP Map” section on page 32-58.
  - For no policing, you can enter the **transmit** keyword to transmit all matched out-of-profile traffic.

This example shows how to create a policy map named *ipp5-policy* that uses the class map named *ipp5*. The class map *ipp5* is configured to rewrite the packet precedence to 6 and to aggregate police the traffic that matches IP precedence value of 5:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# policy-map ipp5-policy
Switch(config-pmap)# class ipp5
Switch(config-pmap-c)# set ip precedence 6
Switch(config-pmap-c)# dbl
Switch(config-pmap-c)# police 2000000000 2000000 conform-action transmit exceed-action
policed-dscp-transmit
Switch(config-pmap-c)# end
```

This example shows how to create a policy map named cs2-policy that uses class map named cs2. The class map cos5 is configured to match on CoS 5 and to aggregate policing the traffic:

```
Switch(config)# class-map cs2
Switch(config-cmap)# match cos 5
Switch(config-cmap)# exit

Switch(config)# policy-map cs2-policy
Switch(config-pmap)# class cs2
police 2000000000 2000000 conform-action transmit exceed-action policed-dscp-transmit

Switch(config)# int g5/1
Switch(config-if)# service-policy input cs2-policy
Switch(config-if)# end

Switch# sh class-map cs2
 Class Map match-all cs2 (id 2)
   Match cos  5

Switch# sh policy-map cs2-policy
  Policy Map cs2-policy
    Class cs2
      police 2000000000 bps 2000000 byte conform-action transmit exceed-action
policed-dscp-transmit Switch#
```

Verifying Policy-Map Configuration

To verify policy-map configuration, perform this task:

	Command	Purpose
Step 1	`Switch(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	`Switch# show policy-map policy_name`	Verifies the configuration.

This example shows how to verify the configuration:

```
Switch# show policy-map ipp5-policy
show policy ipp5-policy
 Policy Map ipp5-policy
  class  ipp5
   set ip precedence 6
   dbl
police 2000000000 2000000 conform-action transmit exceed-action
policed-dscp-transmit
Switch#
```

Attaching a Policy Map to an Interface

To attach a policy map to an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {vlan vlan_ID	{fastethernet	gigabitethernet} slot/interface	Port-channel number}`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] service-policy input policy_map_name`	Attaches a policy map to the input direction of the interface. Use the **no** keyword to detach a policy map from an interface.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show policy-map interface {vlan vlan_ID	{fastethernet	gigabitethernet} slot/interface}`	Verifies the configuration.

Note You cannot enable marking commands on an interface until IP routing is enabled globally. If IP routing is disabled globally and you try to configure the service policy on an interface, the configuration is accepted but it does not take effect. You are prompted with the message: "`Set command will not take effect since CEF is disabled. Please enable IP routing and CEF globally.`" To enable IP routing globally, issue the **ip routing** and **ip cef global** configuration commands. After you do this, the marking commands take effect.

This example shows how to attach the policy map named *pmap1* to Fast Ethernet interface 5/36 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/36
Switch(config-if)# service-policy input pmap1
Switch(config-if)# end
Switch# show policy-map interface fastethernet 5/36
FastEthernet6/1

   service-policy input:p1

     class-map:c1 (match-any)
       238474 packets
       match:access-group 100
        38437 packets
       police:aggr-1
        Conform:383934 bytes Exceed:949888 bytes

    class-map:class-default (match-any)
      0 packets
      match:any
        0 packets
Switch#
```

Configuring CoS Mutation

Note Supervisor Engine 6-E does *not* support this feature.

Service providers providing Layer 2 VPNs carry double tagged or Q in Q traffic with the outer tag representing service provider's VLAN and the inner tag representing the customer's VLAN. Differentiated levels of service can be provided inside the SP network based on the CoS present in the outer tag.

By using CoS Mutation on a dot1q tunnel port, the CoS value on the outer tag of dot1q tunneled packets entering the provider core network can be derived from the CoS of the customer VLAN tag. This allows providers to preserve customer QoS semantics through their network.

CoS mutation is achieved through explicit user configuration to match on specific incoming CoS values and specifying the internal DSCP that is associated for matched packets. This internal DSCP gets converted to CoS through DSCP-CoS mapping during exit from the switch and is the CoS value that gets marked on the outer VLAN tag.

During the process, the CoS in inner tag is preserved and is carried across in the service provider's network.

The following example shows how a policy-map preserves customer VLAN IDs and CoS values throughout the network:

```
Class Map match-any c0
   Match cos  0

 Class Map match-any c1
   Match cos  1

 Class Map match-any c2
   Match cos  2

 Class Map match-any c3
   Match cos  3

 Class Map match-any c4
   Match cos  4

 Class Map match-any c5
   Match cos  5

 Class Map match-any c6
   Match cos  6

 Class Map match-any c7
   Match cos  7

Policy Map cos_mutation
    Class c0
      set dscp default

    Class c1
      set dscp cs1

    Class c2
      set dscp cs2

    Class c3
      set dscp cs3

    Class c4
      set dscp cs4

    Class c5
      set dscp cs5
    Class c6
      set dscp cs6

    Class c7
      set dscp cs7


interface GigabitEthernet5/1
 switchport access vlan 100

 switchport mode dot1q-tunnel
 service-policy input cos_mutation
```

Configuring User Based Rate Limiting

User Based Rate Limiting (UBRL) adopts microflow policing capability to dynamically learn traffic flows and rate limit each unique flow to an individual rate. UBRL is available on Supervisor Engine V-10GE with the built-in NetFlow support. UBRL can be applied to ingress traffic on routed interfaces with source or destination flow masks. It can support up to 85,000 individual flows and 511 rates. UBRL is typically used in environments where a per-user, granular rate-limiting mechanism is required; for example, the per-user outbound traffic rate could differ from the per-user inbound traffic rate.

Note By default, UBRL polices only routed IP traffic. You can use the **ip flow ingress layer2-switched** global command to police switched IP traffic. However, UBRL configuration must remain on a Layer 3 interface. With the UBRL configurations and the **ip flow ingress layer2-switched** global command, you will also be able to police intra-vlan flows. (See the "Configuring Switched/Bridged IP Flows" section on page 46-8). You do not need to enter the **ip flow ingress** command.

A flow is defined as a five-tuple (IP source address, IP destination address, IP head protocol field, Layer 4 source, and destination ports). Flow-based policers enable you to police traffic on a per flow basis. Because flows are dynamic, they require distinguishing values in the class map.

When you specify the **match flow** command with the **source-address** keyword, each flow with a unique source address is treated as a new flow. When you specify the **match flow command** with the **destination-address** keyword, each flow with a unique destination address is treated as a new flow. If the class map used by the policy map has any flow options configured, it is treated as a flow-based policy map. When you specify the **match flow** command with the **ip destination-address ip protocol L4 source-address L4 destination-address** keyword, each flow with unique IP source, destination, protocol, and Layer 4 source and destination address is treated as a new flow.

Note Microflow is only supported on Supervisor Engine V-10GE.

To configure the flow-based class maps and policy maps, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`class-map match-all`** *`class_name`*	Creates a named class map.
Step 2	`Switch(config-cmap)#` **`match flow ip`** `{`**`source-address`** `|` **`ip destination-address ip protocol L4 source-address L4 destination-address`** `|` **`destination-address`**`}`	Specifies the key fields of the flow.

	Command	Purpose
Step 3	`Switch(config-cmap)# end`	Exits class-map configuration mode.
Step 4	`Switch# show class-map class-name`	Verifies the configuration.

Examples

Example 1

This example shows how to create a flow-based class map associated with a source address:

```
Switch(config)# class-map match-all c1
Switch(config-cmap)# match flow ip {source-address [ip destination_address ip protocol L4
source-address L4 destination address]}
Switch(config-cmap)# end
Switch#
Switch# show class-map c1
Class Map match-all c1 (id 2)
  Match flow  ip source-address
```

Example 2

This example shows how to create a flow-based class map associated with a destination address:

```
Switch(config)# class-map match-all c1
Switch(config-cmap)# match flow ip destination-address
Switch(config-cmap)# end
Switch#

Switch# show class-map c1
 Class Map match-all c1 (id 2)
   Match flow  ip destination-address
```

Example 3

Assume there are two active flows on the Fast Ethernet interface 6/1 with source addresses 192.168.10.20 and 192.168.10.21. The following example shows how to maintain each flow to 1 Mbps with an allowed burst value of 9000 bytes:

```
Switch# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map c1
Switch(config-cmap)# match flow ip source-address
Switch(config-cmap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police 1000000 9000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fa6/1
Switch(config-if)# service-policy input p1
Switch(config-if)# end
Switch# write memory

Switch# show policy-map interface
FastEthernet6/1

 Service-policy input: p1
```

```
   Class-map: c1 (match-all)
     15432182 packets
     Match: flow  ip source-address
     police: Per-interface
       Conform: 64995654 bytes Exceed: 2376965424 bytes

   Class-map: class-default (match-any)
     0 packets
     Match: any
       0 packets
```

Example 4

Assume there are two active flows on the Fast Ethernet interface 6/1 with destination addresses of 192.168.20.20 and 192.168.20.21. The following example shows how to maintain each flow to 1 Mbps with an allowed burst value of 9000 bytes:

```
Switch# conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map c1
Switch(config-cmap)# match flow ip destination-address
Switch(config-cmap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police 1000000 9000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fa6/1
Switch(config-if)# service-policy input p1
Switch(config-if)# end
Switch# write memory

Switch# show policy-map interface
 FastEthernet6/1

  Service-policy input: p1

    Class-map: c1 (match-all)
      2965072 packets
      Match: flow  ip destination-address
      police: Per-interface
        Conform: 6105636 bytes Exceed: 476652528 bytes

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
```

Example 5

Assume that there are two active flows on FastEthernet interface 6/1:

```
SrcIp          DstIp          IpProt SrcL4Port DstL4Port
--------------------------------------------------------
192.168.10.10  192.168.20.20  20     6789       81
192.168.10.10  192.168.20.20  20     6789       21
```

With the following configuration, each flow is policed to 1000000 bps with an allowed 9000 burst value.

Note If you use the **match flow ip source-address|destination-address** command, these two flows are consolidated into one flow because they have the same source and destination address.

```
Switch# conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map c1
Switch(config-cmap)# match flow ip source-address ip destination-address ip protocol l4
source-port l4 destination-port
Switch(config-cmap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police 1000000 9000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastEthernet 6/1
Switch(config-if)# service-policy input p1
Switch(config-if)# end
Switch# write memory
Switch# show policy-map interface
FastEthernet6/1

class-map c1
   match flow  ip source-address ip destination-address ip protocol l4 source-port l4
destination-port
!
 policy-map p1
    class c1
       police 1000000 bps 9000 byte conform-action transmit exceed-action drop
!
interface FastEthernet 6/1
  service-policy input p1

Switch# show class-map c1
 Class Map match-all c1 (id 2)
   Match flow  ip source-address ip destination-address ip protocol l4 source-port l4
destination-port

Switch# show policy-map p1
  Policy Map p1
    Class c1
      police 1000000 bps 9000 byte conform-action transmit exceed-action drop

Switch# show policy-map interface
 FastEthernet6/1

  Service-policy input: p1

    Class-map: c1 (match-all)
      15432182 packets
      Match: flow  ip source-address ip destination-address ip protocol l4 source-port l4
destination-port
      police: Per-interface
        Conform: 64995654 bytes Exceed: 2376965424 bytes


    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
```

Configuring Hierarchical Policers

Note Hierarchical policers are only supported on Supervisor Engine V-10GE.

You can tie flow policers with the existing policers to create dual policing rates on an interface. For example, using dual policing, you can limit all incoming traffic rates on a given interface to 50 Mbps and can limit the rate of each flow that is part of this traffic to 2 Mbps.

You can configure hierarchical policers with the **service-policy** policy-map config command. A policy map is termed *flow based* if the class map it uses matches any of the flow-based match criteria (such as **match flow ip source-address**). Each child policy map inherits all the match access-group commands of the parent.

Note You can configure only *flow based* policy maps as child policy maps. A parent policy map cannot be a flow-based policy map. Both the child policy map and parent policy map must have **match-all** in their class-map configuration.

To configure a flow based policy map as a child of an individual or aggregate policer, perform this task:

	Command	Purpose
Step 1	`Switch(config)# policy-map policy_name`	Specifies the individual or aggregate policy-map name.
Step 2	`Switch(config-pmap)# class class_name`	Specifies the class-map name of this policy map.
Step 3	`Switch(config-flow-cache)# service-policy service_policy_name`	Specifies the name of the flow-based policy map.

Note In a hierarchal policer configuration with parent as aggregate policer and child as microflow policer, child microflow policer matched packets report only the packets that are in the profile (that is, match the policing rate). Packets that exceed the policing rate are not reported in the class-map packet match statistics.

This example shows how to create a hierarchical policy map. A policy map with the name *aggregate-policy* has a class map with the name *aggregate-class*. A flow-based policy map with the name *flow-policy* is attached to this policy map as a child policy map.

```
Switch# config terminal
Switch(config)# policy-map aggregate-policy
Switch(config-pmap)# class aggregate-class
Switch(config-pmap-c)# service-policy flow-policy
Switch(config-pmap-c)# end
Switch#
```

In the following example, traffic in the IP address range of 101.237.0.0 to 101.237.255.255 is policed to 50 Mbps. Flows ranging from 101.237.10.0 to 101.237.10.255 are individually policed to a rate of 2 Mbps. This traffic goes through two policers: the aggregate policer and the other flow-based policer.

The following example shows the configuration for this scenario:

```
class-map match-all flow-class
 match flow ip source-address
 match access-group 20
!
class-map match-all aggregate-class
 match access-group 10
!
policy-map flow-policy
 class flow-class
   police 2000000 bps 10000 byte conform-action transmit exceed-action drop
!
policy-map aggregate-policy
 class aggregate-class
   police 50000000 bps 40000 byte conform-action transmit exceed-action drop
  service-policy flow-policy
!
access-list 10 permit 101.237.0.0 0.0.255.255
access-list 20 permit 0.0.10.0 255.255.0.255
```

The following example shows how to verify the configuration:

```
Switch# show policy-map flow-policy
 Policy Map flow-policy
   Class flow-class
     police 2000000 bps 10000 byte conform-action transmit exceed-action drop
Switch# show policy-map aggregate-policy
 Policy Map aggregate-policy
   Class aggregate-class
     police 50000000 bps 40000 byte conform-action transmit exceed-action drop
     service-policy flow-policy

Switch# show policy-map interface
FastEthernet6/1
 Service-policy input: aggregate-policy

   Class-map: aggregate-class (match-all)
     132537 packets
     Match: access-group 10
     police: Per-interface
       Conform: 3627000 bytes Exceed: 0 bytes

     Service-policy : flow-policy

       Class-map: flow-class (match-all)
         8867 packets
         Match: access-group 20
         Match: flow  ip source-address
         police: Per-interface
       Conform: 1649262 bytes Exceed: 59601096 bytes

       Class-map: class-default (match-any)
         0 packets
         Match: any            0 packets

   Class-map: class-default (match-any)
     5 packets
     Match: any        5 packets
```

Enabling Per-Port Per-VLAN QoS

The per-port per-VLAN QoS feature enables you to specify different QoS configurations on different VLANs on a given interface. Typically, you use this feature on trunk or voice VLANs (Cisco IP Phone) ports, as they belong to multiple VLANs.

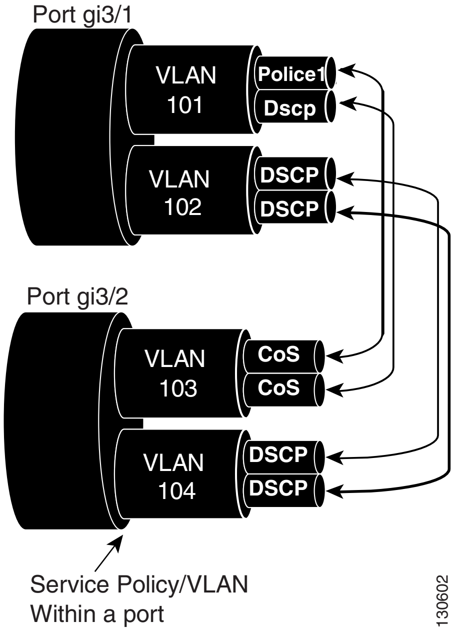
To configure per-port per-VLAN QoS, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet	gigabitethernet	tengigabitethernet} slot/interface	Port-channel number`	Selects the interface to configure.
Step 2	`Switch(config-if)# vlan-range vlan_range`	Specifies the VLANs involved.
Step 3	`Switch(config-if-vlan-range)# service-policy {input	output} policy-map`	Specifies the policy-map and direction.
Step 4	`Switch(config-if-vlan-range)# exit`	Exits class-map configuration mode.
Step 5	`Switch(config-if)# end`	Exits configuration interface mode.
Step 6	`Switch# show policy-map interface interface_name`	Verifies the configuration.

Example 1

Figure 32-6 displays a sample topology for configuring PVQoS. The trunk port gi3/1 is comprised of multiple VLANs (101 and 102). Within a port, you can create your own service policy per VLAN. This policy, performed in hardware, might consist of ingress and egress Policing, trusting DSCP, or giving precedence to voice packet over data.

Figure 32-6 Per-Port Per-VLAN Topology



The following configuration file shows how to perform ingress and egress policing per VLAN using the policy-map P31_QOS applied to port Gigabit Ethernet 3/1:

```
ip access-list 101 permit ip host 1.2.2.2 any
ip access-list 103 permit ip any any
Class-map match-all RT

match ip access-group 101
Class-map Match all PD

match ip access-group 103
Policy-map P31_QoS

Class RT


Police 200m 16k conform transmit exceed drop

Class PD


Police 100m 16k conform transmit exceed drop

Interface Gigabit 3/1
Switchport
Switchport trunk encapsulation dot1q
Switchport trunk allowed vlan 101-102
    Vlan range 101
        Service-policy input P31_QoS
        Service-policy output P31_QoS
    Vlan range 102
        Service-policy input P32_QoS
        Service-policy output P32_QoS
```

Example 2

Let us assume that interface Gigabit Ethernet 6/1 is a trunk port and belongs to VLANs 20, 300-301, and 400. The following example shows how to apply policy-map p1 for traffic in VLANs 20 and 400 and policy map p2 to traffic in VLANs 300 through 301:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet 6/1
Switch(config-if)# vlan-range 20,400
Switch(config-if-vlan-range)# service-policy input p1
Switch(config-if-vlan-range)# exit
Switch(config-if)# vlan-range 300-301
Switch(config-if-vlan-range)# service-policy output p2
Switch(config-if-vlan-range)# end
Switch#
```

Example 3

The following command shows how to display policy-map statistics on VLAN 20 configured on Gigabit Ethernet interface 6/1:

```
Switch# show policy-map interface gigabitethernet 6/1 vlan 20
 GigabitEthernet6/1 vlan 20

  Service-policy input: p1
```

```
    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes
```

Example 4

The following command shows how to display policy-map statistics on all VLANs configured on Gigabit Ethernet interface 6/1:

```
Switch# show policy-map interface gigabitethernet 6/1
 GigabitEthernet6/1 vlan 20

  Service-policy input: p1

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes

 GigabitEthernet6/1 vlan 300

  Service-policy output: p2

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes

 GigabitEthernet6/1 vlan 301

  Service-policy output: p2

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes

 GigabitEthernet6/1 vlan 400

  Service-policy input: p1

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes
```

Enabling or Disabling QoS on an Interface

The **qos** interface command reenables any previously configured QoS features. The **qos** interface command does not affect the interface queueing configuration.

To enable or disable QoS features for traffic from an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# **interface** {**vlan** *vlan_ID* | {**fastethernet** | **gigabitethernet**} *slot/interface* | **Port-channel** *number*}	Selects the interface to configure.
Step 2	Switch(config-if)# [**no**] **qos**	Enables QoS on the interface. Use the **no** keyword to disable QoS on an interface.
Step 3	Switch(config-if)# **end**	Exits configuration mode.
Step 4	Switch# **show qos interface**	Verifies the configuration.

This example shows how to disable QoS on interface VLAN 5:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface vlan 5
Switch(config-if)# no qos
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos | begin QoS is disabled
  QoS is disabled on the following interfaces:
    Vl5
<...Output Truncated...>
Switch#
```

Configuring VLAN-Based QoS on Layer 2 Interfaces

By default, QoS uses policy maps attached to physical interfaces. For Layer 2 interfaces, you can configure QoS to use policy maps attached to a VLAN. (See the "Attaching a Policy Map to an Interface" section on page 32-40.)

To configure VLAN-based QoS on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# **interface** {**fastethernet** | **gigabitethernet**} *slot/interface* | **Port-channel** *number*	Selects the interface to configure.
Step 2	Switch(config-if)# [**no**] **qos vlan-based**	Configures VLAN-based QoS on a Layer 2 interface. Use the **no** keyword to disable VLAN-based QoS on an interface.
Step 3	Switch(config-if)# **end**	Exits configuration mode.
Step 4	Switch# **show qos**	Verifies the configuration.

Note If no input QoS policy is attached to a Layer 2 interface, then the input QoS policy attached to the VLAN (on which the packet is received), if any, is used even if the port is not configured as VLAN-based. If you do not want this default, attach a placeholder input QoS policy to the Layer 2 interface. Similarly, if no output QoS policy is attached to a Layer 2 interface, then the output QoS policy attached to the VLAN (on which the packet is transmitted), if any, is used even if the port is not configured as VLAN-based. If you do not want this default, attach a placeholder output QoS policy to the layer 2 interface.

This example shows how to configure VLAN-based QoS on Fast Ethernet interface 5/42:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/42
Switch(config-if)# qos vlan-based
Switch(config-if)# end
```

This example shows how to verify the configuration:

```
Switch# show qos | begin QoS is vlan-based
QoS is vlan-based on the following interfaces:
    Fa5/42
Switch#
```

Note When a layer 2 interface is configured with VLAN-based QoS, and if a packet is received on the port for a VLAN on which there is no QoS policy, then the QoS policy attached to the port, if any is used. This applies for both Input and Output QoS policies.

Configuring the Trust State of Interfaces

This command configures the trust state of interfaces. By default, all interfaces are untrusted.

To configure the trust state of an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {vlan vlan_ID	{fastethernet	gigabitethernet} slot/interface| Port-channel number}`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] qos trust [dscp	cos]`	Configures the trust state of an interface. Use the **no** keyword to clear a configured value and return to the default.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show qos`	Verifies the configuration.

When configuring the trust state of an interface, note the following:

- You can use the **no qos trust** command to set the interface state to untrusted.
- For traffic received on an ingress interface configured to *trust CoS* using the **qos trust cos** command, the transmit CoS is always the incoming packet CoS (or the ingress interface default CoS if the packet is received untagged).
- When the interface trust state is not configured to *trust dscp* using the **qos trust dscp** command, the security and QoS ACL classification always use the interface DSCP and not the incoming packet DSCP.

- Starting with Cisco IOS Release 12.2(31)SG, the Supervisor Engine V-10GE enables you to classify a packet based on the packet's IP DSCP value irrespective of the port trust state. Packet transmit queuing isn't impacted by this behavior. For information on transmit queues, refer to the "Configuring Transmit Queues" section on page 32-54".

This example shows how to configure Gigabit Ethernet interface 1/1 with the **trust cos** keywords:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# qos trust cos
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos interface gigabitethernet 1/1 | include trust
  Trust state: trust COS
Switch#
```

Configuring the CoS Value for an Interface

QoS assigns the CoS value specified with this command to untagged frames from ingress interfaces configured as trusted and to all frames from ingress interfaces configured as untrusted.

To configure the CoS value for an ingress interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface | Port-channel number}`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] qos cos default_cos`	Configures the ingress interface CoS value. Use the **no** keyword to clear a configured value and return to the default.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show qos interface {fastethernet | gigabitethernet} slot/interface`	Verifies the configuration.

This example shows how to configure the CoS 5 as the default on Fast Ethernet interface 5/24:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/24
Switch(config-if)# qos cos 5
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos interface fastethernet 5/24 | include Default COS
  Default COS is 5
Switch#
```

Configuring DSCP Values for an Interface

QoS assigns the DSCP value specified with this command to non IPv4 frames received on interfaces configured to trust DSCP and to all frames received on interfaces configured as untrusted.

To configure the DSCP value for an ingress interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`interface`** `{`**`fastethernet`** `|` **`gigabitethernet`**`}` *`slot/interface`* `|` **`Port-channel`** *`number`*	Selects the interface to configure.
Step 2	`Switch(config-if)# [`**`no`**`]` **`qos dscp`** *`default_dscp`*	Configures the ingress interface DSCP value. Use the **no** keyword to clear a configured value and return to the default.
Step 3	`Switch(config-if)#` **`end`**	Exits configuration mode.
Step 4	`Switch#` **`show qos interface`** `{`**`fastethernet`** `|` **`gigabitethernet`**`}` *`slot/interface`*	Verifies the configuration.

This example shows how to configure the DSCP 5 as the default on Fast Ethernet interface 5/24:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/24
Switch(config-if)# qos dscp 5
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos interface fastethernet 6/1
QoS is enabled globally
Port QoS is enabled
  Port Trust State:CoS
  Default DSCP:0 Default CoS:0

  Tx-Queue   Bandwidth   ShapeRate   Priority   QueueSize
               (bps)       (bps)                (packets)
    1        31250000    disabled    N/A        240
    2        31250000    disabled    N/A        240
    3        31250000    disabled    normal     240
    4        31250000    disabled    N/A        240
Switch#
```

Configuring Transmit Queues

The following sections describe how to configure transmit queues:

- Mapping DSCP Values to Specific Transmit Queues, page 32-55
- Allocating Bandwidth Among Transmit Queues, page 32-56

- Configuring Traffic Shaping of Transmit Queues, page 32-56
- Configuring a High Priority Transmit Queue, page 32-57

Depending on the complexity of your network and your QoS solution, you might need to perform all of the procedures in the following sections. However, you will first need to answer the following questions:

- Which packets are assigned (by DSCP value) to each queue?
- What is the size of a transmit queue relative to other queues for a given port?
- How much of the available bandwidth is allotted to each queue?
- What is the maximum rate and burst of traffic that can be transmitted out of each transmit queue?

Mapping DSCP Values to Specific Transmit Queues

To map the DSCP values to a transmit queue, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [`**`no`**`]` **`qos map dscp`** ***`dscp-values`*** **`to tx-queue`** ***`queue-id`***	Maps the DSCP values to the transit queue. *dscp-list* can contain up 8 DSCP values. The *queue-id* can range from 1 to 4. Use the **no qos map dscp to tx-queue** command to clear the DSCP values from the transit queue.
Step 2	`Switch(config)#` **`end`**	Exits configuration mode.
Step 3	`Switch#` **`show qos maps dscp tx-queues`**	Verifies the configuration.

This example shows how to map DSCP values to transit queue 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos map dscp 50 to tx-queue 2
Switch(config)# end
Switch#
```

This example shows how to verify the configuration.

```
Switch# show qos maps dscp tx-queue
DSCP-TxQueue Mapping Table (dscp = d1d2)
d1 :d2  0  1  2  3  4  5  6  7  8  9
-------------------------------------
0 :   02 02 02 01 01 01 01 01 01 01
1 :   01 01 01 01 01 01 02 02 02 02
2 :   02 02 02 02 02 02 02 02 02 02
3 :   02 02 03 03 03 03 03 03 03 03
4 :   03 03 03 03 03 03 03 03 04 04
5 :   04 04 04 04 04 04 04 04 04 04
6 :   04 04 04 04
Switch#
```

Allocating Bandwidth Among Transmit Queues

To configure the transmit queue bandwidth, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface gigabitethernet slot/interface`	Selects the interface to configure.
Step 2	`Switch(config-if)# tx-queue queue_id`	Selects the transmit queue to configure.
Step 3	`Switch(config-if-tx-queue)# [no] [bandwidth rate | percent percent]`	Sets the bandwidth rate for the transmit queue. Use the **no** keyword to reset the transmit queue bandwidth ratios to the default values.
Step 4	`Switch(config-if-tx-queue)# end`	Exits configuration mode.
Step 5	`Switch# show qos interface`	Verifies the configuration.

The bandwidth rate varies with the interface.

Bandwidth can only be configured on these interfaces:

- Uplink ports on Supervisor Engine III (WS-X4014)
- Ports on the WS-X4306-GB module
- The 2 1000BASE-X ports on the WS-X4232-GB-RJ module
- The first 2 ports on the WS-X4418-GB module
- The two 1000BASE-X ports on the WS-X4412-2GB-TX module

This example shows how to configure the bandwidth of 1 Mbps on transmit queue 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# tx-queue 2
Switch(config-if-tx-queue)#bandwidth 1000000
Switch(config-if-tx-queue)# end
Switch#
```

Configuring Traffic Shaping of Transmit Queues

To guarantee that packets transmitted from a transmit queue do not exceed a specified maximum rate, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface`	Selects the interface to configure.
Step 2	`Switch(config-if)# tx-queue queue_id`	Selects the transmit queue to configure.
Step 3	`Switch(config-if-tx-queue)# [no] [shape rate | percent percent]`	Sets the transmit rate for the transmit queue. Use the **no** keyword to clear the transmit queue maximum rate.
Step 4	`Switch(config-if-tx-queue)# end`	Exits configuration mode.
Step 5	`Switch# show qos interface`	Verifies the configuration.

This example shows how to configure the shape rate to 1 Mbps on transmit queue 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if-tx-queue)# tx-queue 2
Switch(config-if-tx-queue)# shape 1000000
Switch(config-if-tx-queue)# end
Switch#
```

Configuring a High Priority Transmit Queue

To configure transmit queue 3 at a higher priority, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface`	Selects the interface to configure.
Step 2	`Switch(config-if)# tx-queue 3`	Selects transmit queue 3 to configure.
Step 3	`Switch(config-if)# [no] priority high`	Sets the transmit queue to high priority. Use the **no** keyword to clear the transmit queue priority.
Step 4	`Switch(config-if)# end`	Exits configuration mode.
Step 5	`Switch# show qos interface`	Verifies the configuration.

This example shows how to configure transmit queue 3 to high priority.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if-tx-queue)# tx-queue 3
Switch(config-if-tx-queue)# priority high
Switch(config-if)# end
Switch#
```

Configuring DSCP Maps

The following sections describes how to configure the DSCP maps. It contains this configuration information:

- Configuring the CoS-to-DSCP Map, page 32-57
- Configuring the Policed-DSCP Map, page 32-58
- Configuring the DSCP-to-CoS Map, page 32-59

All the maps are globally defined and are applied to all ports.

Configuring the CoS-to-DSCP Map

You use the CoS-to-DSCP map to map CoS values in incoming packets to a DSCP value that QoS uses internally to represent the priority of the traffic.

Table 32-4 shows the default CoS-to-DSCP map.

Table 32-4 Default CoS-to-DSCP Map

CoS value	0	1	2	3	4	5	6	7
DSCP value	0	8	16	24	32	40	48	56

If these values are not appropriate for your network, you need to modify them.

To modify the CoS-to-DSCP map, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# qos map cos cos1 ... cos8 to dscp dscp`	Modifies the CoS-to-DSCP map. For *cos1...cos8*, you can enter up to 8 CoS; valid values range from 0 to 7. Separate each CoS value with a space. The *dscp* range is 0 to 63.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show qos maps cos-dscp`	Verifies your entries.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to configure the ingress CoS-to-DSCP mapping for CoS 0:

```
Switch# configure terminal
Switch(config)# qos map cos 0 to dscp 20
Switch(config)# end
Switch# show qos maps cos dscp

CoS-DSCP Mapping Table:
CoS:  0   1  2  3  4  5  6  7
--------------------------------
DSCP: 20  8 16 24 32 40 48 56
Switch(config)#
```

Note To return to the default map, use the **no qos cos to dscp** global configuration command.

This example shows how to clear the entire CoS-to-DSCP mapping table:

```
Switch(config)# no qos map cos to dscp
Switch(config)#
```

Configuring the Policed-DSCP Map

You use the policed-DSCP map to mark down a DSCP value to a new value as the result of a policing and marking action.

The default policed-DSCP map is a null map, which maps an incoming DSCP value to the same DSCP value.

To modify the CoS-to-DSCP map, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`qos map dscp policed`** *`dscp-list`* **`to dscp`** *`mark-down-dscp`*	Modifies the policed-DSCP map. • For *dscp-list*, enter up to 8 DSCP values separated by spaces. Then enter the **to** keyword. • For *mark-down-dscp*, enter the corresponding policed (marked down) DSCP value.
Step 3	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Switch#` **`show qos maps dscp policed`**	Verifies your entries.
Step 5	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To return to the default map, use the **no qos dscp policed** global configuration command.

This example shows how to map DSCP 50 to 57 to a marked-down DSCP value of 0:

```
Switch# configure terminal
Switch(config)# qos map dscp policed 50 51 52 53 54 55 56 57 to dscp 0
Switch(config)# end
Switch# show qos maps dscp policed
Policed-dscp map:
     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 :    00 00 00 00 00 00 00 00 58 59
      6 :    60 61 62 63
```

Note In the previous policed-DSCP map, the marked-down DSCP values are shown in the body of the matrix. The d1 column specifies the most-significant digit of the original DSCP; the d2 row specifies the least-significant digit of the original DSCP. The intersection of the d1 and d2 values provides the marked-down value. For example, an original DSCP value of 53 corresponds to a marked-down DSCP value of 0.

Configuring the DSCP-to-CoS Map

You use the DSCP-to-CoS map to generate a CoS value.

Table 32-5 shows the default DSCP-to-CoS map.

Table 32-5 Default DSCP-to-CoS Map

DSCP value	0–7	8–15	16–23	24–31	32–39	40–47	48–55	56–63
CoS value	0	1	2	3	4	5	6	7

If the values above are not appropriate for your network, you need to modify them.

To modify the DSCP-to-CoS map, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)# [`**`no`**`]` **`qos map dscp`** *`dscp-list`* **`to cos`** *`cos`*	Modifies the DSCP-to-CoS map. • For *dscp-list*, enter up to 8 DSCP values separated by spaces. Then enter the **to** keyword. • For *cos*, enter only one CoS value to which the DSCP values correspond. The DSCP range is 0 to 63; the CoS range is 0 to 7. To return to the default map, use the **no qos dscp to cos** global configuration command.
Step 3	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Switch#` **`show qos maps dscp to cos`**	Verifies your entries.
Step 5	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

This example shows how to map DSCP values 0, 8, 16, 24, 32, 40, 48, and 50 to CoS value 0 and to display the map:

```
Switch# configure terminal
Switch(config)# qos map dscp 0 8 16 24 32 40 48 50 to cos 0
Switch(config)# end
Switch# show qos maps dscp cos
Dscp-cos map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :    00 00 00 00 00 00 00 00 00 01
      1 :    01 01 01 01 01 01 00 02 02 02
      2 :    02 02 02 02 00 03 03 03 03 03
      3 :    03 03 00 04 04 04 04 04 04 04
      4 :    00 05 05 05 05 05 05 05 00 06
      5 :    00 06 06 06 06 06 07 07 07 07
      6 :    07 07 07 07
```

Note In the previous DSCP-to-CoS map, the CoS values are shown in the body of the matrix. The d1 column specifies the most-significant digit of the DSCP; the d2 row specifies the least-significant digit of the DSCP. The intersection of the d1 and d2 values provides the CoS value. For example, in the DSCP-to-CoS map, a DSCP value of 08 corresponds to a CoS value of 0.

Enabling Layer 2 Control Packet QoS

Note Layer 2 Control Packet QoS is *not* supported on Supervisor Engine 6-E.

This feature solves the problem of high CPU utilization due to the ingress of a large amount of control packets. It does this by invalidating the QoS static entries corresponding to the protocol that you want to control (installed in the QoS CAM).

With this solution, hardware applies the actions corresponding to any user defined service policies that match the Layer 2 control traffic. You can deploy this mode of control through the CLI because the default mode will be the existing one.

You should configure policies to match on the required Layer 2 packets and police them to the desired level. Layer 2 control packets are essentially identified by a destination MAC address. When this feature is enabled on that packet type, if MACLs matching the desired control packets and the corresponding class-maps matching these MACLs are not already present, they will be auto-generated.

You are only required to use these class-maps in the policy-maps you create to police the control packets. You can then apply the policy-map on a per port, per vlan, or per-port-per-vlan just like any other policy-maps.

To enable Layer 2 Control Packet QoS, perform the following task:

	Command	Purpose
Step 1	`Switch# interface t`	Enters configuration mode.
Step 2	`Switch(config)# qos control-packets [bpdu-range | cdp-vtp | sstp]`	Enables Layer 2 Control Policing. You can specify the packet type that you will enable the feature on. The default is to select all packet types.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show run | inc qos control-packets`	Verifies the configuration.

The following table lists the types of packets impacted by this feature.

Table 32-6 Packet Type and Actionable Address Range

Type of packet the feature is enabled on	Range of address it acts on
BPDU-range	0180.C200.0000 BPDU 0180.C200.0002 OAM, LACP 0180.C200.0003 Eapol
CDP-VTP	0100.0CCC.CCCC
SSTP	0100.0CCC.CCCD

The following example illustrates how to configure Layer 2Control Packet QoS on cdp-vtp packets:

```
Switch# config terminal
Switch(config)# qos control-packets cdp-vtp
Switch(config)# end
Switch# show run | inc qos control-packets
qos control-packets cdp-vtp
```

When the feature is enabled on all the packet types, the **qos control-packets** string appears in the output of the **show running** command:

```
Switch# show running | inc qos control-packets
qos control-packets
```

Now, if you disable the feature for sstp packets, you'll observe the following output:

```
Switch# show running | inc qos control-packets
qos control-packets bpdu-range
qos control-packets cdp-vtp
```

You can observe the status of the MACL and of the user-configured policies that will likely capture and drop/police the desired control packets with the **show running** and common relative commands.

To disable the feature, issue the **no qos control-packets [bpdu-range | cdp-vtp | sstp]** command. For example, to disable the feature on CDP-VTP packets, issue the **no qos control-packets cdp-vtp** command.

Note When you un-configure this feature for a specified protocol-type, the user- configured policies dealing with that protocol-type are immediately rendered ineffective. To save TCAM resources, you can remove the policies as well as the MACLs and class-maps (auto generated or user defined).

The following table displays the auto-generated MACLs and class-maps that are created when the feature is enabled on the corresponding packet type.

Table 32-7 Packet Types and Auto-Generated MACL/Class-Maps

Packet Type	Auto-Generated MACL/Class-Map
BPDU-range	mac access-list extended system-control-packet-bpdu-range permit any 0180.c200.0000 0000.0000.000c class-map match-any system-control-packet-bpdu-range match access-group name system-control-packet-bpdu-range
SSTP	mac access-list extended system-control-packet-sstp permit any host 0100.0ccc.cccd class-map match-any system-control-packet-sstp match access-group name system-control-packet-sstp
CDP-VTP	mac access-list extended system-control-packet-cdp-vtp permit any host 0100.0ccc.cccc class-map match-any system-control-packet-cdp-vtp match access-group name system-control-packet-cdp-vtp

The following example illustrates how to apply a MACL and policer configuration to BPDU-range packets:

- Enable the feature on the bpdu-range:

```
qos control-packets bpdu-range
```

- Create the corresponding MACL / class-maps (happens automatically):

```
mac access-list extended system-control-packet-bpdu-range
    permit any 0180.C200.0000 0000.0000.000C
```

```
class-map match-any system-control-packet-bpdu-range
    match access-group name system-control-packet-bpdu-range
```

- Create the policy-map and attach it to the desired interface / VLAN:

```
policy-map police-bpdu
  class system-control-packet-bpdu-range
    police 32000 bps 1000 byte conform-action transmit exceed-action drop

interface GigabitEthernet 1/1
      switchport trunk encapsulation dot1q
      switchport mode trunk
      vlan-range 100
          service-policy input police_bpdu
```

You should not modify the class-maps and MACLs that are generated by the system. Doing so might lead to unexpected behavior when the switch reloads or when the running configuration is updated from a file.

If you need to refine or modify these system generated class-maps or MACLs, you should create user-defined class-maps and MACLs. You can then use the newly created user-defined MACLs / class-maps to accomplish the desired policing.

Note The only restriction is that user-defined class-map names must begin with the prefix **system-control-packet-**. If the class-map does not begin with **system-control-packet-**, the configured QoS action may not be taken on certain supervisor engines.

Note There are no restrictions on the names you can use for user-defined MACLs.

For example, here are valid user defined class-maps names to police the control packets:

```
system-control-packet-bpdu1
system-control-packet-control-packet
system-control-packet-bla
```

Creating the user-defined MACLs / class-maps (as the example above) might be useful, for example, if you plan to define different class-maps for EAPOL, OAM, or BPDUs packets because the auto generated class-map system-control-packet-bpdu-range will match all of them:

```
mac access-list extended system-control-packet-bpdu
   permit any 0180.c200.0000
class-map match-any system-control-packet-bpdu
   match access-group name system-control-packet-bpdu

mac access-list extended system-control-packet-eapol
   permit any 0180.c200.0003
class-map match-any system-control-packet-eapol
   match access-group name system-control-packet-eapol

mac access-list extended system-control-packet-oam
   permit any 0180.c200.0002
class-map match-any system-control-packet-oam
   match access-group name system-control-packet-oam
```

Subsequently, you could use these class-maps to define different policers for each, instead of applying a common policer to the system-control-packet-bpdu-range.

Usage Guidelines

When this feature is enabled, you need to ensure that the existing policies applied to ports and VLANs are such that the Layer 2 control packets that you want to control are not inadvertently subjected to undesired QoS actions, and that the functionality of this feature is not impacted by other policies configured on the switch.

Before enabling QoS on the above mentioned control packets, you must examine and edit your new and existing policies to ensure that the classifiers in the policy-map matching the selected control packets are defined and configured in the correct sequence. To prevent undesirable results from actions of another classifier that may appear later in the same policy-map, you should place the classifiers matching control packets at the beginning of the policy map.

For actions associated with the class class-default, the behavior will depend on the type of supervisor engine:

- Supervisor Engine V-10GE with the built-in NetFlow support

  The actions associated with the class-default will never be applied on unmatched control packets, and a default permit action will be applied in case no control-packet class-maps are catching the control-packets before. Only the actions associated to policers that use the class-maps beginning with system-control-packet- will be applied on control packets.

- All other supervisor engines

  Actions associated with class-default are applied on unmatched control packets.

Note On Supervisor Engine V-10GE with the built-in NetFlow support, no micro-flow stats will be available for these types of packets.

Note When the feature is enabled on BPDU-range, you can police EAPOL packets only after that initial 802.1X authentication phase has completed.

Note When port security is enabled on a port that is in forwarding spanning tree state, Layer 2 control packets cannot be policed on that port.

Feature Interaction

After applying user-configured policies on each single flow, you might configure a CoPP policy *on top of* Layer 2 Control Packet QoS to rate limit the aggregate flow coming to the CPU. If so, CoPP essentially provides another level of protection for CPU by further rate-limiting on the egress side the packets already filtered in ingress on a per port per VLAN basis by the user defined policies. CoPP becomes the second level of defense while user-defined policies applied on ports and VLANs become the first level of defense.

For example, if you configure a policy-map matching and policing the BPDU-range traffic coming from interface Gigabit 1/1, VLAN 1 as follows:

```
policy-map police_bpdu_1
      class system-control-packet-bpdu-range
  police 32000 bps 1000 byte conform-action transmit exceed-action drop

   interface GigabitEthernet1/1
      switchport trunk encapsulation dot1q
      switchport mode trunk
      vlan-range 1
      service-policy input police_bpdu_1
```

and configure a second one on interface Gigabit 1/2 VLAN 2, matching and policing BPUD-range packets as follows:

```
policy-map police_bpdu_2
      class system-control-packet-bpdu-range
  police 34000 bps 1000 byte conform-action transmit exceed-action drop

   interface GigabitEthernet1/2
      switchport trunk encapsulation dot1q
      switchport mode trunk
      vlan-range 2
      service-policy input police_bpdu_2
```

the CoPP would be configured as follows:

```
policy-map system-cpp-policy
      class system-cpp-bpdu-range
  police 50000 bps 1000 byte conform-action transmit exceed-action drop
```

Note the following:

- On interface 1/1, VLAN 1, the BPDU-range packets will be policed accordingly to police_bpdu_1 at the rate of 32000 bit per second.
- On interface 1/2, VLAN 2, the BPDU-range packets will be policed accordingly to police_bpdu_2 at the rate of 34000 bit per second.
- The aggregate flow will be policed from CoPP at the CPU port at the rate of 50000 bit per second.

It is also possible to use named-aggregate policers applied to a group of ports or of VLANs to reduce the consumption of policer resources.

Note When port security is enabled on a port that is in forwarding spanning tree state, Layer 2 control packets cannot be policed on that port.

Configuring Auto-QoS on Supervisor Engine 6-E

Unlike auto-QoS on Supervisor Engines II-Plus to V-10GE, auto-QoS on Supervisor Engine 6-E employs the MQC model. This means that instead of using certain global configurations (like qos and qos dbl), auto-QoS applied to any interface on a switch with Supervisor Engine 6-E configures several global class-maps and policy-maps.

The class-maps are as follows:

```
class-map match-all AutoQos-VoIP-Control-Dscp26
  match  dscp af31
class-map match-all AutoQos-VoIP-Control-Dscp24
  match  dscp cs3
class-map match-all AutoQos-VoIP-Bearer-Cos
  match cos  5
class-map match-all AutoQos-VoIP-Control-QosGroup24
  match qos-group 24
class-map match-all AutoQos-VoIP-Control-QosGroup26
  match qos-group 26
class-map match-all AutoQos-VoIP-Bearer-QosGroup
  match qos-group 46
```

```
class-map match-all AutoQos-VoIP-Bearer-Dscp
  match  dscp ef
class-map match-all AutoQos-VoIP-Control-Cos
  match cos  3
```

The class maps are intended to identify control and data (bearer) voice traffic for either an Layer 2 or Layer 3 interface.

The policy maps are as follows:

```
policy-map AutoQos-VoIP-Input-Dscp-Policy
  class AutoQos-VoIP-Bearer-Dscp
   set qos-group 46
  class AutoQos-VoIP-Control-Dscp26
   set qos-group 26
  class AutoQos-VoIP-Control-Dscp24
   set qos-group 24
policy-map AutoQos-VoIP-Input-Cos-Policy
  class AutoQos-VoIP-Bearer-Cos
   set qos-group 46
  class AutoQos-VoIP-Control-Cos
   set qos-group 24
policy-map AutoQos-VoIP-Output-Policy
  class AutoQos-VoIP-Bearer-QosGroup
   set dscp ef
   set cos 5
    priority
   police cir percent 33
  class AutoQos-VoIP-Control-QosGroup26
   set dscp af31
   set cos 3
    bandwidth remaining percent 5
  class AutoQos-VoIP-Control-QosGroup24
   set dscp cs3
   set cos 3
    bandwidth remaining percent 5
  class class-default
    dbl
```

The three policy maps are defined as follows:

- policy-map AutoQos-VoIP-Input-Dscp-Policy

  This policy map is applied as an input service policy on an Layer 3 interface (such as an uplink connection to a neighboring switch) when auto-QoS is configured on the port.

- policy-map AutoQos-VoIP-Input-Cos-Policy

  This policy map is applied as an input service policy on an Layer 2 interface that could be either an uplink connection or a port hooked to a Cisco IP Phone.

- policy-map AutoQos-VoIP-Output-Policy

  This policy map is applied as an output policy for any port on which auto-QoS is configured, establishing policy governing egress traffic on the port based on whether it is voice data or control traffic.

The purpose of the input policy maps is to identify voice data or control traffic and mark it as such as it traverses the switch. The output policy map matches the packets on the marking occurring on ingress and then applies egress parameters such as bandwidth, policing and/or priority queuing.

The invocation of auto-QoS on a switch employing Supervisor Engine 6-E uses the same config commands used on Supervisor Engines II-Plus to V-10GE.

For switch-to-switch connections, the **[no] auto qos voice trust** command is used to apply an input and output service policy on the interface:

```
service-policy input AutoQos-VoIP-Input-Cos-Policy
```

OR

```
service-policy input AutoQos-VoIP-Input-Dscp-Policy
```

AND

```
service-policy output AutoQos-VoIP-Output-Policy
```

The selection of the input policy depends on whether the port is Layer 2 or Layer 3. For Layer 2, the policy trusts the Cos setting in the received packets. For Layer 3 ports, it relies on the DSCP value contained in the packets.

For phone connected ports, the **[no] auto qos voice cisco-phone** command is used to apply the following service policy to the port:

```
qos trust device cisco-phone

service-policy input AutoQos-VoIP-Input-Cos-Policy
```

AND

```
service-policy output AutoQos-VoIP-Output-Policy
```

It establishes a trusted boundary that recognizes Cisco IP Phones and trusts the Cos setting of the packets from the phone. If a Cisco IP Phone is not detected, the Cos field is ignored and the packets are not classified as voice traffic. Upon detecting a Cisco phone, the ingress packets are marked based on the Cos value in the packets. This marking is used on egress for proper traffic classification and handling.

Configuring QoS on Supervisor Engine 6-E

Topics include:

- MQC-based QoS Configuration, page 32-68
- Overview, page 32-68
- Platform-supported Classification Criteria and QoS Features, page 32-69
- Platform Hardware Capabilities, page 32-70
- Prerequisites for Applying a QoS Service Policy, page 32-70
- Restrictions for Applying a QoS Service Policy, page 32-70
- Classification, page 32-71
- Policing, page 32-71
- Marking Network Traffic, page 32-72
- Shaping, Sharing(Bandwidth), Priority Queuing and DBL, page 32-79

MQC-based QoS Configuration

Starting with Cisco IOS Release 12.2(40)SG, a Catalyst 4500 Series Switch using Supervisor Engine 6-E employs the MQC model of QoS. To apply QoS, you use the Modular QoS Command-Line Interface (MQC), which is a CLI structure that allows you to complete the following tasks:

- Specify the matching criteria used to define a traffic class.
- Create a traffic policy (policy map). The traffic policy defines the QoS policy actions to be taken for each traffic class.
- Apply the policy actions specified in the policy map to an interface, VLAN, or port and VLAN.

For more information about the MQC, see the "Modular Quality of Service Command-Line Interface" section of the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3*.

Overview

The Supervisor Engine 6-E supports the best effort and DiffServ types of QoS deployment (RFCs 2597, 2598, 2474, 2475 define the DiffServ standards). A high level Supervisor Engine 6-E QoS model is as follows:

Step 1 The incoming packet is classified (based on different packet fields, receive port and/or VLAN) to belong to a traffic class.

Step 2 Depending on the traffic class, the packet is rate-limited/policed and its priority is optionally *marked* (typically at the edge of the network) so that lower priority packets are dropped or marked with lower priority in the packet fields (DSCP and CoS).

Step 3 After the packet has been marked, it is *looked up* for forwarding. This action obtains the transmit port and VLAN to transmit the packet.

Step 4 The packet is classified in the output direction based on the transmit port and/or VLAN. The classification takes into account any marking of the packet by input QoS.

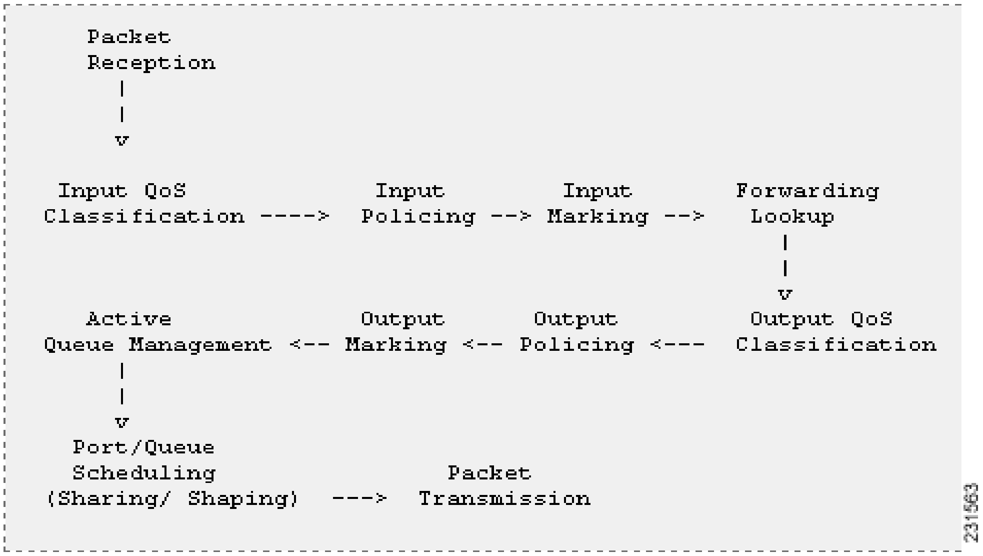
Step 5 Depending on the output classification, the packet is policed, its priority is optionally *(re-)marked*, and the transmit queue for the packet is determined depending on the traffic class.

Step 6 The transmit queue state is dynamically monitored via the AQM (Active Queue Management) algorithm and drop threshold configuration to determine whether the packet should be dropped or enqueued for transmission.

Step 7 If eligible for transmission, the packet is enqueued to a transmit queue. The transmit queue is selected based on output QoS classification criteria. The selected queue provides the desired behavior in terms of latency and bandwidth.

Figure 32-7 illustrates a high level model of Supervisor Engine 6-E.

Figure 32-7 QoS Packet Processing



```
     Packet
     Reception
       |
       |
       v

   Input QoS              Input        Input       Forwarding
  Classification ---->  Policing --> Marking -->    Lookup
                                                      |
                                                      |
                                                      v
     Active              Output      Output         Output QoS
  Queue Management <-- Marking <-- Policing <---   Classification
       |
       |
       v
    Port/Queue
    Scheduling                  Packet
  (Sharing/ Shaping)  --->  Transmission
```

Platform-supported Classification Criteria and QoS Features

The following table provides a summary of various classification criteria and actions supported on the Supervisor Engine 6-E. For details, refer to the *Catalyst 4500 Series Switch Command Reference*.

Supported classification actions	Descriptions
match access-group	Configures the match criteria for a class map on the basis of the specified ACL.
match any	Configures the match criteria for a class map to be successful match criteria for all packets.
match cos	Matches a packet based on a Layer 2 class of service (CoS) marking.
match destination-address mac	Uses the destination MAC address as a match criterion.
match source-address mac	Uses the source MAC address as a match criterion.
match [ip] dscp	Identifies a specific IP differentiated service code point (DSCP) value as a match criterion. Up to eight DSCP values can be included in one match statement.
match [ip] precedence	Identifies IP precedence values as match criteria.
match protocol	Configures the match criteria for a class map on the basis of the specified protocol.
match qos-group	Identifies a specific QoS group value as a match criterion. Applies only on the egress direction.
Supported Qos Features	**Descriptions**
police	Configures traffic policing.
police (percent)	Configures traffic policing on the basis of a percentage of bandwidth available on an interface.
police (two rates)	Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR).

Supported classification actions	Descriptions
service-policy	Specifies the name of a traffic policy used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another). At most two levels are supported.
set cos	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
set dscp	Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte of IPv4 or traffic class byte of IPv6 packet.
set precedence	Sets the precedence value in the packet header.
set qos-group	Sets a QoS group identifier (ID) that can be used later to classify packets.
table map support	Unconditional marking of one packet field based on another packet field.
priority	Gives priority to a class of traffic belonging to a policy map.
shape	Shapes traffic to the indicated bit rate according to the algorithm specified.
bandwidth	Provides a guaranteed minimum bandwidth to each of the four queues.
dbl	Dynamic buffer limit.

Platform Hardware Capabilities

Qos Actions	Numbers of entries supported
Classification	64k input and 64k output classification entries are supported. A given policy can use at most 24k ACLs
Policing	16K policers are supported. Policers are allocated to given direction in blocks of 2k. For example, 2k policers can be used in for input and 14k policers can be used for output. Single rate policers uses one policer entry. Single Rate Three Color Marker (srTCM) (RFC 2697) and Two Rate Three Color Marker (trTCM) (RFC 2698) uses two policer entries
Marking	Marking of Cos and DSCP/Precedence is supported through two marking tables, each capable of supporting 512 entries. There are separate tables for each direction.
Queuing	The queue size is fixed, depending on the number of ports. No maximum limit exists.
DBL	You can enable DBL action on all configured class-maps.

Prerequisites for Applying a QoS Service Policy

Unlike the Switch QoS model, there is no prerequisite for enabling QoS on various targets. Just the attachmenta of a serive policy enables QoS and detachment of that policy disables QoS on that target.

Restrictions for Applying a QoS Service Policy

Traffic marking can be configured on an interface, a VLAN, or a port and VLAN. An interface can be a Layer 2 access port, a Layer 2 switch trunk, a Layer 3 routed port, or an EtherChannel. A policy is attached to a VLAN using the *vlan configuration* mode.

Note There is no support for attaching a policy to an SVI.

Classification

Supervisor Engine 6-E supports classification of Layer 2, IP and IPv6 packets. Packet marking performed on input can be matched in the output direction. The previous table lists the full set of capabilities. By default, the Supervisor Engine 6-E also supports classification resources sharing.

By default, when the same policy is attached to a port or a VLAN or on per-port per-vlan targets, ACL entries are shared on the Supervisor Engine 6-E. Even though CAM entries are shared, QoS actions is unique on each target.

For example:

```
    class-map c1
      match ip any

Policy Map p1
    class ipp5
      police rate 1 m burst 200000
```

If policy-map p1 is applied to interfaces Gig 1/1 and Gig 1/2, 1 CAM entry is used (one ACE that matches IP packets), but 2 policers are allocated (one per target). So, all IP packets are policed to 1 mbps on interface Gig 1/1 and packets on interface Gig 1/2 are policed to 1 mbps.

Policing

Supervisor Engine 6-E supports policers in the following operation modes:

- Single Rate Policer Two Color Marker

  This kind of policer is configured with just the committed rate (CIR) and normal burst and it has only conform and exceed actions.

  This is the only form supported in the Supervisor Engine II-Plus to V-10GE based systems.

- Single Rate Three Color Marker (srTCM) (RFC 2697)
- Two Rate Three Color Marker (trTCM) (RFC 2698)
- Color Blind Mode

  Policing accuracy of 0.75% of configured policer rate.

  Supervisor Engine 6-E supports 16384 (16 x 1024, 16K) single rate, single burst policers. 16K policers are organized as 8 banks of 2K policers. The policer banks are dynamically assigned (input or output policer bank) by the software depending on the QoS configuration. So, the 16K policers are dynamically partitioned by software as follows:

  - 0 Input Policers and 16K Output Policers
  - 2K Input Policers and 14K Output Policers
  - 4K Input Policers and 12K Output Policers
  - 6K Input Policers and 10K Output Policers
  - 8K Input Policers and 8K Output Policers
  - 10K Input Policers and 6K Output Policers

  - 12K Input Policers and 4K Output Policers
  - 14K Input Policers and 2K Output Policers
  - 16K Input Policers and 0 Output Policers

These numbers represent individual policer entries in the hardware that support a single rate and burst parameter. Based on this, Supervisor Engines 6-E supports the following number of policers:

- 16K Single Rate Policer with Single Burst (Two Color Marker)
- 8K Single Rate Three Color Marker (srTCM)
- 8K Two Rate Three Color Marker (trTCM)

These policers are partitioned between Input and Output in chunks of 2K policer banks. The different types of policers can all co-exist in the system. However, a given type of policer (srTCM, trTCM etc.) is configurable as a block of 128 policers.

How to Implement Policing

For details on how to implement the policing features on a Catalyst 4500 series switch, refer to the Cisco IOS documentation at the following link:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a0080435d50.html

Platform Restrictions

Platform restrictions include the following:

- Multi-policer actions can be specified (setting CoS and IP DSCP is supported).
- Simultaneous unconditional and policer based marking is not supported.
- If policer based service-policy is attached to both a port and a VLAN, port-based policed is preferred by default. To over-ride a specific VLAN policy on a given port, then you must configure a per-port per-vlan policy.
- When you delete a port-channel with a per-port per-VLAN QoS policy, the switch crashes.

  Workaround: Before deleting the port-channel, do the following:

  1. Remove any per-port per-VLAN QoS policies, if any.

  2. Remove the VLAN configuration on the port-channel with the **no vlan-range** command.

Marking Network Traffic

Marking network traffic allows you to set or modify the attributes of traffic (that is, packets) belonging to a specific class or category. When used in conjunction with network traffic classification, marking network traffic is the foundation for enabling many quality of service (QoS) features on your network This module contains conceptual information and the configuration tasks for marking network traffic.

Contents

- "Traffic Marking Procedure Flowchart" section on page 32-75
- "Restrictions for Marking Network Traffic" section on page 32-76
- "Multi-attribute Marking Support" section on page 32-76
- "Hardware Capabilities for Marking" section on page 32-77
- "Configuring the Policy Map Marking Action" section on page 32-77
- "Marking Statistics" section on page 32-79

Information About Marking Network Traffic

To mark network traffic, you should understand the following concepts:

- "Purpose of Marking Network Traffic" section on page 32-73
- "Benefits of Marking Network Traffic" section on page 32-73
- "Two Methods for Marking Traffic Attributes" section on page 32-74

Purpose of Marking Network Traffic

Traffic marking is used to identify certain traffic types for unique handling, effectively partitioning network traffic into different categories.

After the network traffic is organized into classes by traffic classification, traffic marking allows you to mark (that is, set or change) a value (attribute) for the traffic belonging to a specific class. For instance, you may want to change the class of service (CoS) value from 2 to 1 in one class, or you may want to change the differentiated services code point (DSCP) value from 3 to 2 in another class. In this module, these values are referred to as attributes or marking fields.

Attributes that can be set and modified include the following:

- CoS value of a tagged Ethernet frame
- DSCP/Precedence value in the Type of Service (ToS) byte of IPv4.
- QoS group identifier (ID)
- DSCP /Precedence value in the traffic class byte of IPv6

Benefits of Marking Network Traffic

Traffic marking allows you to fine-tune the attributes for traffic on your network. This increased granularity helps isolate traffic that requires special handling, and thus, helps to achieve optimal application performance.

Traffic marking allows you to determine how traffic will be treated, based on how the attributes for the network traffic are set. It allows you to segment network traffic into multiple priority levels or classes of service based on those attributes, as follows:

- Traffic marking is often used to set the IP precedence or IP DSCP values for traffic entering a network. Networking devices within your network can then use the newly marked IP precedence values to determine how traffic should be treated. For example, voice traffic can be marked with a particular IP precedence or DSCP and strict priority can then be configured to put all packets of that marking into that queue. In this case, the marking was used to identify traffic for strict priority queue.
- Traffic marking can be used to identify traffic for any class-based QoS feature (any feature available in policy map class configuration mode, although some restrictions exist).

- Traffic marking can be used to assign traffic to a QoS group within a switch. The switch can use the QoS groups to determine how to prioritize traffic for transmission. The QoS group value is usually used for one of the two following reasons:
  - To leverage a large range of traffic classes. The QoS group value has 64 different individual markings, similar to DSCP.
  - If changing the Precedence or DSCP value is undesirable.

Two Methods for Marking Traffic Attributes

Note This section describes *Unconditional* marking, which differs from *Policer-based* marking. Unconditional marking is based solely on classification.

Method One: Unconditional Explicit Marking (using the set command)

You specify the traffic attribute you want to change with a set command configured in a policy map. The following table lists the available set commands and the corresponding attribute. For details on the set command, refer to the *Catalyst 4500 Series Switch Command Reference*.

Table 32-8 set Commands and Applicable Packet Types

set Commands	Traffic Attribute	Packet Type
set cos	Layer 2 CoS value of the outgoing traffic	Ethernet IPv4, IPv6
set dscp	DSCP value in the ToS byte	IPv4, IPv6
set precendence	precendence value in the packet header	IPv4, IPv6
set qos-group	QoS group ID	Ethernet, IPv4, IPv6

If you are using individual **set** commands, those set commands are specified in a policy map. The following is a sample of a policy map configured with one of the set commands listed in Table 32-8.

In this sample configuration, the **set cos** command has been configured in the policy map (policy1) to mark the CoS attribute:

```
enable
configure terminal
policy map p1
    class class1
      set cos 3
end
```

For information on configuring a policy map, see the "Creating a Policy Map" section on page 32-36.

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the "Attaching a Policy Map to an Interface" section on page 32-40.

Method Two: Unconditional Tablemap-based Marking

You can create a table map that can be used to mark traffic attributes. A table map is a kind of two-way conversion chart that lists and maps one traffic attribute to another. A table map supports a many-to-one type of conversion and mapping scheme. The table map establishes a to-from relationship for the traffic attributes and defines the change to be made to the attribute. That is, an attribute is set to one value that is taken from another value. The values are based on the specific attribute being changed. For instance, the Precedence attribute can be a number from 0 to 7, while the DSCP attribute can be a number from 0 to 63.

The following is a sample table map configuration:

```
table-map table-map1
map from 0 to 1
map from 2 to 3
exit
```

The following table lists the traffic attributes for which a to-from relationship can be established using the table map.

Table 32-9 Traffic Attributes for Which a To-From Relationship Can Be Established

The "To" Attribute	The "From" Attribute
Precedence	CoS, QoS group, DSCP, Precedence
DSCP	COS, QoS group, DSCP, Precendence
CoS	DSCP, QoS group, CoS, Precedence

The following is an example of a policy map (policy2) configured to use the table map (table-map1) created earlier:

```
Policy map policy
    class class-default
      set cos dscp table table-map

exit
```

In this example, a mapping relationship was created between the CoS attribute and the DSCP attribute as defined in the table map.

For information on configuring a policy map to use a table map, "Configuring a Policy Map" section on page 32-35.

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the "Attaching a Policy Map to an Interface" section on page 32-40.

Marking Action Drivers

A marking action can be triggered based on one of the two QoS processing steps.

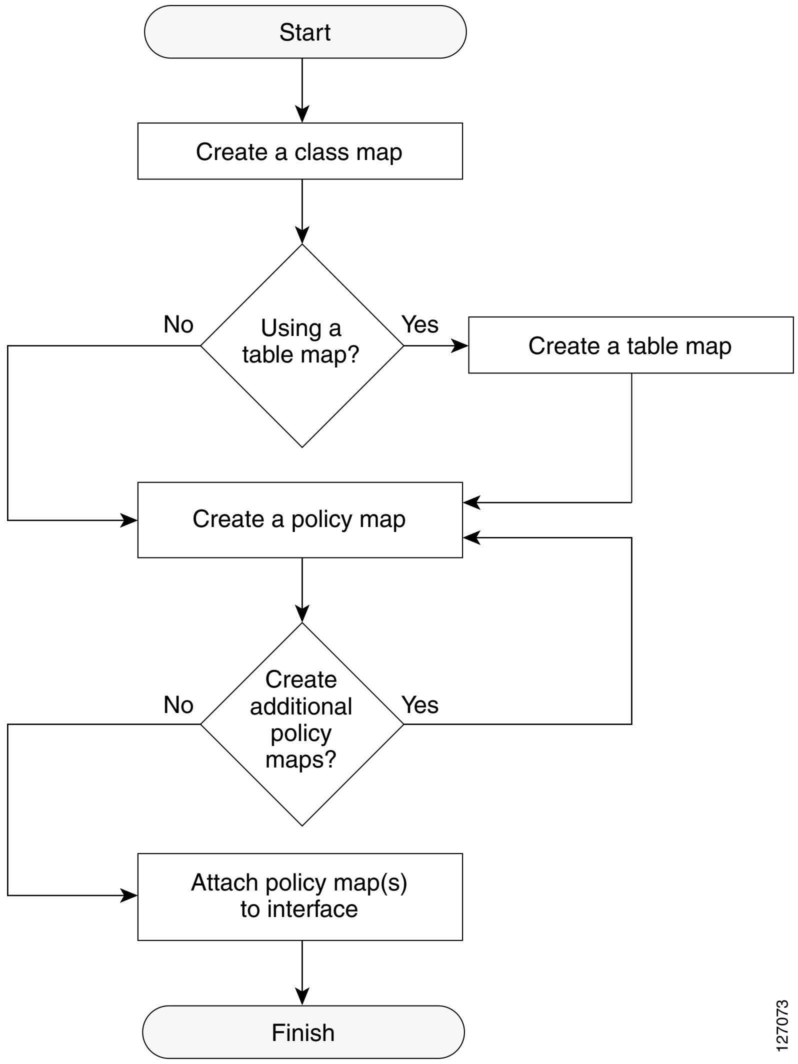
Classification based: In this case, all the traffic matching a class is marked using either explicit or tablemap based method. This method is referred to as *unconditional* marking.

Policer result-based: In this case, a class of traffic is marked differently based on the policer result (conform/exceed/violate) applicable to that packet. This method is referred to as *conditional* marking.

Traffic Marking Procedure Flowchart

Figure 32-8 illustrates the order of the procedures for configuring traffic marking.

Figure 32-8 Traffic marking Procedure Flowchart



Restrictions for Marking Network Traffic

The following restrictions apply to packet marking actions:

- QoS-group can be marked only in the input direction and can only support unconditional explicit marking.
- Only explicit marking is supported for policer-based marking.

Multi-attribute Marking Support

Supervisor Engine 6-E can mark more than one QoS attribute of a packet matching a class of traffic. For example, DSCP, CoS, and QoS-group can all be set together, using either explicit or tablemap-based marking.

Note When using unconditional explicit marking of multiple fields or policer-based multifield, multi-region (conform/exceed/violate) marking the number of tablemaps that can be setup in TOS or COS marking tables will be less than the maximum supported.

Hardware Capabilities for Marking

Supervisor Engine 6-E provides a 128 entry marking action table where each entry specifies the type of marking actions on COS and DSCP/precdence fields as well as policer action to transmit/markdown/drop a packet. One such table is supported for each direction, input and output. This table is used for both unconditional marking as well as policer-based marking. It can be used to support 128 unique marking actions or 32 unique policer-based actions or a combinations of the two.

For each of the marking fields (COS and DSCP), the Supervisor Engine 6-E provides 512 entry marking tables for each direction. These are similar to mapping tables available on supervisor engines that support the switch QoS model. However, these provide an ability to have multiple unique mapping tables that are setup by the user.

For example, the TOS marking table provides marking of DSCP/Precedence fields and can be used as one of the following:

- 8 different tablemaps with each mapping the 64 DSCP or qos-group values to another DSCP
- 64 (32) different tablemaps with each one mapping 8 CoS (16 CoS and CFi) values to DSCP in input (output) direction
- a combination of above two types of tablemaps

Similar mappings are available on the 512 entry COS marking table.

Configuring the Policy Map Marking Action

This section describes how to establish unconditional marking action for network traffic.

Prerequisites

Perform the following:

- Create a class map (*ipp5)* and a policy map. (Refer to the“Configuring a QoS Policy” section on page 32-32)
- Configure the marking action. (Refer to the “Configuring Policy-Map Class Actions” section on page 32-36)

Note On the Supervisor Engine 6-E, the marking action command options have been extended (refer to Table 32-8 on page 32-74 andTable 32-9 on page 32-75).

Configuring Tablemap-based Unconditional Marking

To configure table-map based unconditional marking, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# table-map name`	Configure a tablemap.

	Command	Purpose
Step 3	`Switch(config-tablemap)#` **`map from`** *`from_value`* **`to`** *`to_value`*	Create a map from a *from_value* to a *to_value*
Step 4	`Switch(config-tablemap)#` **`exit`**	Exit table-map configuration mode.
Step 5	`Switch(config)#` **`policy-map`** *`name`*	Enter policy-map configuration mode.
Step 6	`Switch(config-p)#` **`class`** `name`	Selects the class for QoS actions.
Step 7	`Switch(config-p-c)#` **`set cos`** `|` **`dscp`** `|` **`prec  cos`** `|` **`dscp`** `|` **`prec`** `|` **`qos-group`** `[`**`table`** *`name`*`]`	Selects the marking action based on an implicit or explicit table-map.
Step 8	`Switch(config-p-c)#` **`end`**	Exits configuration mode.
Step 9	`Switch#` **`show policy-map`** *`name`*	Verifies the configuration of the policy-map.
Step 10	`Switch#` **`show table-map name`**	Verifies the configuration of the table-map.

The following example shows how to enable marking action using table-map.

```
Switch(config)# table-map dscp2Cos
Switch(config-tablemap)# map from 8 to 1
Switch(config-tablemap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class ipp5
Switch(config-pmap-c)# set cos dscp table dscp2Cos
Switch(config-pmap-c)# end
Switch# show policy-map p1

Policy Map p1
    Class ipp5
      set cos dscp table dscp2Qos

Switch# show table-map dscp2Cos

 Table Map dscp2Cos
    from 8 to 1
    default copy
```

Configuring Policer Result-based Conditional Marking

To configure policer result-based conditional marking, setup a single rate or dual rate policer. Refer to the “How to Implement Policing” section on page 32-72.

This example shows how to configure a two rate three-color policer with explicit actions for each policer region:

```
Switch# configure terminal
Switch(config-pmap-c)# policer cir percent 20 pir percent 30
Switch(config-pmap-c-policer)# conform-action set-cos-transmit 3 set-dscp-transmit 10
Switch(config-pmap-c-policer)# exceed-action set-cos-transmit 4 set-dscp-transmit 20
Switch(config-pmap-c-policer)# violate action drop
Switch# show policy-map p1

  Policy Map police
    Class ipp5
     police cir percent 20 pir percent 30
       conform-action set-cos-transmit 3
       conform-action set-dscp-transmit af11
       exceed-action set-cos-transmit 4
       exceed-action set-dscp-transmit af22
       violate-action drop
```

Marking Statistics

The marking statistics indicate the number of packets that are *marked*.

For unconditional marking, the *classification entry* points to an entry in the marking action table that in turn indicates the fields in the packet that are marked. Therefore, the classification statistics by itself indicates the unconditional marking statistics.

For a conditional marking using policer, provided the policer is a packet rate policer, you cannot determine the number packets marked because the policer only provides byte statistics for different policing results.

Shaping, Sharing(Bandwidth), Priority Queuing and DBL

Supervisor Engine 6-E supports the Classification-based (class-based) mode for transmit queue selection. In this mode, the transmit queue selection is based on the Output QoS classification lookup.

Note Only output (egress) queuing is supported.

The Supervisor Engine 6-E hardware supports 4 transmit queues per port. Once the forwarding decision has been made to forward a packet out a port, the output QoS classification determines the transmit queue into which the packet needs to be enqueued.

By default, in Supervisor Engine 6-E, without any service policies associated with a port, there are two queues (a control packet queue and a default queue) with no guarantee as to the bandwidth or kind of prioritization. The only exception is that system generated control packets are euqueued into control packet queue so that control traffic receives some minimum link bandwidth.

Queues are assigned when an output policy attached to a port with one or more queuing related actions for one or more classes of traffic. Because there are only 4 queues per port, there can be at most 4 classes of traffic (including the reserved class, class-default) with queuing action(s). Classes of traffic that do not have any queuing action are referred to as *non-queuing* classes. Non-queuing class traffic ends up using the queue corresponding to class class-default.

When a queuing policy (a policy with queuing action) is attached, the control packet queue is deleted and the control packets are enqueued into respective queue per their classification.

Dynamic resizing of queues(queue limit class-map action) is not supported. Based on the chassis and line card type, all 4 queues on a port are configured with equal queue size.

Shaping

Shaping enables you to delay out-of-profile packets in queues so that they conform to a specified profile. Shaping is distinct from policing. Policing drops packets that exceed a configured threshold, whereas shaping *buffers* packets so that traffic remains within a given threshold. Shaping offers greater *smoothness* in handling traffic than policing. You enable average-rate traffic shaping on a traffic class with the **policy-map** class configuration command.

Supervisor Engine 6-E supports a range of 32kbps to 10 gbps for shaping, with a precision of approximately 1.5 percent.

When a queuing class is configured without any explicit shape configuration, the queue shape is set to the link rate.

To configure class-level shaping in a service policy, follow these steps:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# policy-map policy-map-name`	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	`Switch(config-pmap)# class class-name`	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	`Switch(config-pmap-class)# shape average {cir-bps kbps | percent percent}`	Enable average-rate traffic shaping. You can specify the bandwidth in kbps or as a percentage: • For *cir-bps*, specify the committed information rate, the bit rate that traffic is shaped to, in bps. The range is 32000 to 10000000000 bps. • For *percent*, specify the percentage of link rate to shape the class of traffic. The range is 1 to 100. By default, average-rate traffic shaping is disabled.
Step 5	`Switch(config-pmap-class)# exit`	Return to policy-map configuration mode.
Step 6	`Switch(config-pmap)# exit`	Return to global configuration mode.
Step 7	`Switch(config)# interface interface-id`	Specify a physical port and enter interface configuration mode.
Step 8	`Switch(config-interface)# service-policy output policy-map-name`	Specify the policy-map name, and apply it a physical interface.
Step 9	`Switch(config-interface)# end`	Return to privileged EXEC mode.

Step 10	`Switch# show policy-map [policy-map-name [class class-map-name]]`
	or
	`Switch# show policy-map interface interface-id`
		Verifies your entries.

Step 11	`Switch# copy running-config startup-config`	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map policy-map-name** global configuration command. To delete an existing class, use the **no class class-name policy-map** configuration command. To disable the average-rate traffic shaping, use the **no shape average policy-map** class configuration command.

This example shows how to configure class-level, average-rate shaping. It limits traffic class class1 to a data transmission rate of 256 kbps:

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
       shape average 256000
```

This example shows how to configure class-level, average shape percentage to 32% of link bandwidth for queuing-class traffic:

```
Switch# configure terminal
Switch(config)# policy-map queuing-policy
Switch(config-pmap)# class queuing-class
Switch(config-pmap-c)# shape average percent 32
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch #

Switch# show policy-map queuing-policy
  Policy Map queuing-policy
    Class queuing-class
       Average Rate Traffic Shaping
       cir 32%
```

Sharing(bandwidth)

The bandwidth assigned to a class of traffic is the minimum bandwidth that is guaranteed to the class during congestion. Transmit Queue Sharing is the processs by which output link bandwidth is shared among multiple queues of a given port.

Supervisor Engine 6-E supports a range of 32 kbps to 10 gbps for sharing, with a precision of approximately 1.5%. The sum of configured bandwidth across all queuing classes should not exceed the link bandwidth.

To configure class-level bandwidth action in a service policy, follow these steps:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# policy-map` *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.

	Command	Purpose
Step 3	`Switch(config-pmap)# class class-name`	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	`Switch(config-pmap-class)# bandwidth {bandwidth-kbps | percent percent}`	Specify the minimum bandwidth provided to a class belonging to the policy map when there is traffic congestion in the switch. If the switch is not congested, the class receives more bandwidth than you specify with the **bandwidth** command. By default, no bandwidth is specified. You can specify the bandwidth in kbps or as a percentage: o For *bandwidth-kbps*, specify the bandwidth amount in kbps assigned to the class. The range is 32 to 10000000. o For *percent*, specify the percentage of available bandwidth assigned to the class. The range is 1 to 100. Specify all the class bandwidths in either kbps or in percentages, but not a mix of both.
Step 5	`Switch(config-pmap-class)# exit`	Return to policy-map configuration mode.
Step 6	`Switch(config-pmap)# exit`	Return to global configuration mode.
Step 7	`Switch(config)# interface interface-id`	Specify a physical port and enter interface configuration mode.
Step 8	`Switch(config-interface)# service-policy output policy-map-name`	Specify the policy-map name, and apply it a physical interface.
Step 9	`Switch(config-interface)# end`	Return to privileged EXEC mode.

Step 10	`Switch# show policy-map [policy-map-name [class class-map-name]]`
	or
	`Switch# show policy-map interface interface-id`
		Verifies your entries.

Step 11	`Switch# copy running-config startup-config`	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map policy-map-name** global configuration command. To delete an existing class, use the **no class class-name policy-map** configuration command. To return to the default bandwidth, use the **no bandwidth policy-map** class configuration command.

This example shows how to create a class-level policy map called policy11 for three classes called prec1, prec2, and prec3. In the policy for these classes, 30 percent of the available bandwidth is assigned to the queue for the first class, 20 percent is assigned to the queue for the second class, and 10 percent is assigned to the queue for the third class.

```
Switch # configure terminal
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy11
Switch(config-if)# end
Switch #

Switch# show policy-map policy11
  Policy Map policy11
    Class prec1
      bandwidth percent 30
    Class prec2
      bandwidth percent 20
    Class prec3
      bandwidth percent 10
```

This example shows how to create a class-level policy map called policy11 for three classes called prec1, prec2, and prec3. In the policy for these classes, 300 mbps of the available bandwidth is assigned to the queue for the first class, 200 mbps is assigned to the queue for the second class, and 100 mbps is assigned to the queue for the third class.

```
Switch # configure terminal
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth 300000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth 200000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth 100000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy11
Switch(config-if)# end
Switch #

Switch# show policy-map policy11
  Policy Map policy11
    Class prec1
      bandwidth 300000 (kbps)
    Class prec2
      bandwidth 200000 (kbps)
    Class prec3
      bandwidth 100000 (kbps)
```

When a queuing class is configured without any explicit share/bandwidth configuration, because the queue is not guaranteed any minimum bandwidth, the hardware queue is programmed to get a share of any unallocated bandwidth on the port as shown in the following example.

If there is no bandwidth remaining for the new queue or if the unallocated bandwidth is not sufficient to meet the minimum configurable rate (32kbps) for all queues which do not have any explicit share/bandwidth configuration, then the policy association is rejected.

For example, if there are two queues as given below

```
policy-map queue-policy
   class q1
     bandwidth percent 10

   class q2
     bandwidth percent 20
```

then the bandwidth allocation for the queues is as follows

```
q1 = 10%
           q2 = 20%
class-default = 70%
```

Similarly, when another queuing class (say q3) is added without any explicit bandwidth (say, just a shape command), then the bandwidth allocation is

```
q1 = 10%
           q2 = 20%
           q3 = min(35%, q3-shape-rate)
class-default = max(35%, (100 - (q1 + q2 + q3 )))
```

Priority queuing

On Supervisor Engine 6-E only one transmit queue on a port can be configured as *strict priority* (termed Low Latency Queue, or LLQ).

LLQ provides strict-priority queuing for a traffic class. It enables delay-sensitive data, such as voice, to be sent *before* packets in other queues. The priority queue is serviced first until it is empty or until it is under its shape rate. Only one traffic stream can be destined for the priority queue per class-level policy. You enable the priority queue for a traffic class with the **priority policy-map class** configuration command at the class mode.

A LLQ can starve other queues unless it is rate limited. Supervisor Engine 6-E does not support *conditional policing* where a 2-parameter policer (rate, burst) becomes effective when the queue is *congested* (based on queue length). However, it supports application of an unconditional policer to rate limit packets enqueued to the strict priority queue.

To enable class-level priority queuing in a service policy, follow these steps:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`policy-map`** *`policy-map-name`*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	`Switch(config-pmap)#` **`class`** *`class-name`*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	`Switch(config-pmap-class)#` **`priority`**	Enable the strict-priority queue, and give priority to a class of traffic. By default, strict-priority queueing is disabled.
Step 5	`Switch(config-pmap-class)#` **`exit`**	Return to policy-map configuration mode.
Step 6	`Switch(config-pmap)#` **`exit`**	Return to global configuration mode.

	Command	Purpose
Step 7	`Switch(config)#` **`interface`** *`interface-id`*	Specify a physical port and enter interface configuration mode.

Step 8	`Switch(config-interface)#`
	`service-policy output` *`policy-map-name`*
		Specify the policy-map name, and apply it a physical interface.

Step 9	`Switch(config-interface)#` **`end`**	Return to privileged EXEC mode.

Step 10	`Switch#` **`show policy-map`** [*`policy-map-name`* [**`class`** *`class-map-name`*]]
	`or`
	`Switch#` **`show policy-map interface`** *`interface-id`*
		Verifies your entries.

Step 11	`Switch#` **`copy running-config startup-config`**	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map policy-map-name** global configuration command. To delete an existing class, use the **no class class-name policy-map** configuration command. To disable the priority queue, use the **no priority policy-map class** configuration command.

This example shows how to configure a class-level policy called policy1. Class 1 is configured as the priority queue, which is serviced first until it is empty.

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch #

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
      priority
```

Active Queue Management (AQM) via Dynamic Buffer Limiting (DBL)

AQM provides buffering control of traffic flows prior to queuing a packet into a transmit queue of a port. This is of significant interest in a shared memory switch, ensuring that certain flows do not hog the switch packet memory.

Note Supervisor Engine 6-E supports active switch buffer management via DBL.

Except for the default class of traffic (class class-default), you can configure DBL action only when at least one of the other queuing action is configured.

To configure class-level dbl action along with shaping in a service policy, follow these steps:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`policy-map`** *`policy-map-name`*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	`Switch(config-pmap)#` **`class`** *`class-name`*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	`Switch(config-pmap-class)#` **`shape average`** *`cir-bps`*	Enable average-rate traffic shaping. Specify the committed information rate, the bit rate that traffic is shaped to, in bps. The range is 32000 to 10000000000 bps. By default, average-rate traffic shaping is disabled.
Step 5	`Switch(config-pmap-class)#` **`dbl`**	Enable DBL on the queue associated with this class of traffic
Step 6	`Switch(config-pmap-class)#` **`exit`**	Return to policy-map configuration mode.
Step 7	`Switch(config-pmap)#` **`exit`**	Return to global configuration mode.
Step 8	`Switch(config)#` **`interface`** *`interface-id`*	Specify a physical port and enter interface configuration mode.
Step 9	`Switch(config-interface)#` **`service-policy output`** *`policy-map-name`*	Specify the policy-map name, and apply it a physical interface.
Step 10	`Switch(config-interface)#` **`end`**	Return to privileged EXEC mode.

Step 11	`Switch#` **`show policy-map`** `[`*`policy-map-name`* `[`**`class`** *`class-map-name`*`]]`
	`or`
	`Switch#` **`show policy-map interface`** *`interface-id`*
		Verifies your entries.

Step 12	`Switch#` **`copy running-config startup-config`**	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map policy-map-name** global configuration command. To delete an existing class, use the **no class class-name policy-map** configuration command. To disable DBL on the associated queue, use the **no dbl policy-map class** configuration command.

The following example shows how to configure class-level, DBL action along with average-rate shaping. It enables DBL on the queue associated with traffic-class *class1*.

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# dbl
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#
```

```
Switch# show policy-map policy1
  Policy Map policy1
    Class class1
       shape average 256000
     dbl
```

Transmit Queue Statistics

Transmit queue statistics are visible via the **show policy-map interface** command.

Hierarchical Policies

To support logical QoS semantics, support for hierarchical policies is essential. The following examples describe different ways to set up hierarchical policies to achieve different behaviors.

Example 1

The following example assumes that you have policed/marked a subset of the traffic sent to certain queues, and policed/marked the remaining traffic in the default queue:

```
Policy-map queue-policy
   class queue-class
     shape <...>
     bandwidth <...>
service-policy police-and-mark-traffic-class-1

   class queue2-class
     set <...>
     police <...>
service-policy police-and-mark-traffic-class-2

   class class-default
     set <...>
     police <...>
...
policy-map police-and-mark-traffic-class-1
   class traffic-class-1
     set <...>
     police <...>
```

Example 2

The following example assumes that all the traffic is policed/marked in aggregate and uses the same queueing classification policies. As per the MQC semantics, if a class does not use queueing action, the "class-default" acts as the queue for that class of traffic.

```
Policy-map queue-policy
   class queue1-class
     shape <...>
     bandwidth <...>

   class queue8-class
     shape <...>
     bandwidth <...>

class queue8-class
     shape <...>
     bandwidth <...>
policy-map port-level-policy
```

```
class traffic-class-1
     set <...>
     police <...>
...
class traffic-class-n
     set <...>
     police <...>

class class-default
service-policy police-and-mark-traffic-class-2
```

In contrast to Example 1, Example 2 is a typical way of configuring queueing.

Example 3

The following example shows how to shape a given port to a sub-rate and to apply queueing/policing policies.

Note Port shaping is *not* supported on the Supervisor Engine 6-E.

This configuration model involves the following steps:

Step 1 Setting policing/marking policy actions (bottom child policy).

Step 2 Setting queueing actions and policing/marking packets in each queue as set up in Step 1 (middle child policy).

Step 3 Setting of the port level shaping (parent policy).

The policy configuration might look like the following:

```
Policy-map port-level-policy
   class class-default
     shape percent 100
service-policy queuing-policy

Policy-map queuing-policy
   class queue1-class
     shape <...>
     bandwidth <...>
service-policy police-and-mark-traffic-class-1

...
class class-default
     shape <...>
     bandwidth <...>
service-policy police-and-mark-traffic-class-default

Policy-map police-and-mark-traffic-class-1
   class traffic-class-1
     shape <...>
     police <...>
...
```

Policy Associations

Supervisor Engine 6-E supports per-port, per-VLAN policies. The associated policies are attached to the interface, VLAN, and a specific VLAN on a given port, respectively.

For details, refer to the following link:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a0080435d50.html

Qos Action Restrictions

- The same actions cannot be performed multiple times in a given direction on different targets. In other words, it is not possible to police the packets both on port and VLAN in the input direction. However, the user can police on the input port and on the output VLAN.
- Queuing actions are only allowed in the egress direction and only on the physical port.

Qos Policy priorities

- If a policy on a port and a VLAN are configured with conflicting actions (such as policing or marking actions on both a port and VLAN), the port policy is picked.
- If policy on a VLAN on a given port must be over-written, the user can configure PV policy.

Qos Policy merging

Applicable policies are applied to a given packet in given direction. For example, if the user configures egress VLAN-based police and marking, followed by selective queuing on the port, then for this packet, actions from both policies will be applied.

Software QoS

At the highest level, there are two types of locally sourced traffic (such as control protocol packets, pings, and telnets) from the switch: high priority traffic (typically the control protocol packets like OSPF Hellos and STP) and low priority packets (all other packet types).

The QoS treatment for locally-sourced packets differs for the two types.

Supervisor Engine 6-E provides a way to apply QoS to packets processed in the software path. The packets that get this QoS treatment in software can be classified into two types: software switched packets and software generated packets.

On reception, software switched packets are sent to the CPU that in turn sends them out of another interface. For such packets, input software QoS provides input marking and output software QoS provides output marking and queue selection.

The software generated packets are the ones locally sourced by the switch. The type of output software QoS processing applied to these packets is the same as the one applied to software switched packets. The only difference in the two is that the software switched packets take input marking of the packet into account for output classification purpose.

High Priority Packets

High priority packets are marked as one of the following:

- internally with PAK_PRIORITY

- with IP Precedence of 6 (for IP packets)
- with CoS of 6 (for VLAN Tagged packets)

These packets behave as follows:

- They are not dropped because of any policing, AQM, drop thresholds (or any feature that can drop a packet) configured as per the egress service policy. However, they might be dropped because of hardware resource constraints (packet buffers, queue full, etc.).
- They are classified and marked as per the marking configuration of the egress service policy that could be a port or VLAN (refer to the "Policy Associations" section on page 32-89.
- These high priority packets are enqueued to queue on the egress port based on the following criteria:
  - If there is no egress queuing policy on the port, the packet is queued to a control packet queue that is setup separately from the default queue and has 5 percent of the link bandwidth reserved for it.
  - If there is an egress queuing policy on the port, the queue is selected based on the classification criteria applicable to the packet.

Low Priority Packets

Packets that are not considered high priority (as described previously) are considered *unimportant*. These include locally sourced pings, telnet, and other protocol packets. They undergo the same treatment as any other packet that is transiting the given transmit port including egress classification, marking and queuing.

CHAPTER 33

Configuring Voice Interfaces

This chapter describes how to configure voice interfaces for the Catalyst 4500 series switches.

This chapter includes the following major sections:

- Overview of Voice Interfaces, page 33-1
- Configuring a Port to Connect to a Cisco 7960 IP Phone, page 33-2
- Configuring Voice Ports for Voice and Data Traffic, page 33-3
- Overriding the CoS Priority of Incoming Frames, page 33-4
- Configuring Power, page 33-5

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of Voice Interfaces

Catalyst 4500 series switches can connect to a Cisco 7960 IP phone and carry IP voice traffic. If necessary, the switch can supply electrical power to the circuit connecting it to the Cisco 7960 IP phone.

Because the sound quality of an IP telephone call can deteriorate if the data is unevenly sent, the switch uses quality of service (QoS) based on IEEE 802.1p class of service (CoS). QoS uses classification and scheduling to transmit network traffic from the switch in a predictable manner. See Chapter 32, “Configuring Quality of Service,” for more information on QoS.

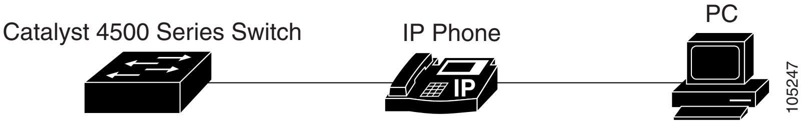
You can configure the Cisco 7960 IP phone to forward traffic with an 802.1p priority. You can use the CLI to configure a Catalyst 4000 Family to honor or ignore a traffic priority assigned by a Cisco 7960 IP phone.

The Cisco 7960 IP phone contains an integrated three-port 10/100 switch. The ports are dedicated connections as described below:

- Port 1 connects to the Catalyst 4500 series switch or other device that supports voice-over-IP.
- Port 2 is an internal 10/100 interface that carries the phone traffic.
- Port 3 connects to a PC or other device.

Figure 33-1 shows one way to configure a Cisco 7960 IP phone.

Figure 33-1 Cisco 7960 IP Phone Connected to a Catalyst 4500 Series Switch



Cisco IP Phone Voice Traffic

You can configure an 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 phone. You can configure access ports on the switch to send Cisco Discovery Protocol (CDP) packets that instruct an attached phone to send voice traffic to the switch in any of these 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).

Cisco IP Phone Data Traffic

The switch can also process tagged data traffic (traffic in IEEE 802.1Q or IEEE 802.1p frame types) from the device attached to the access port on the Cisco IP Phone (see Figure 33-1). You can configure Layer 2 access ports on the switch to send CDP packets that instruct the attached phone to configure the phone access port in one of these modes:

- In trusted mode, all traffic received through the access port on the Cisco IP Phone passes through the phone unchanged.
- In untrusted mode, all traffic in IEEE 802.1Q or IEEE 802.1p frames received through the access port on the Cisco IP Phone receive a configured Layer 2 CoS value. The default Layer 2 CoS value is 0. Untrusted mode is the default.

Note Untagged traffic from the device attached to the Cisco IP Phone passes through the phone unchanged, regardless of the trust state of the access port on the phone.

Configuring a Port to Connect to a Cisco 7960 IP Phone

Because a Cisco 7960 IP phone also supports connection to a PC or another device, an interface connecting a Catalyst 4500 series switch to a Cisco 7960 IP phone can carry a mix of voice and data traffic.

There are three configurations for a port connected to a Cisco 7960 IP phone:

- All traffic is transmitted according to the default CoS priority of the port. This is the default.

- Voice traffic is given a higher priority by the phone (CoS priority is always 5), and all traffic is in the same VLAN.
- Voice and data traffic are carried on separate VLANs.

To configure a port to instruct the phone to give voice traffic a higher priority and to forward all traffic through the 802.1Q native VLAN, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters configuration mode.
Step 2	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`	Specifies the interface to configure.
Step 3	`Switch(config-if)# switchport voice vlan dot1p`	Instructs the switch to use 802.1p priority tagging for voice traffic and to use VLAN 1 (default native VLAN) to carry all traffic.
Step 4	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show interface {fastethernet | gigabitethernet} slot/port switchport`	Verifies the port configuration.

Configuring Voice Ports for Voice and Data Traffic

Because voice and data traffic can travel through the same voice port, you should specify a different VLAN for each type of traffic. You can configure a switch port to forward voice and data traffic on different VLANs.

Note For information on configuring sticky port security on voice VLANs, see the Configuring Port Security on Voice Ports, page 35-22.

Note For information on using 802.1X with voice VLANs, see the “Using 802.1X with Voice VLAN Ports” section on page 34-18.

To configure a port to receive voice and data traffic from a Cisco IP Phone on different VLANs, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters configuration mode.
Step 2	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`	Specifies the interface to configure.
Step 3	`Switch(config-if)# switchport mode access`	Configures the interface as an access port. The voice VLAN is active only on access ports.
Step 4	`Switch(config-if)# switchport voice vlan vlan_num`	Instructs the Cisco IP phone to forward all voice traffic through a specified VLAN. The Cisco IP phone forwards the traffic with an 802.1p priority of 5.
Step 5	`Switch(config-if)# switchport access vlan data_vlan_num`	Configures the access VLAN (the data VLAN) on the port

	Command	Purpose
Step 6	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show interface {fastethernet | gigabitethernet} slot/port switchport`	Verifies the configuration.

In the following example, VLAN 1 carries data traffic, and VLAN 2 carries voice traffic. In this configuration, you must connect all Cisco IP phones and other voice-related devices to switch ports that belong to VLAN 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastEthernet 3/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 2
Switch(config-if)# switchport access vlan 3
Switch(config-if)# end
Switch# show interfaces fastEthernet 3/1 switchport
Name: Fa3/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 3 (VLAN0003)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 2 (VLAN0002)
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
Switch#
```

Overriding the CoS Priority of Incoming Frames

A PC or another data device can connect to a Cisco 7960 IP phone port. The PC can generate packets with an assigned CoS value. You can also use the switch CLI to override the priority of frames arriving on the phone port from connected devices, and you can set the phone port to accept (trust) the priority of frames arriving on the port.

To override the CoS priority setting received from the non-voice port on the Cisco 7960 IP phone, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters configuration mode.
Step 2	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`	Specifies the interface to configure.
Step 3	`Switch(config-if)# [no] qos trust extend cos 3`	Sets the phone port to override the priority received from the PC or the attached device and forward the received data with a priority of 3. Use the **no** keyword to return the port to its default setting.
Step 4	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show interface {fastethernet | gigabitethernet} slot/port switchport`	Verifies the change.

Configuring Power

The Catalyst 4500 series switch senses if it is connected to a Cisco 7960 IP phone. The Catalyst 4500 series switch can supply Power over Ethernet (PoE) to the Cisco 7960 IP phone if there is no power on the circuit. The Cisco 7960 IP phone can also be connected to an AC power source and supply its own power to the voice circuit. If there is power on the circuit, the switch does not supply it.

You can configure the switch not to supply power to the Cisco 7960 IP phone and to disable the detection mechanism. For information on the CLI commands that you can use to supply PoE to a Cisco 7960 IP phone, see Chapter 11, “Configuring Power over Ethernet.”

CHAPTER 34

Configuring 802.1X Port-Based Authentication

This chapter describes how to configure IEEE 802.1X port-based authentication to prevent unauthorized client devices from gaining access to the network.

This chapter includes the following major sections:

- Understanding 802.1X Port-Based Authentication, page 34-1
- Configuring 802.1X, page 34-21
- Displaying 802.1X Statistics and Status, page 34-48

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Understanding 802.1X Port-Based Authentication

802.1X defines 802.1X port-based authentication as a client-server based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. An authentication server validates each supplicant (client) connected to an authenticator (network access switch) port before making available any services offered by the switch or the LAN.

Note 802.1X support requires an authentication server that is configured for Remote Authentication Dial-In User Service (RADIUS). 802.1X authentication does not work unless the network access switch can route packets to the configured RADIUS server. To verify that the switch can route packets, you must ping the server from the switch.

Until a client is authenticated, only Extensible Authentication Protocol over LAN (EAPOL) traffic is allowed through the port to which the client is connected. After authentication succeeds, normal traffic can pass through the port.

To configure 802.1X port-based authentication, you need to understand the concepts in these sections:

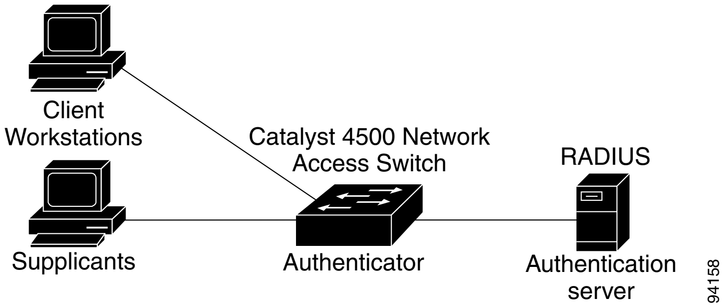
- Device Roles, page 34-2
- 802.1X and Network Access Control, page 34-3
- Authentication Initiation and Message Exchange, page 34-3

- Ports in Authorized and Unauthorized States, page 34-4
- 802.1X Host Mode, page 34-6
- Using 802.1X with VLAN Assignment, page 34-7
- Using 802.1X for Guest VLANs, page 34-8
- Using 802.1X with MAC Authentication Bypass, page 34-9
- Using 802.1X with Inaccessible Authentication Bypass, page 34-12
- Using 802.1X with Unidirectional Controlled Port, page 34-12
- Using 802.1X with Authentication Failed VLAN Assignment, page 34-13
- Using 802.1X with Port Security, page 34-15
- Using 802.1X with RADIUS-Provided Session Timeouts, page 34-16
- Using 802.1X with RADIUS Accounting, page 34-16
- Using 802.1X with Voice VLAN Ports, page 34-18
- Using Multiple Domain Authentication, page 34-19
- Supported Topologies, page 34-21

Device Roles

With 802.1X port-based authentication, network devices have specific roles. Figure 34-1 shows the role of each device, which is described below.

Figure 34-1 802.1X Device Roles



- Client—The workstation that requests access to the LAN, and responds to requests from the switch. The workstation must be running 802.1X-compliant client software.

Note For more information on 802.1X-compliant client application software such as Microsoft Windows 2000 Professional or Windows XP, refer to the Microsoft Knowledge Base article at this URL: http://support.microsoft.com

- Authenticator—Controls physical access to the network based on the authentication status of the client. The Catalyst 4500 series switch acts as an intermediary 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 encapsulates and decapsulates the Extensible Authentication Protocol (EAP) frames and interacts with the RADIUS 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 frame header is removed from the server, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client.

Note The Catalyst 4500 series switches must be running software that supports the RADIUS client and 802.1X.

- Authentication server—Performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch that the client is authorized to access the LAN and switch services. (The only supported authentication server is the RADIUS authentication server with EAP extensions; it is available in Cisco Secure Access Control Server version 3.2 and later.)

802.1X and Network Access Control

Network Access Control is a feature that allows port access policies to be influenced by the anti-virus posture of the authenticating device.

Anti-virus posture includes such elements as the operating system running on the device, the operating system version, whether anti-virus software is installed, what version of anti-virus signatures is available, etc. If the authenticating device has a NAC-aware 802.1X supplicant and the authentication server is configured to support NAC via 802.1X, anti-virus posture information is automatically included as part of the 802.1X authentication exchange.

For information on configuring NAC, refer to the URL:
http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_configuration_guide09186a00805764fd.html

Authentication Initiation and Message Exchange

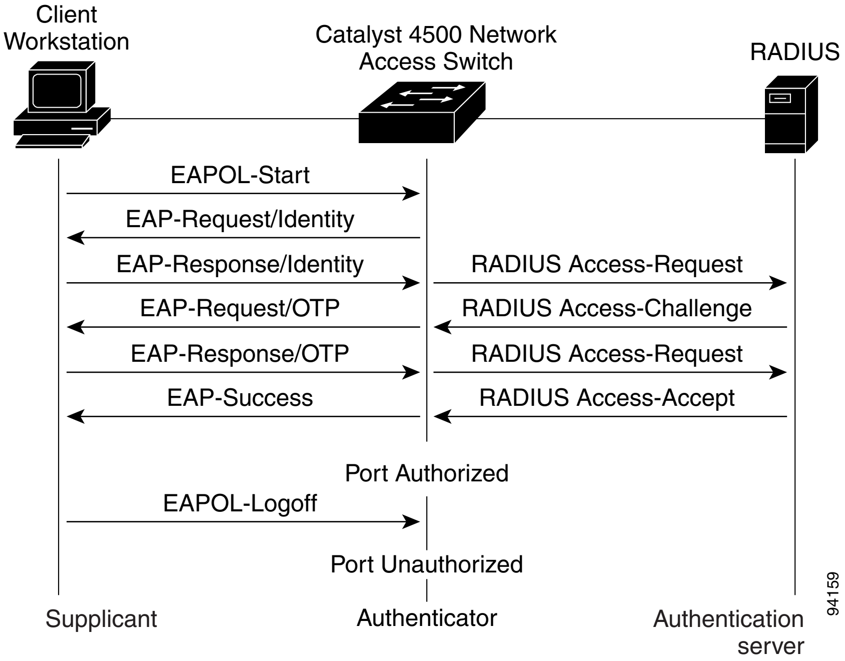
The switch or the client can initiate authentication. If you enable authentication on a port with the **dot1x port-control auto** interface configuration command, the switch must initiate authentication when it determines that the port link state has changed. It then sends an EAP-request/identity frame 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). Upon receipt of the frame, the client responds with an EAP-response/identity frame.

However, if during bootup, the client does not receive an EAP-request/identity frame from the switch, the client can initiate authentication by sending an EAPOL-start frame, which prompts the switch to request the client's identity.

If 802.1X is not enabled or supported on the network access switch, 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 means that the client has been successfully authenticated. 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.

The specific exchange of EAP frames depends on the authentication method being used. Figure 34-2 shows a message exchange that is initiated by the client using the One-Time Password (OTP) authentication method with an authentication server.

Figure 34-2 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 non-802.1X capable client 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. If a guest VLAN is configured on a port that connects to a client that does not support 802.1X, the port is placed in the configured guest VLAN and in the authorized state. For more information, see the “Using 802.1X for Guest VLANs” section on page 34-8.

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 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 can control the port authorization state with the **dot1x port-control** interface configuration command and these keywords:

- **force-authorized**—Disables 802.1X authentication and causes the port to transition to the authorized state without requiring authentication exchange. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This setting is the default.
- **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 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. The switch can uniquely identify each client attempting to access the network by 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 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.

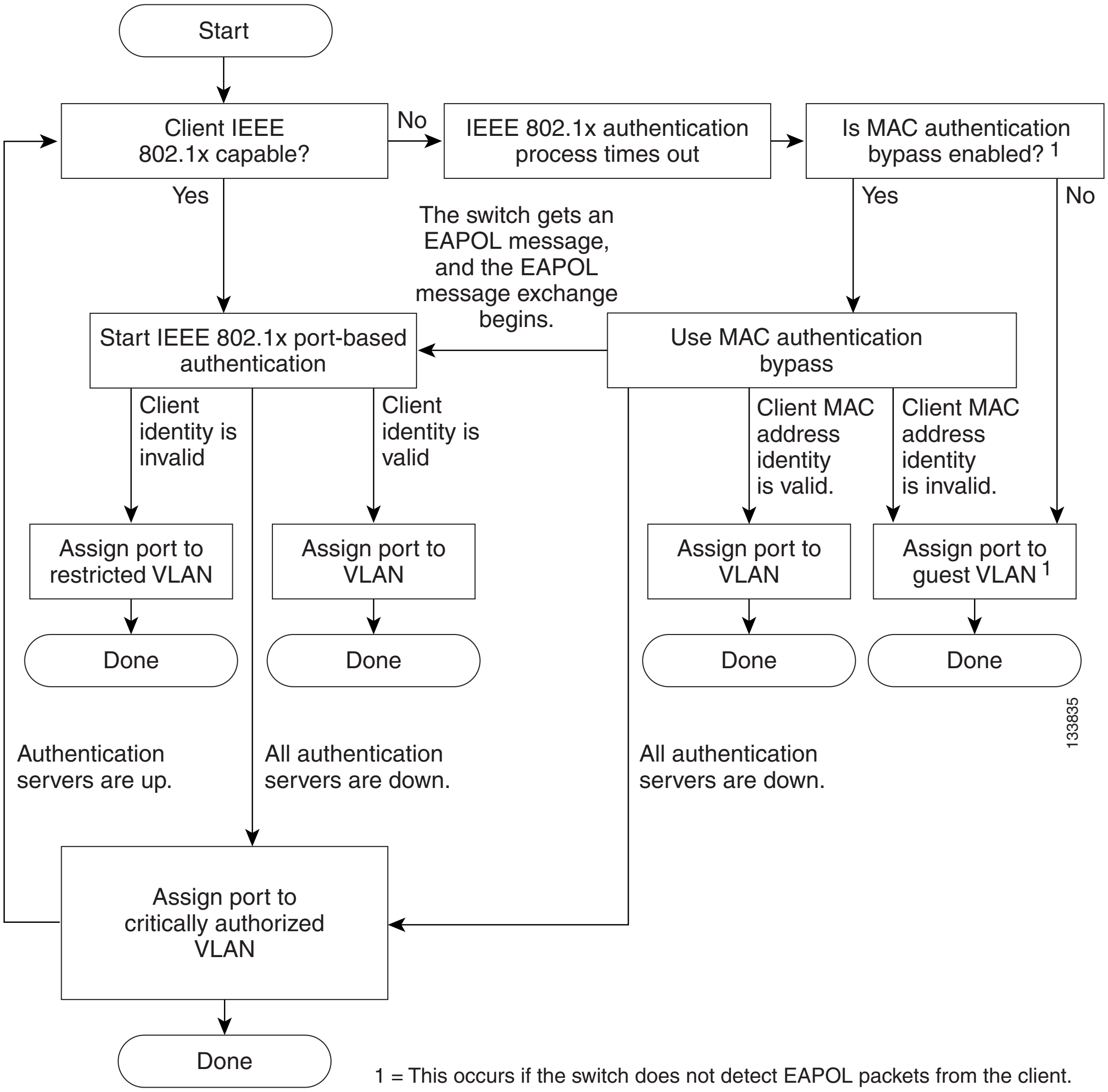
If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received by the port, the port returns to the unauthorized state.

Figure 34-3 shows the authentication process.

If Multidomain Authentication (MDA) is enabled on a port, this flow can be used with some exceptions that are applicable to voice authorization. For more information on MDA, see "Using Multiple Domain Authentication" section on page 34-19.

Note Supervisor Engine 6-E does *not* support MDA.

Figure 34-3 Authentication Flowchart


Start
Client IEEE 802.1x capable?
No
IEEE 802.1x authentication process times out
Is MAC authentication bypass enabled? 1
Yes
Yes
No
The switch gets an EAPOL message, and the EAPOL message exchange begins.
Start IEEE 802.1x port-based authentication
Use MAC authentication bypass
Client identity is invalid
Client identity is valid
Client MAC address identity is valid.
Client MAC address identity is invalid.
Assign port to restricted VLAN
Assign port to VLAN
Assign port to VLAN
Assign port to guest VLAN 1
Done
Done
Done
Done
133835
Authentication servers are up.
All authentication servers are down.
All authentication servers are down.
Assign port to critically authorized VLAN
Done
1 = This occurs if the switch does not detect EAPOL packets from the client.

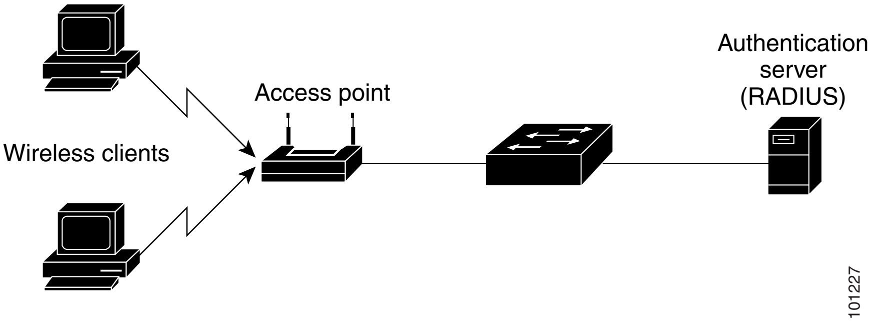
802.1X Host Mode

You can configure an 802.1X port for single-host or multiple-hosts mode. In single-host mode (see Figure 34-1 on page 34-2), only one client can be connected to the 802.1X-enabled switch port. The switch detects the client by sending an EAPOL frame when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

In multiple-hosts mode, you can attach multiple hosts to a single 802.1X-enabled port. Figure 34-4 on page 34-7 shows 802.1X port-based authentication in a wireless LAN. In this mode, only one of the attached clients must be authorized for all clients to be granted network access. If the port becomes unauthorized (re-authentication fails or an EAPOL-logoff message is received), the switch denies network access to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and it also acts as a client to the switch.

With multiple-hosts mode enabled, you can use 802.1X authentication to authenticate the port and port security to manage network access for all MAC addresses, including that of the client.

Figure 34-4 Multiple Host Mode Example



Cisco IOS Release 12.2(37)SG and later releases support Multi-Domain Authentication (MDA), which allows both a data device and a voice device, such as an IP Phone (Cisco or non-Cisco), to connect to the same switch port. For details on how to configure MDA, see the "Using Multiple Domain Authentication" section on page 34-19.

Using 802.1X with VLAN Assignment

You can use the VLAN assignment to limit network access for certain users. With the VLAN assignment, 802.1X-authenticated ports are assigned to a VLAN based on the username of the client connected to that port. The RADIUS server database maintains the username-to-VLAN mappings. After successful 802.1X authentication of the port, the RADIUS server sends the VLAN assignment to the switch. The VLAN can be a "standard" VLAN or a PVLAN.

On platforms that support PVLANs, you can isolate hosts by assigning ports into PVLANs.

When configured on the switch and the RADIUS server, 802.1X with VLAN assignment has these characteristics:

- If no VLAN is supplied by the RADIUS server, the port is configured in its access VLAN or isolated PVLAN when authentication succeeds.
- If the authentication server provides invalid VLAN information, the port remains unauthorized. This situation prevents ports from appearing unexpectedly in an inappropriate VLAN due to a configuration error.
- If the authentication server provides valid VLAN information, the port is authorized and placed in the specified VLAN when authentication succeeds.
- If the multiple-hosts mode is enabled, all hosts are in the same VLAN as the first authenticated user.
- If 802.1X is disabled on the port, the port is returned to the configured access VLAN.
- A port must be configured as an access port (which can be assigned only into "regular" VLANs), or as a PVLAN host port (which can be assigned only into PVLANs). Configuring a port as a PVLAN host port implies that all hosts on the port are assigned into PVLANs, whether their posture is compliant or non-compliant. If the type of the VLAN named in the Access-Accept does not match the type of VLAN expected to be assigned to the port (regular VLAN to access port, secondary private VLAN to private VLAN host port), the VLAN assignment fails.

- If a guest VLAN is configured to handle non-responsive hosts, the type of VLAN configured as the guest VLAN must match the port type (that is, guest VLANs configured on access ports must be standard VLANs, and guest VLANs configured on PVLAN host ports must be PVLANs. If the guest VLAN's type does not match the port type, non-responsive hosts are treated as if no guest VLAN is configured (that is, they are denied network access).
- To assign a port into a PVLAN, the named VLAN must be a secondary PVLAN. The switch determines the implied primary VLAN from the locally configured secondary-primary association.

Note If you change the access VLAN or PVLAN host VLAN mapping on a port that is already authorized in a RADIUS assigned VLAN, the port remains in the RADIUS assigned VLAN.

To configure VLAN assignment you need to perform these tasks:

- Enable AAA authorization with the **network** keyword to allow interface configuration from the RADIUS server. For an illustration of how to apply the **aaa authorization network group radius** command, refer to the section "Enabling 802.1X Authentication" on page 23.
- Enable 802.1X. (The VLAN assignment feature is automatically enabled when you configure 802.1X on an access port.)
- Assign vendor-specific tunnel attributes in the RADIUS server. To ensure proper VLAN assignment, the RADIUS server must return these attributes to the switch:
  - Tunnel-Type = VLAN
  - Tunnel-Medium-Type = 802
  - Tunnel-Private-Group-ID = VLAN NAME

Using 802.1X for Guest VLANs

Note Supervisor Engine 6-E does *not* support this feature.

You can use guest VLANs to enable non-802.1X-capable hosts to access networks that use 802.1X authentication. For example, you can use guest VLANs while you are upgrading your system to support 802.1X authentication.

Guest VLANs are supported on a per-port basis, and you can use any VLAN as a guest VLAN as long as its type matches the type of the port. If a port is already forwarding on the guest VLAN and you enable 802.1X support on the network interface of the host, the port is immediately moved out of the guest VLAN and the authenticator waits for authentication to occur.

Enabling 802.1X authentication on a port starts the 802.1X protocol. If the host fails to respond to packets from the authenticator within a certain amount of time, the authenticator brings the port up in the configured guest VLAN.

If the port is configured as a PVLAN host port, the guest VLAN must be a secondary PVLAN. If the port is configured as an access port, the guest VLAN must be a regular VLAN. If the guest VLAN configured on a port is not appropriate for the type of the port, the switch behaves as if no guest VLAN is configured (that is, non-responsive hosts are denied network access).

For details on how to configure guest VLANs, see the "Configuring 802.1X with Guest VLANs" section on page 34-32.

Usage Guidelines for Using 802.1X Authentication with Guest VLANs

The usage guidelines for using 802.1X authentication with guest VLANs are as follows:

- When you reconfigure a guest VLAN to a different VLAN, any authentication failed ports are also moved and the ports stay in their current authorized state.
- When you shut down or remove a guest VLAN from the VLAN database, any authentication failed ports are immediately moved to an unauthorized state and the authentication process is restarted.

Note No periodic reauthentication is allowed with guest VLANs.

Usage Guidelines for Using 802.1X Authentication with Guest VLANs on Windows-XP Hosts

The usage guidelines for using 802.1X authentication with guest VLANs on Windows-XP hosts are as follows:

- If the host fails to respond to the authenticator, the port attempts to connect three times (with a 30 second timeout between each attempt). After this time, the login/password window does not appear on the host, so you must unplug and reconnect the network interface cable.
- Hosts responding with an incorrect login/password fail authentication. Hosts failing authentication are not put in the guest VLAN. The first time that a host fails authentication, the quiet-period timer starts, and no activity occurs for the duration of the quiet-period timer. When the quiet-period timer expires, the host is presented with the login/password window. If the host fails authentication for the second time, the quiet-period timer starts again, and no activity occurs for the duration of the quiet-period timer. The host is presented with the login/password window a third time. If the host fails authentication the third time, the port is placed in the unauthorized state, and you must disconnect and reconnect the network interface cable.

Using 802.1X with MAC Authentication Bypass

Note Supervisor Engine 6-E does *not* support this feature.

The 802.1X protocol has 3 entities: client (supplicant), authenticator, and authentication server. Typically, the host PC runs the supplicant software and tries to authenticate itself by sending its credentials to the authenticator which in turn relays that info to the authentication server for authentication.

However, not all hosts may have supplicant functionality. Devices that cannot authenticate themselves using 802.1X , which still should have network access, can use MAC Authentication Bypass (MAB), which uses the connecting device's MAC address to grant/deny network access.

Typically, you would use this feature on ports where devices such as printers are connected. Such devices do not have 802.1X supplicant functionality.

In a typical deployment, the RADIUS server maintains a database of MAC addresses that require access. When this feature detects a new MAC address on a port, it generates a RADIUS request with both username and password as the device's MAC address. After authorization succeeds, the port is accessible to the particular device through the same code path that 802.1X authentication would take when processing an 802.1X supplicant. If authentication fails, the port moves to the guest VLAN if configured, or it remains unauthorized.

The Catalyst 4500 series switch also supports re-authentication of MACs on a per port level. Be aware that the re-authentication functionality is provided by 802.1X and is not MAB specific. In the re-authentication mode, a port stays in the previous RADIUS-sent VLAN and tries to re-authenticate itself. If the re-authentication succeeds, the port stays in the RADIUS-sent VLAN. Otherwise, the port becomes unauthorized and moves to the guest VLAN if one is configured.

For details on how to configure MAB, see the "Configuring 802.1X with MAC Authentication Bypass" section on page 34-35.

Feature Interaction

This section lists feature interactions and restrictions when MAB is enabled. If a feature is not listed, assume that it interacts seamlessly with MAB (such as Unidirectional Controlled Port).

- MAB can only be enabled if 802.1X is configured on a port. MAB functions as a fall back mechanism for authorizing MACs. If you configure both MAB and 802.1X on a port, the port attempts to authenticate using 802.1X. If the host fails to respond to EAPOL requests and MAB is configured, the 802.1X port is opened up to listen to packets and to grab a MAC address, rather than attempt to authenticate endlessly.

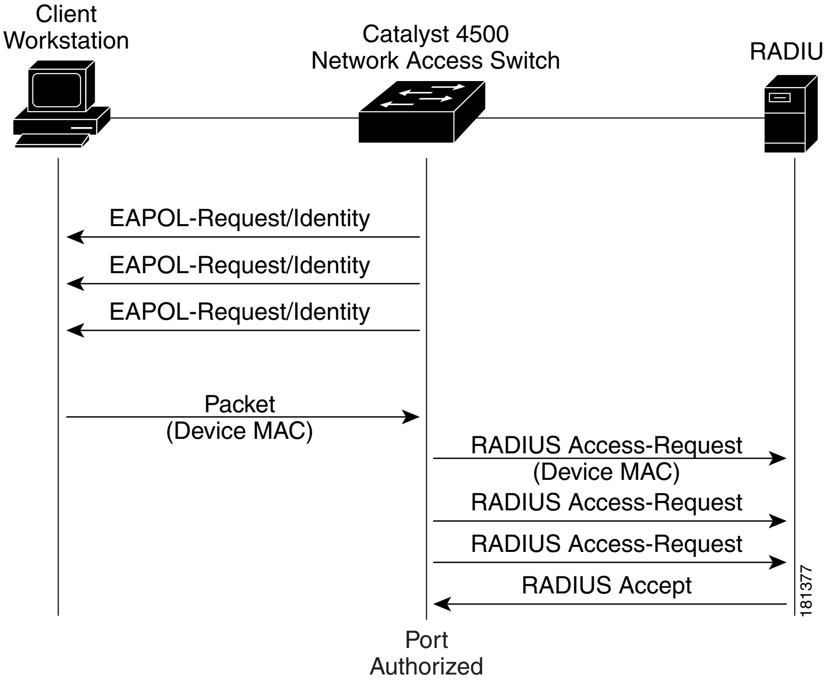
  Based on the default 802.1X timer values, the transition between mechanisms takes approximately 90 seconds. You can shorten the time by reducing the value of the transmission period time, which affects the frequency of EAPOL transmission. A smaller timer value results in EAPOLs sent during a shorter period of time. With MAB enabled, after 802.1X performs one full set of EAPOLs, the learned MAC address is forwarded to the authentication server for processing.

  The MAB module performs authorization for the first MAC address detected on the wire. The port is considered authorized once a valid MAC address is received that RADIUS approves of.

  802.1X authentication can re-start if an EAPOL packet is received on a port that was initially authorized as a result of MAB.

  Figure 34-5 shows the message exchange during MAB.

Figure 34-5 Message Exchange during MAC Authentication Bypass



- The authentication-failed VLAN is used only with dot1x-authentication-failed users. MAB is not attempted with dot1x-authentication-failed users. If 802.1X authentication fails, a port moves to the authentication-failed VLAN (if configured) whether MAB is configured or not.
- When both MAB and guest VLAN are configured and no EAPOL packets are received on a port, the 802.1X state-machine is moved to a MAB state where it opens the port to listen to traffic and grab MAC addresses. The port remains in this state forever waiting to see a MAC on the port. A detected MAC address that fails authorization causes the port to be moved to the guest VLAN if configured.

  While in a guest VLAN, a port is open to all traffic on the specified guest VLAN. Therefore, non-802.1X supplicants that normally would be authorized but are in guest VLAN due to the earlier detection of a device that failed authorization, would remain in the guest VLAN indefinitely. However, loss of link or the detection of an EAPOL on the wire causes a transition out of the guest VLAN and back to the default 802.1X mode.
- Once a new MAC has been authenticated by MAB, the responsibility to limit access falls upon the 802.1X Authenticator (or port security) to secure the port. The 802.1X default host parameter is defined only for a single host. If the port is changed to multi-user host, port security must be employed to enforce the number of MAC addresses allowed thru this port.

- Catalyst 4500 series switch supports MAB with VVID, with the restriction that the MAC address appears on a port data VLAN only. All IP phone MACs learned via CDP are allowed on voice VLANs.
- MAB and VMPS are mutually exclusive because their functionality overlaps.

Using 802.1X with Inaccessible Authentication Bypass

Note Supervisor Engine 6-E does *not* support this feature.

When a switch cannot reach the configured RADIUS servers and clients (supplicants) cannot be authenticated, you can configure a switch to allow network access to hosts connected to *critical* ports that are enabled for Inaccessible Authentication Bypass.

When this feature is enabled, a switch monitors the status of the configured RADIUS servers. If no RADIUS servers are available, ports with Inaccessible Authentication Bypass enabled are authorized. You can specify a Inaccessible Authentication Bypass VLAN on a per-port basis.

Ports that were already authorized when RADIUS becomes unavailable are unaffected by Inaccessible Authentication Bypass. However, if re-authentication is applied and RADIUS is not restored by the next polling cycle, ports already authorized falls back to the critical auth VLAN.

When RADIUS becomes available, critically-authorized ports may be configured to automatically reauthenticate themselves.

For details on how to configure Inaccessible Authentication Bypass, see the “Configuring 802.1X with Inaccessible Authentication Bypass” section on page 34-36.

Using 802.1X with Unidirectional Controlled Port

Note Supervisor Engine 6-E does *not* support this feature.

Unidirectional Controlled Port is a combined hardware/software feature that allows dormant PCs to be “powered on” based on the receipt of a specific Ethernet frame, known as the *magic packet*. Generally, Unidirectional Controlled Port is used in environments where administrators plan to manage remote systems during off-hours, when it’s likely that the systems have been powered down.

Use of Unidirectional Controlled Port with hosts attached through 802.1X ports presents a unique problem; when the host powers down, a 802.1X port becomes unauthorized. In this state, the port allows the receipt and transmission of EAPoL packets only. Therefore, the Unidirectional Controlled Port magic packet cannot reach the host; without powering up, the PC cannot authenticate and open the port.

Unidirectional Controlled Port solves this problem by allowing packets to be transmitted on unauthorized 802.1X ports.

Note Unidirectional Controlled Port only works when Spanning Tree Portfast is enabled on the port.

For details on how to configure 802.1X with Unidirectional Controlled Port, see the “Configuring 802.1X with Unidirectional Controlled Port” section on page 34-39

Unidirectional State

When you configure a port as unidirectional with the **dot1x control-direction in** interface configuration command, the port changes to the spanning-tree forwarding state.

When Unidirectional Controlled Port is enabled, the connected host is in the sleeping mode or power-down state. The host does not exchange traffic with other devices in the network. If the host connected to the unidirectional port that cannot send traffic to the network, the host can only receive traffic from other devices in the network.

Bidirectional State

When you configure a port as bidirectional with the **dot1x control-direction both** interface configuration command, the port is access-controlled in both directions. In this state, except EAPOL packets, the switch port does not receive or send packets.

Using 802.1X with Authentication Failed VLAN Assignment

Note Supervisor Engine 6-E does *not* support this feature.

You can use authentication-failed VLAN assignment on a per-port basis to provide access for authentication failed users. Authentication failed users are end hosts that are 802.1X- capable but do not have valid credentials in an authentication server or end hosts that do not give any username and password combination in the authentication pop-up window on the user side.

If a user fails the authentication process, that port is placed in the authentication-failed VLAN. The port remains in the authentication-failed VLAN until the reauthentication timer expires. When the reauthentication timer expires the switch starts sending the port re-authentication requests. If the port fails reauthentication it remains in the authentication-failed VLAN. If the port is successfully reauthenticated, the port is moved either to the VLAN sent by RADIUS server or to the newly authenticated ports configured VLAN; the location depends on whether RADIUS is configured to send VLAN information.

Note When enabling periodic reauthentication (see the "Enabling Periodic Reauthentication" section on page 34-42), only local reauthentication timer values are allowed. You cannot utilize a RADIUS server to assign the reauthentication timer value.

You can set the maximum number of authentication attempts that the authenticator sends before moving a port into the authentication-failed VLAN. The authenticator keeps a count of the failed authentication attempts for each port. A failed authentication attempt is either an empty response or an EAP failure. The authenticator tracks any mix of failed authentication attempts towards the authentication attempt count. After the maximum number of attempts is reached the port is placed in the authentication-failed VLAN until the reauthentication timer expires again.

Note RADIUS may send a response without an EAP packet in it when it does not support EAP, and sometimes third party RADIUS servers also send empty responses. When this happens, the authentication attempt counter is incremented.

For details on how to configure Authentication Failed VLAN Assignment, see the "Configuring 802.1X with Authentication Failed VLAN Assignment" section on page 34-40.

Usage Guidelines for Using Authentication Failed VLAN Assignment

- You should enable reauthentication. The ports in authentication-failed VLANs do not receive reauthentication attempts if reauthentication is disabled. In order to start the reauthentication process the authentication-failed VLAN must receive a link down event or an EAP logoff event from the port. If the host is behind a hub, you may never get a link down event and may not detect the new host until the next reauthentication occurs. Therefore, it is recommended to have re-authentication enabled in that case.
- EAP failure messages are not sent to the user. If the user failures authentication the port is moved to an authentication-failed VLAN and a EAP success message is sent to the user. Because the user is not notified of the authentication failure there may be confusion as to why there is restricted access to the network. A EAP Success message is sent for the following reasons:
  - If the EAP Success message is not sent, the user tries to authenticate every 60 seconds (by default) by sending an EAP-start message.
  - In some cases, users have configured DHCP to EAP-Success and unless the user sees a success, DHCP does not work on the port.
- Sometimes a user caches an incorrect username and password combination after receiving a EAP success message from the authenticator and reuses that information in every re-authentication. Until the user passes the correct username and password combination the port remains in the authentication-failed VLAN.
- When an authentication failed port is moved to an unauthorized state the authentication process is restarted. If you should fail the authentication process again the authenticator waits in the held state. After you have correctly reauthenticated all 802.1X ports are reinitialized and treated as normal 802.1X ports.
- When you reconfigure an authentication-failed VLAN to a different VLAN, any authentication failed ports are also moved and the ports stay in their current authorized state.
- When you shut down or remove an authentication-failed VLAN from the VLAN database, any authentication failed ports are immediately moved to an unauthorized state and the authentication process is restarted. The authenticator does not wait in a held state because the authentication-failed VLAN configuration still exists. While the authentication-failed VLAN is inactive, all authentication attempts are counted, and as soon as the VLAN becomes active the port is placed in the authentication-failed VLAN.
- If you reconfigure the maximum number of authentication failures allowed by the VLAN, the change takes affect after the reauthentication timer expires.
- All internal VLANs which are used for Layer 3 ports cannot be configured as an authentication-failed VLAN.
- You cannot configure a VLAN to be both an authentication-failed VLAN and a voice VLAN. If you do, a syslog message is generated when the port tries to come up in the authentication-failed VLAN.
- The authentication-failed VLAN is supported only in single-host mode (the default port mode).
- When a port is placed in an authentication-failed VLAN the user's MAC address is added to the mac-address-table. If a new MAC address appears on the port, it is treated as a security violation.
- When an authentication failed port is moved to an authentication-failed VLAN, the Catalyst 4500 series switch does not transmit a RADIUS-Account Start Message like it does for regular 802.1X authentication.

Using 802.1X with Port Security

You can enable port security on an 802.1X port in either single- or multiple-host mode. (To do so, you must configure port security with the **switchport port-security** interface configuration command. Refer to the nb chapter in this guide.) When you enable port security and 802.1X on a port, 802.1X authenticates the port, and port security manages the number of MAC addresses allowed on that port, including that of the client. Hence, you can use an 802.1X port with port security enabled to limit the number or group of clients that can access the network.

For information on selecting multi-host mode, see the "Resetting the 802.1X Configuration to the Default Values" section on page 34-48.

These examples describe the interaction between 802.1X and port security on a switch:

- When a client is authenticated, and the port security table is not full, the client's MAC address is added to the port security list of secure hosts. The port then proceeds to come up normally.

  When a client is authenticated and manually configured for port security, it is guaranteed an entry in the secure host table (unless port security static aging has been enabled).

  A security violation occurs if an additional host is learned on the port. The action taken depends on which feature (802.1X or port security) detects the security violation:

  - If 802.1X detects the violation, the action is to err-disable the port.
  - If port security detects the violation, the action is to shutdown or restrict the port (the action is configurable).

  The following describes when port security and 802.1X security violations occur:

  - In single host mode, after the port is authorized, any MAC address received other than the client's causes a 802.1X security violation.
  - In single host mode, if installation of an 802.1X client's MAC address fails because port security has already reached its limit (due to a configured secure MAC addresses), a port security violation is triggered.
  - In multi host mode, once the port is authorized, any additional MAC addresses that cannot be installed because the port security has reached its limit triggers a port security violation.

- When an 802.1X client logs off, the port transitions back to an unauthenticated state, and all dynamic entries in the secure host table are cleared, including the entry for the client. Normal authentication then ensues.
- If you administratively shutdown the port, the port becomes unauthenticated, and all dynamic entries are removed from the secure host table.
- Only 802.1X can remove the client's MAC address from the port security table. Note that in multi host mode, with the exception of the client's MAC address, all MAC addresses that are learned by port security can be deleted using port security CLIs.
- Whenever port security ages out a 802.1X client's MAC address, 802.1X attempts to reauthenticate the client. Only if the reauthentication succeeds is the client's MAC address be retained in the port security table.
- All of the 802.1X client's MAC addresses are tagged with (dot1x) when you display the port security table by using CLI.

Using 802.1X with RADIUS-Provided Session Timeouts

You can specify whether a switch uses a locally configured or a RADIUS-provided reauthentication timeout. If the switch is configured to use the local timeout, it reauthenticates the host when the timer expires.

If the switch is configured to use the RADIUS-provided timeout, it looks in the RADIUS Access-Accept message for the Session-Timeout and optional Termination-Action attributes. The switch uses the value of the Session-Timeout attribute to determine the duration of the session, and it uses the value of the Termination-Action attribute to determine the switch action when the session's timer expires.

If the Termination-Action attribute is present and its value is RADIUS-Request, the switch reauthenticates the host. If the Termination-Action attribute is not present, or its value is Default, the switch terminates the session.

Note The supplicant on the port detects that its session has been terminated and attempts to initiate a new session. Unless the authentication server treats this new session differently, the client may see only a brief interruption in network connectivity as the switch sets up a new session.

If the switch is configured to use the RADIUS-supplied timeout, but the Access-Accept message does not include a Session-Timeout attribute, the switch never reauthenticates the supplicant. This behavior is consistent with Cisco's wireless access points.

For details on how to configure RADIUS-provided session timeouts, see the “Configuring RADIUS-Provided Session Timeouts” section on page 34-31.

Using 802.1X with RADIUS Accounting

Note Supervisor Engine 6-E does *not* support this feature.

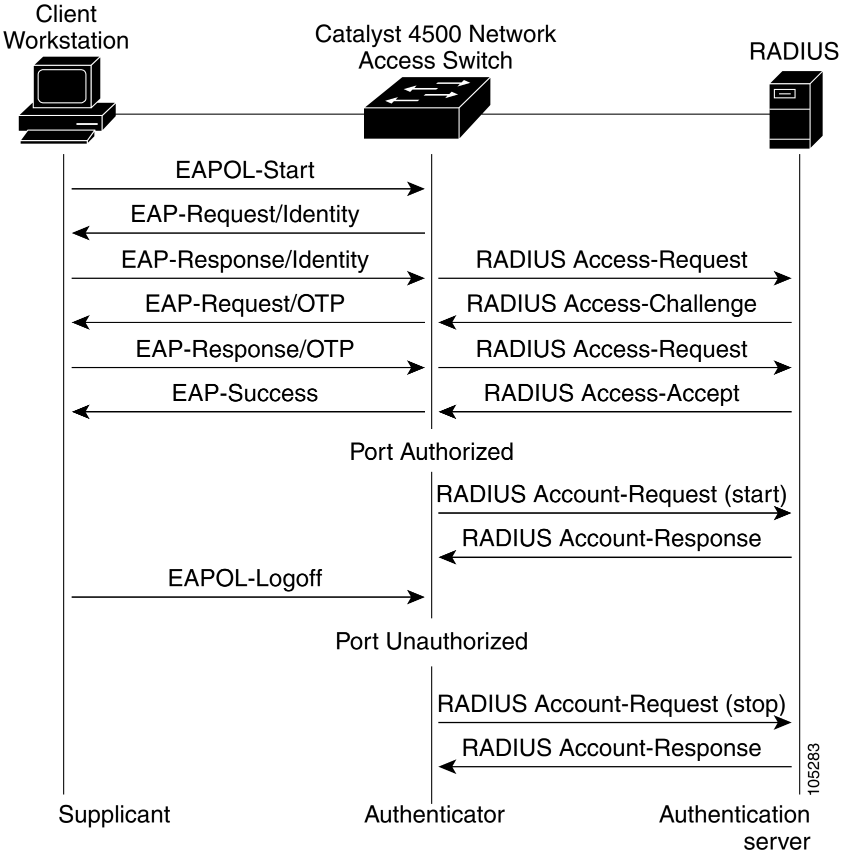
Note If you plan to implement system-wide accounting, you should also configure 802.1X accounting. Moreover, you need to inform the accounting server of the system reload event when the system is reloaded. Doing this ensures that the accounting server is aware that all outstanding 802.1X sessions on this system are closed.

Note To enable 802.1X accounting, you must first configure 802.1X authentication and switch-to-RADIUS server communication.

802.1X RADIUS accounting relays important events to the RADIUS server (such as the client’s connection session). This session is defined as the interval beginning when the client is authorized to use the port and ending when the client stops using the port.

Figure 34-6 illustrates the RADIUS accounting process.

Figure 34-6 RADIUS Accounting



Note You must configure the 802.1X client to send an EAP-logoff (Stop) message to the switch when the user logs off. If you do not configure the 802.1X client, an EAP-logoff message is not sent to the switch and the accompanying accounting Stop message is not be sent to the authentication server. Refer to the Microsoft Knowledge Base article at the location: http://support.microsoft.com. Also refer to the Microsoft article at this location:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/columns/cableguy/cg0703.asp,

and set the SupplicantMode registry to 3 and the AuthMode registry to 1.

After the client is authenticated, the switch sends accounting-request packets to the RADIUS server, which responds with accounting-response packets to acknowledge the receipt of the request.

A RADIUS accounting-request packet contains one or more Attribute-Value pairs to report various events and related information to the RADIUS server. The following events are tracked:

- User successfully authenticates
- User logs-off
- Link-down occurs on a 802.1X port
- Reauthentication succeeds
- Reauthentication fails

When the port state transitions between authorized and unauthorized, the RADIUS messages are transmitted to the RADIUS server.

The switch does not log any accounting information. Instead, it sends such information to the RADIUS server, which must be configured to log accounting messages.

The 802.1X authentication, authorization and accounting process is as follows:

Step 1 A user connects to a port on the switch.

Step 2 Authentication is performed, for example, using the username/password method.

Step 3 VLAN assignment is enabled, as appropriate, per RADIUS server configuration.

Step 4 The switch sends a start message to an accounting server.

Step 5 Reauthentication is performed, as necessary.

Step 6 The switch sends an interim accounting update to the accounting server that is based on the result of reauthentication.

Step 7 The user disconnects from the port.

Step 8 The switch sends a stop message to the accounting server.

To configure 802.1X accounting, you need to do the following tasks:

- Enable logging of “Update/Watchdog packets from this AAA client” in your RADIUS server’s Network Configuration tab.
- Enable “Logging>CVS RADIUS Accounting” in your RADIUS server System Configuration tab.
- Enable 802.1X accounting on your switch.
- Enable AAA accounting by using the **aaa system accounting** command. Refer to the “Enabling 802.1X RADIUS Accounting” section on page 34-32.

Enabling AAA system accounting along with 802.1X accounting allows system reload events to be sent to the accounting RADIUS server for logging. By doing this, the accounting RADIUS server can infer that all active 802.1X sessions are appropriately closed.

Because RADIUS uses the unreliable transport protocol UDP, accounting messages may be lost due to poor network conditions. If the switch does not receive the accounting response message from the RADIUS server after a configurable number of retransmissions of an accounting request, the following system message appears:

```
Accounting message %s for session %s failed to receive Accounting Response.
```

When the stop message is not transmitted successfully, a message like the following appears:

```
00:09:55: %RADIUS-3-NOACCOUNTINGRESPONSE: Accounting message Start for session
172.20.50.145 sam 11/06/03 07:01:16 11000002 failed to receive Accounting Response.
```

Use the **show radius statistics** command to display the number of RADIUS messages that do not receive the accounting response message.

Using 802.1X with Voice VLAN Ports

Note Supervisor Engine 6-E does *not* support this feature.

A voice VLAN port is a special access port associated with two VLAN identifiers:

- Voice VLAN ID (VVID) to carry voice traffic to and from the IP phone. The VVID is used to configure the IP phone connected to the port.
- Port VLAN ID (PVID) to carry the data traffic to and from the workstation connected to the switch through the IP phone. The PVID is the native VLAN of the port.

Each port that you configure for a voice VLAN is associated with a VVID and a PVID. This configuration allows voice traffic and data traffic to be separated onto different VLANs.

A voice VLAN port becomes active when there is a link whether or not the port is AUTHORIZED or UNAUTHORIZED. All traffic coming through the voice VLAN is learned correctly and appears in the MAC-address-table. Cisco IP phones do not relay CDP messages from other devices. As a result, if several Cisco IP phones are connected in series, the switch recognizes only the one directly connected to it. When 802.1X is enabled on a voice VLAN port, the switch drops packets from unrecognized Cisco IP phones more than one hop away.

When 802.1X is enabled on a port, you cannot configure a PVID that is equal to a VVID. For more information about voice VLANs, see Chapter 33, “Configuring Voice Interfaces.”

Be aware of the following feature interactions:

- 802.1X VLAN assignment cannot assign to the port the same VLAN as the voice VLAN; otherwise, the 802.1X authentication fails.
- 802.1X guest VLAN works with the 802.1X voice VLAN port feature. However, the guest VLAN cannot be the same as the voice VLAN.
- 802.1X port security works with the 802.1X voice VLAN port feature and is configured per port. Two MAC addresses must be configured: one for the Cisco IP phone MAC address on the VVID and one for the PC MAC-address on PVID.

  However, you cannot use the 802.1X voice VLAN port feature with 802.1X port security’s sticky MAC address configuration and statically configured MAC address configuration.
- 802.1X accounting is unaffected by the 802.1X voice VLAN port feature.
- When 802.1X is configured on a port, you cannot connect multiple IP-phones to a Catalyst 4500 series switch through a hub.
- Because voice VLANs cannot be configured as private VLAN host ports, and because only private VLANs can be assigned to private VLAN host ports, VLAN assignment cannot assign a private VLAN to a port with a voice VLAN configured.

For details on how to configure 802.1X with voice VLANs, see the “Configuring 802.1X with Voice VLAN” section on page 34-41.

Using Multiple Domain Authentication

Note Supervisor Engine 6-E does *not* support this feature.

Multiple Domain Authentication (MDA) allows both a data device and a voice device, such as an IP phone (Cisco or non-Cisco), to authenticate on the same switch port, which is divided into a data domain and a voice domain.

MDA does not enforce the order of device authentication. For best results, however, you should authenticate a voice device before you authenticate a data device on an MDA-enabled port.

Observe the following guidelines for configuring MDA:

- To configure a switch port for MDA, see the “Configuring Multiple Domain Authentication” section on page 34-28.
- You must configure the voice VLAN for the IP phone when the host mode is set to multidomain. For more information, see Chapter 33, “Configuring Voice Interfaces.”

Note If you use a dynamic VLAN to assign a voice VLAN on an MDA-enabled switch port, the voice device fails authorization.

- To authorize a voice device, the AAA server must be configured to send a Cisco Attribute-Value (AV) pair attribute with a value of `device-traffic-class=voice`. Without this value, the switch treats the voice device as a data device.
- The guest VLAN and restricted VLAN features only apply to the data devices on an MDA-enabled port. The switch treats a voice device that fails authorization as a data device.
- If more than one device attempts authorization on either the voice or the data domain of a port, it is error disabled.
- Until a device is authorized, the port drops its traffic. Non-Cisco IP phones or voice devices are allowed into both the data and voice VLANs. The data VLAN allows the voice device to contact a DHCP server to obtain an IP address and acquire the voice VLAN information. After the voice device starts sending on the voice VLAN, its access to the data VLAN is blocked.
- You can use dynamic VLAN assignment from a RADIUS server only for data devices.

Note Supervisor Engine 6-E does *not* support dynamic VLAN.

- MDA can use MAC authentication bypass as a fallback mechanism to allow the switch port to connect to devices that do not support 802.1X authentication. This is especially useful for 3rd-party phones without 802.1X supplicant. For more information, see the “Using 802.1X with MAC Authentication Bypass” section on page 34-9.
- When a *data* or a *voice* device is detected on a port, its MAC address is blocked until authorization succeeds. If the authorization fails, the MAC address remains blocked for 5 minutes.
- If more than five devices are detected on the *data* VLAN or more than one voice device is detected on the *voice* VLAN while a port is unauthorized, the port is error disabled.
- When a port host mode is changed from single- or multihost to multidomain mode, an authorized data device remains authorized on the port. However, a Cisco IP phone that has been allowed on the port in the voice VLAN is automatically removed and must be reauthenticated on that port.
- Active fallback mechanisms such as guest VLAN and restricted VLAN remain configured after a port changes from single- or multihost mode to multidomain mode.
- Switching a port host mode from multidomain to single- or multihost mode removes all authorized devices from the port.
- If a data domain is authorized first and placed in the guest VLAN, non-802.1X-capable voice devices need to tag their packets on the voice VLAN to trigger authentication.
- We do not recommend per-user ACLs with an MDA-enabled port. An authorized device with a per-user ACL policy might impact traffic on both the voice and data VLANs of the port. If used, only one device on the port should enforce per-user ACLs.

Supported Topologies

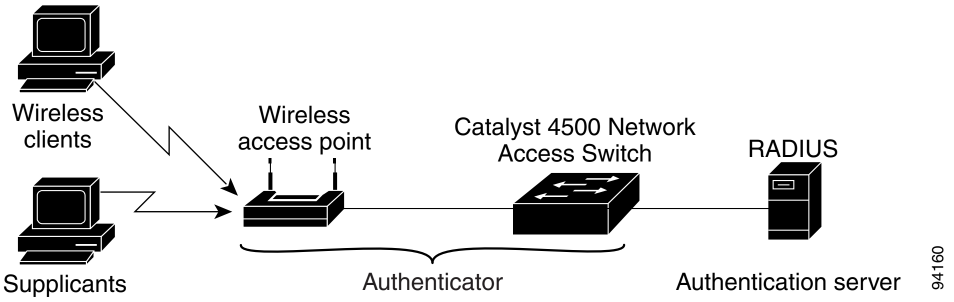
The 802.1X port-based authentication supports two topologies:

- Point to point
- Wireless LAN

In a point-to-point configuration (see Figure 34-1 on page 34-2), only one client can be connected to the 802.1X-enabled switch port when the multi-host mode is not enabled (the default). 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.

For 802.1X port-based authentication in a wireless LAN (Figure 34-7), you must configure the 802.1X port as a multiple-host port that is authorized as a wireless access point once the client is authenticated. (See the "Resetting the 802.1X Configuration to the Default Values" section on page 34-48.) When the port is authorized, all other hosts that are 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 for all wireless access point-attached clients. In this topology, the wireless access point is responsible for authenticating clients attached to it, and the wireless access point acts as a client to the switch.

Figure 34-7 Wireless LAN Example



Configuring 802.1X

To configure 802.1X, follow this procedure:

Step 1 Enable 802.1X authentication. See the "Enabling 802.1X Authentication" section on page 34-23.

Step 2 Configure switch to RADIUS server communication. See the "Configuring Switch-to-RADIUS-Server Communication" section on page 34-26.

Step 3 Adjust the 802.1X timer values. See the "Changing the Quiet Period" section on page 34-44.

Step 4 Configure optional features. See the "Configuring RADIUS-Provided Session Timeouts" section on page 34-31.

These sections describe how to configure 802.1X:

- Default 802.1X Configuration, page 34-22
- 802.1X Configuration Guidelines, page 34-23
- Enabling 802.1X Authentication, page 34-23 (required)
- Configuring Switch-to-RADIUS-Server Communication, page 34-26 (required)
- Configuring Multiple Domain Authentication, page 34-28
- Configuring RADIUS-Provided Session Timeouts, page 34-31 (optional)
- Enabling 802.1X RADIUS Accounting, page 34-32 (optional)
- Configuring 802.1X with Guest VLANs, page 34-32 (optional)
- Configuring 802.1X with MAC Authentication Bypass, page 34-35 (optional)
- Configuring 802.1X with Inaccessible Authentication Bypass, page 34-36 (optional)
- Configuring 802.1X with Unidirectional Controlled Port, page 34-39 (optional)
- Configuring 802.1X with Authentication Failed VLAN Assignment, page 34-40 (optional)
- Configuring 802.1X with Voice VLAN, page 34-41 (optional)
- Enabling Periodic Reauthentication, page 34-42 (optional)
- Enabling Multiple Hosts, page 34-43 (optional
- Changing the Quiet Period, page 34-44 (optional)
- Changing the Switch-to-Client Retransmission Time, page 34-45 (optional)
- Setting the Switch-to-Client Frame-Retransmission Number, page 34-46 (optional)
- Manually Reauthenticating a Client Connected to a Port, page 34-47 (optional)
- Initializing the 802.1X Authentication State, page 34-47
- Removing 802.1X Client Information, page 34-48
- Resetting the 802.1X Configuration to the Default Values, page 34-48 (optional)

Default 802.1X Configuration

Table 34-1 shows the default 802.1X configuration.

Table 34-1 Default 802.1X Configuration

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled
RADIUS server • IP address • UDP authentication port • Key	 • None specified • 1812 • None specified
Per-interface 802.1X protocol enable state	Force-authorized The port transmits and receives normal traffic without 802.1X-based authentication of the client.

Table 34-1 Default 802.1X Configuration (continued)

Feature	Default Setting
Periodic reauthentication	Disabled
Time between reauthentication attempts	3600 sec
Quiet period	60 sec Number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client.
Retransmission time	30 sec 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 Number of times that the switch sends an EAP-request/identity frame before restarting the authentication process.
Multiple host support	Disabled
Client timeout period	30 sec When relaying a request from the authentication server to the client, the amount of time that the switch waits for a response before retransmitting the request to the client.
Authentication server timeout period	30 sec When relaying a response from the client to the authentication server, the amount of time that the switch waits for a reply before retransmitting the response to the server. This setting is not configurable.

802.1X Configuration Guidelines

This section describes the guidelines for configuring 802.1X authentication:

- The 802.1X Protocol is supported only on Layer 2 static access, private VLAN host ports, and Layer 3 routed ports. You cannot configure 802.1X for any other port modes.
- If you are planning to use either 802.1X accounting or VLAN assignment, be aware that both features utilize general AAA commands. For information on how to configure AAA, refer to the “Enabling 802.1X Authentication” section on page 34-23. Alternatively, you can refer to the Cisco IOS security documentation:
  - http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fsecur_c/index.htm
  - http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/fsecur_r/index.htm

Enabling 802.1X Authentication

To enable 802.1X port-based authentication, you first must enable 802.1X globally on your switch, then enable AAA and specify the authentication method list. A method list describes the sequence and authentication methods that must be queried to authenticate a user.

The software uses the first method listed in the method list to authenticate users; if that method fails to respond, the software selects the next authentication method in the 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.

Note To allow VLAN assignment, you must enable AAA authorization to configure the switch for all network-related service requests.

To configure 802.1X port-based authentication, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.

Step 2	`Switch(config)#`
	`dot1x system-auth-control`
		Enables 802.1X on your switch.
		To disable 802.1X globally on the switch, use the **no dot1x system-auth-control** command.

Step 3	`Switch(config)#` **`aaa new-model`**	Enables AAA. To disable AAA, use the **no aaa new-model** command.
Step 4	`Switch(config)#` **`aaa authentication dot1x`** {**`default`**} *`method1`* [*`method2`*...]	Creates an 802.1X AAA authentication method list. 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. To disable 802.1X AAA authentication, use the **no aaa authentication dot1x** {**default** | *list-name*} *method1* [*method2*...] global configuration command.
Step 5	`Switch(config)#` **`aaa authorization network {default} group radius`**	(Optional) Configure the switch for user RADIUS authorization for all network-related service requests, such as VLAN assignment.
Step 6	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 7	`Switch(config-if)#` **`switchport mode access`**	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 8	`Switch(config-if)#` **`dot1x pae authenticator`**	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 9	`Switch(config-if)#` **`dot1x port-control auto`**	Enables 802.1X authentication on the interface.
Step 10	`Switch(config-if)#` **`end`**	Returns to privileged EXEC mode.

	Command	Purpose
Step 11	`Switch # `**`show dot1x interface`** *`interface-id`* **`details`**	Verifies your entries. Check the PortControl row in the 802.1X port summary section of this display. The PortControl value is set to **auto**.
Step 12	`Switch# `**`show running-config`**	Verifies your entries.
Step 13	`Switch# `**`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Note Enabling Spanning Tree PortFast ensures that a port comes up immediately after authorization.

Note Whenever you configure any 802.1X parameter on a port, a dot1x authenticator is automatically created on the port. As a result **dot1x pae authenticator** appears in the configuration. This is to ensure that dot1x authentication still works on legacy configurations without manual intervention. This is likely to change in future releases.

This example shows how to enable 802.1X and AAA on Fast Ethernet port 2/1, and how to verify the configuration:

```
Switch# configure terminal
Switch(config)# dot1x system-auth-control
Switch(config)# aaa new-model
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# interface fastethernet2/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch# show dot1x interface f7/1 details

Dot1x Info for FastEthernet7/1
-----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = AUTO
ControlDirection          = Both
HostMode                  = SINGLE_HOST
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod           = 0

Dot1x Authenticator Client List
-------------------------------
Supplicant                = 1000.0000.2e00
        Auth SM State     = AUTHENTICATED
        Auth BEND SM Stat = IDLE
Port Status               = AUTHORIZED
```

```
Authentication Method      = Dot1x
Authorized By              = Authentication Server
Vlan Policy                = N/A
```

Configuring Switch-to-RADIUS-Server Communication

A RADIUS security server is identified by its host name or IP address, host name and specific UDP port number, or 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 fail-over backup to the first one. The RADIUS host entries are tried in the order they were configured.

To configure the RADIUS server parameters on the switch, perform this task:

	Command	Purpose
Step 1	`Switch#` **configure terminal**	Enters global configuration mode.
Step 2	`Switch(config)#` **radius-server host** {*hostname* | *ip-address*} **auth-port** *port-number* [**acct-port** *port-number*] [**test username** *name*] [**ignore-auth-port**] [**ignore-acct-port**] [**idle-time** *min*] **key** *string*	Configures the RADIUS server parameters on the switch. For *hostname* | *ip-address,* specify the hostname or IP address of the remote RADIUS server. To delete the specified RADIUS server, use the **no radius-server host** {*hostname* | *ip-address*} global configuration command. The **auth-port** *port-number* specifies the UDP destination port for authentication requests. The default is 1812. The **acct-port** *port-number* specifies the UDP destination port for accounting requests. The default is 1813. Use **test username** *name* to enable automated RADIUS server testing, and to detect the RADIUS server going up and down. The **name** parameter is the username used in the test access request sent to the RADIUS server; it does not need to be a valid user configured on the server. The **ignore-auth-port** and **ignore-acct-port** options disable testing on the authentication and accounting ports respectively. The **idle-time** *min* parameter specifies the number of minutes before an idle RADIUS server is tested to verify that it is still up. The default is 60 minutes. The **key** *string* specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server. **Note** Always configure the key as the last item in the **radius-server host** command syntax because leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon. If you want to use multiple RADIUS servers, use this command multiple times.
Step 3	`Switch(config-if)#` **radius deadtime** *min*	(Optional) Configures the number of minutes before a dead RADIUS server is tested to check whether it has come back up. The default is 1 minute.
Step 4	`Switch(config-if)#` **radius dead-criteria time** *seconds* **tries** *num*	(Optional) Configures the criteria used to decide whether a RADIUS server is dead. The **time** parameter specifies the number of seconds after which a request to the server is unanswered before it is considered dead. The **tries** parameter specifies the number of times a request to the server is unanswered before it is considered dead. The recommended values for these parameters are **tries** equal to **radius-server retransmit** and **time** equal to **radius-server retransmit** x **radius-server timeout**.

	Command	Purpose
Step 5	`Switch(config-if)# ip radius source-interface m/p`	Establishes the IP address to be used as the source address for all outgoing RADIUS packets.
Step 6	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show running-config`	Verifies your entries.
Step 8	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to specify the server with IP address 172.120.39.46 as the RADIUS server. The first command specifies port 1612 as the authorization port, sets the encryption key to rad123.

The second command dictates that key matches are performed on the RADIUS server:

```
Switch# configure terminal
Switch(config)# radius-server host 172.120.39.46 auth-port 1612 key rad123
Switch(config)# ip radius source-interface m/p
Switch(config)# end
Switch#
```

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.

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.

Configuring Multiple Domain Authentication

To configure MDA, perform these steps.

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# radius-server vsa send authentication`	Configures the network access server to recognize and use vendor-specific attributes (VSAs).
Step 3	`Switch(config)# interface` *interface-id*	Specifies the port to which multiple hosts are indirectly attached, and enters interface configuration mode.

	Command	Purpose
Step 4	`Switch(config-if)# [no] dot1x host-mode {single-host | multi-host | multi-domain}`	The keywords have these meanings: • **single-host**–Allow a single host (client) on an IEEE 802.1X-authorized port. • **multi-host**–Allow multiple hosts on an 802.1X-authorized port after a single host has been authenticated. • **multi-domain**–Allow both a host and a voice device, such as an IP phone (Cisco or non-Cisco), to be authenticated on an IEEE 802.1X-authorized port. **Note** You must configure the voice VLAN for the IP phone when the host mode is set to **multi-domain**. For more information, see Chapter 33, "Configuring Voice Interfaces." Ensure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface. To disable multiple hosts on the port, use the **no dot1x host-mode multi-host** interface configuration command.
Step 5	`Switch(config-if)# switchport voice vlan vlan-id`	(Optional) Configures the voice VLAN.
Step 6	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show dot1x interface interface-id [detail]`	Verifies your entries.
Step 8	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to enable 802.1X authentication and to allow multiple hosts:

```
Switch(config)# interface gigabitethernet2/1
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x host-mode multi-host
Switch(config-if)# end
```

This example shows how to enable MDA and to allow both a host and a 802.1X voice device (e.g., a Cisco or 3rd-party phone with 802.1X supplicant) on the port:

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet3/1
Switch(config-if)# shut
Switch(config-if)# switchport access vlan 12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x host-mode multi-domain
Switch(config-if)# no shut
Switch(config-if)# end
```

This example shows how to enable MDA and to allow both a host and a non-802.1X voice device on the port:

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet3/1
Switch(config-if)# shut
Switch(config-if)# switchport access vlan 12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x host-mode multi-domain
Switch(config-if)# dot1x mac-auth-bypass
Switch(config-if)# no shut
Switch(config-if)# end
```

This example shows how to verify the dot1x MDA settings on interface FastEthernet6/1:

```
Switch# show dot1x interface FastEthernet3/1 detail

Dot1x Info for FastEthernet3/1
-----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = AUTO
ControlDirection          = Both
HostMode                  = MULTI_DOMAIN
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod           = 0

Dot1x Authenticator Client List
-------------------------------
Domain                    = DATA
Supplicant                = 0000.0000.ab01
        Auth SM State     = AUTHENTICATED
        Auth BEND SM Stat = IDLE
Port Status               = AUTHORIZED
Authentication Method     = Dot1x
Authorized By             = Authentication Server
Vlan Policy               = 12

Domain                    = VOICE
Supplicant                = 0060.b057.4687
        Auth SM State     = AUTHENTICATED
        Auth BEND SM Stat = IDLE
Port Status               = AUTHORIZED
Authentication Method     = Dot1x
Authorized By             = Authentication Server

Switch#
```

Configuring RADIUS-Provided Session Timeouts

You can configure the Catalyst 4500 series switch to use a RADIUS-provided reauthentication timeout.

To configure RADIUS-provided timeouts, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode.
Step 3	`Switch(config-if)# switchport mode access`	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	`Switch(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 5	`Switch(config-if)# dot1x timeout reauth-period {interface | server}`	Sets the re-authentication period (seconds).
Step 6	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show dot1x interface interface-id details`	Verifies your entries.
Step 8	`Switch # copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to configure the switch to derive the re-authentication period from the server and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface f7/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x timeout reauth-period server
Switch(config-if)# end
Switch# show dot1x interface f7/1 det

Dot1x Info for FastEthernet7/11
-----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = FORCE_AUTHORIZED
ControlDirection          = Both
HostMode                  = SINGLE_HOST
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = (From Authentication Server)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod           = 0

Dot1x Authenticator Client List Empty

Port Status               = AUTHORIZED

Switch#
```

Enabling 802.1X RADIUS Accounting

Note Supervisor Engine 6-E does *not* support this feature.

To configure 802.1X accounting, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# aaa accounting dot1x default start-stop group radius`	Enables 802.1X accounting, using the list of all RADIUS servers.
Step 3	`Switch(config)# clock timezone PST -8`	Sets the time zone for the accounting event-time stamp field.
Step 4	`Switch(config)# clock calendar-valid`	Enables the date for the accounting event-time stamp field.
Step 5	`Switch(config)# aaa accounting system default start-stop group radius`	(Optional) Enables system accounting (using the list of all RADIUS servers) and generates system accounting reload event messages when the switch reloads.
Step 6	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show running-config`	Verifies your entries.
Step 8	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to specify the server with IP address 172.120.39.46 as the RADIUS server. The first command configures the RADIUS server, specifying port 1612 as the authorization port, 1813 as the UDP port for accounting, and rad123 as the encryption key:

```
Switch# configure terminal
Switch(config)# radius-server host 172.120.39.46 auth-port 1812 acct-port 1813 key rad123
Switch(config)# aaa accounting dot1x default start-stop group radius
Switch(config)# aaa accounting system default start-stop group radius
Switch(config)# end
Switch#
```

Note You must configure the RADIUS server to perform accounting tasks, such as logging start, stop, and interim-update messages and time stamps. To turn on these functions, enable logging of “Update/Watchdog packets from this AAA client” in your RADIUS server Network Configuration tab. Next, enable “CVS RADIUS Accounting” in your RADIUS server System Configuration tab.

Configuring 802.1X with Guest VLANs

Note Supervisor Engine 6-E does *not* support this feature.

You can configure a guest VLAN for each 802.1X port on the Catalyst 4500 series switch to provide limited services to clients, such as downloading the 802.1X client. These clients might be upgrading their system for 802.1X authentication, and some hosts, such as Windows 98 systems, might not be 802.1X-capable.

When you enable a guest VLAN on an 802.1X port, the Catalyst 4500 series switch assigns clients to a guest VLAN provided (1) the authentication server does not receive a response to its EAPOL request or identity frame, or (2) the EAPOL packets are not sent by the client.

Starting with Cisco IOS Release 12.2(25)EWA, the Catalyst 4500 series switch maintains the EAPOL packet history. If another EAPOL packet is detected on the interface during the lifetime of the link, network access is denied. The EAPOL history is reset upon loss of the link.

Any number of 802.1X-incapable clients are allowed access when the switch port is moved to the guest VLAN. If an 802.1X-capable client joins the same port on which the guest VLAN is configured, the port is put into the unauthorized state in the user-configured access VLAN, and authentication is restarted.

Guest VLANs are supported on 802.1X ports in single-host or multiple-hosts mode.

Note When a port is put into a guest VLAN, it is automatically placed into multihost mode, and an unlimited number of hosts can connect through the port. Changing the multihost configuration does not effect a port in a guest VLAN.

Note Except for an RSPAN VLAN or a voice VLAN, you can configure any active VLAN as an 802.1X guest VLAN.

To configure 802.1X with guest VLAN on a port, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.

Step 3	`Switch(config-if)#` **`switchport mode access`**
	`or`
	`Switch(config-if)#` **`switchport mode private-vlan host`**
		Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
		Specifies that the ports with a valid PVLAN trunk association become active host private VLAN trunk ports.

Step 4	`Switch(config-if)#` **`dot1x pae authenticator`**	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 5	`Switch(config-if)#` **`dot1x guest-vlan`** *`vlan-id`*	Enables a guest VLAN on a particular interface. To disable the guest VLAN feature on a particular port, use the **no dot1x guest-vlan** interface configuration command.
Step 6	`Switch(config-if)#` **`dot1x port-control auto`**	Enables 802.1X authentication on the interface.
Step 7	`Switch(config-if)#` **`end`**	Returns to configuration mode.
Step 8	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.

This example shows how to enable a regular VLAN 50 on Fast Ethernet 4/3 as a guest VLAN on a static access port:

```
Switch# configure terminal
Switch(config)# interface fa4/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x guest-vlan 50
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

This example shows how to enable a secondary private VLAN 100 as a guest VLAN on a private VLAN host port:

```
Switch# configure terminal
Switch(config)# interface fa4/3
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x guest-vlan 100
Switch(config-if)# end
Switch#
```

To enable supplicants to be allowed into guest VLAN on a switch, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch# dot1x guest-vlan supplicant`	(Optional) Enables supplicants to be allowed into the guest VLANs globally on the switch. **Note** Although not visible in the CLI for Cisco IOS Release 12.3(31)SG, legacy configurations that include the **dot1x guest-vlan supplicant** command still work. However, use of this command is not recommended because the authentication failed VLAN option obviates the need for this command. To disable the supplicant guest VLAN feature on a switch, use the **no dot1x guest-vlan supplicant** global configuration command.
Step 3	`Switch(config)# interface` *`interface-id`*	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.

Step 4	`Switch(config-if)# switchport mode access`
	or
	`Switch(config-if)# switchport mode private-vlan host`
		Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
		Specifies that the ports with a valid PVLAN trunk association become active host private VLAN trunk ports.

Step 5	`Switch(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 6	`Switch(config-if)# dot1x guest-vlan` *`vlan-id`*	Specifies an active VLAN as an 802.1X guest VLAN. The range is 1 to 4094.
Step 7	`Switch(config-if)# dot1x port-control auto`	Enables 802.1X authentication on the interface.
Step 8	`Switch(config-if)# end`	Returns to privileged EXEC mode.

	Command	Purpose
Step 9	`Switch#` **`show dot1x interface`** *`interface-id`*	Verifies your entries.
Step 10	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

This example shows how to enable the guest VLAN feature and to specify VLAN 5 as a guest VLAN:

```
Switch# configure terminal
Switch(config)# dot1x guest-vlan supplicant
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x guest-vlan 5
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Configuring 802.1X with MAC Authentication Bypass

Note Supervisor Engine 6-E does *not* support this feature.

To enable MAB, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Specifies the port to be configured, and enters interface configuration mode.

Step 3	`Switch(config-if)#` **`switchport mode access`**
	`or`
	`Switch(config-if)#` **`switchport mode private-vlan host`**
		Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
		Specifies that the ports with a valid PVLAN trunk association become active host private VLAN trunk ports.

Step 4	`Switch(config-if)#` **`dot1x pae authenticator`**	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 5	`Switch(config-if)#` **`dot1x port-control auto`**	Enables 802.1X authentication on the interface.
Step 6	`Switch(config-if)#` **`dot1x mac-auth-bypass`** `[`**`eap`**`]`	Enables MAB on a switch.
Step 7	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 8	`Switch#` **`show dot1x interface`** *`interface-id`* **`details`**	(Optional) Verifies your entries.
Step 9	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Note Removing a 802.1X MAB configuration from a port does not impact the authorized/authenticated state of the port. If the port is in an unauthenticated state, it remains in that state. If the port is in an authenticated state because of MAB, the switch reverts to the 802.1X Authenticator. If the port was already authorized with a MAC address and the MAB configuration was removed, the port remains in an authorized state until re-authentication occurs. At that time, if an 802.1X supplicant is detected on the wire, the MAC address is removed.

This example shows how to enable MAB on Gigabit Ethernet interface 3/3 and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet3/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x mac-auth-bypass
Switch(config-if)# end
Switch# show dot1x int g3/3 details
Dot1x Info for GigabitEthernet3/3
-----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = AUTO
ControlDirection          = Both
HostMode                  = SINGLE_HOST
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 1
RateLimitPeriod           = 0
Mac-Auth-Bypass           = Enabled

Dot1x Authenticator Client List
-------------------------------
Supplicant                = 0000.0000.0001
 Auth SM State     = AUTHENTICATED
 Auth BEND SM Stat = IDLE
Port Status               = AUTHORIZED
Authentication Method     = MAB
Authorized By             = Authentication Server
Vlan Policy               = N/A

Switch#
```

Configuring 802.1X with Inaccessible Authentication Bypass

Note Supervisor Engine 6-E does *not* support this feature.

Caution You must configure the switch to monitor the state of the RADIUS server as described in the section Configuring Switch-to-RADIUS-Server Communication, page 34-26 for Inaccessible Authentication Bypass to work properly. Specifically, you must configure the RADIUS test username, idle-time, deadtime and dead-criteria. Failure to do so results in the switch failing to detect that the RADIUS server has gone down, or prematurely marking a dead RADIUS server as alive again.

To configure a port as a critical port and to enable the Inaccessible Authentication Bypass feature, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`dot1x critical eapol`**	(Optional) Configures whether to send an EAPOL-Success packet when a port is critically-authorized partway through an EAP exchange. **Note** Some supplicants require this. The default is not to send EAPOL-Success packets when ports are critically-authorized.
Step 3	`Switch(config)#` **`dot1x critical recovery delay`** *`msec`*	(Optional) Specifies a throttle rate for the reinitialization of critically-authorized ports when the RADIUS server becomes available. The default throttle rate is 100 milliseconds. This means that 10 ports reinitialize per second.
Step 4	`Switch(config)#` **`interface`** *`interface-id`*	Specifies the port to be configured, and enters interface configuration mode.

Step 5	`Switch(config-if)#` **`switchport mode access`**
	`or`
	`Switch(config-if)#` **`switchport mode private-vlan host`**
		Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
		Specifies that the ports with a valid PVLAN trunk association become active host private VLAN trunk ports.

Step 6	`Switch(config-if)#` **`dot1x pae authenticator`**	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 7	`Switch(config-if)#` **`dot1x port-control auto`**	Enables 802.1X authentication on the interface.
Step 8	`Switch(config-if)#` **`dot1x critical`**	Enables the Inaccessible Authentication Bypass feature on the port. To disable the feature, use the **no dot1x critical** interface configuration command.
Step 9	`Switch(config-if)#` **`dot1x critical vlan`** *`vlan`*	(Optional) Specifies a VLAN into which the port is assigned when it is critically authorized. **Note** Supervisor Engine 6-E does not support this feature. The default is to use the configured VLAN on the port.
Step 10	`Switch(config-if)#` **`dot1x critical recovery action reinitialize`**	(Optional) Specifies that the port should be reinitialized if it is critically authorized and RADIUS becomes available. The default is not to reinitialize the port.
Step 11	`Switch(config)#` **`end`**	Return to privileged EXEC mode.

	Command	Purpose
Step 12	Switch# **show dot1x interface** *interface-id* **details**	(Optional) Verify your entries.
Step 13	Switch# **copy running-config startup-config**	(Optional) Save your entries in the configuration file.

The following example shows a full configuration of 802.1X with Inaccessible Authentication Bypass, including required AAA and RADIUS configuration as specified in the "Enabling 802.1X Authentication" section on page 34-23and "Configuring Switch-to-RADIUS-Server Communication" section on page 34-26.

The RADIUS server configured is at IP address 10.1.2.3, using port 1812 for authentication and 1813 for accounting. The RADIUS secret key is *mykey*. The username used for the test server probes is *randomuser*. The test probes for both living and dead servers are generated once per minute. The interface FastEthernet 3/1 is configured to critically authenticate into VLAN 17 when AAA becomes unresponsive, and to reinitialize automatically when AAA becomes available again.

```
Switch# configure terminal
Switch(config)# aaa new-model
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# dot1x system-auth-control
Switch(config)# radius-server host 10.1.2.3 auth-port 1812 acct-port 1813 test username
randomuser idle-time 1 key mykey
Switch(config)# radius deadtime 1
Switch(config)# radius dead-criteria time 15 tries 3
Switch(config)# interface f3/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x critical
Switch(config-if)# dot1x critical vlan 17
Switch(config-if)# dot1x critical recovery action reinitialize
Switch(config-if)# end
Switch# show dot1x int fastethernet 3/1 det

Dot1x Info for FastEthernet3/1
-----------------------------------
PAE                       = AUTHENTICATOR
PortControl               = AUTO
ControlDirection          = Both
HostMode                  = SINGLE_HOST
ReAuthentication          = Disabled
QuietPeriod               = 60
ServerTimeout             = 30
SuppTimeout               = 30
ReAuthPeriod              = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod           = 0
Critical-Auth             = Enabled
Critical Recovery Action  = Reinitialize
Critical-Auth VLAN        = 17

Dot1x Authenticator Client List
-------------------------------
Supplicant                = 0000.0000.0001

Auth SM State      = AUTHENTICATING
Auth BEND SM Stat = RESPONSE
Port Status               = AUTHORIZED
```

```
Authentication Method      = Dot1x
Authorized By              = Critical-Auth
Operational HostMode       = SINGLE_HOST
Vlan Policy                = 17

Switch#
```

Configuring 802.1X with Unidirectional Controlled Port

Note Supervisor Engine 6-E does *not* support this feature.

To configure unidirectional controlled port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Specifies the port to be configured, and enter interface configuration mode.

Step 3	`Switch(config-if)# switchport mode access`
	or
	`Switch(config-if)# switchport mode private-vlan host`
		Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
		Specifies that the ports with a valid PVLAN trunk association become active host private VLAN trunk ports.

Step 4	`Switch(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 5	`Switch(config-if)# dot1x control-direction {in | both}`	Enables unidirectional port control on a per-port basis.
Step 6	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show dot1x interface interface-id details`	(Optional) Verifies your entries.
Step 8	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to enable unidirectional port control:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet3/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x control-direction in
Switch(config-if)# end
Switch# show dot1x int g3/3
Dot1x Info for GigabitEthernet3/3
-----------------------------------
PAE                        = AUTHENTICATOR
PortControl                = AUTO
ControlDirection           = In (Inactive)
HostMode                   = SINGLE_HOST
ReAuthentication           = Disabled
QuietPeriod                = 60
ServerTimeout              = 30
SuppTimeout                = 30
ReAuthPeriod               = 3600 (Locally configured)
```

```
ReAuthMax                   = 2
MaxReq                      = 2
TxPeriod                    = 30
RateLimitPeriod             = 0

Switch#
```

Configuring 802.1X with Authentication Failed VLAN Assignment

Note Supervisor Engine 6-E does *not* support this feature.

By configuring authentication-failed VLAN alignment on any Layer 2 port on the Catalyst 4500 series switch, you can provide limited network services to clients that fail the authentication process.

Note You can use authentication-failed VLAN assignment with other security features, such as Dynamic ARP Inspection (DAI), Dynamic Host Configuration Protocol (DHCP) snooping, and IP Source Guard. Each of these features can be enabled and disabled independently on the authentication-failed VLAN.

Note You cannot configure an authentication-failed VLAN and a voice VLAN on the same port. When you try to configure these two features on the same port, a syslog message is generated.

To configure 802.1X with authentication-failed VLAN assignment, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 3	`Switch(config-if)#` **`switchport mode access`**	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	`Switch(config-if)#` **`dot1x port-control auto`**	Enables 802.1X authentication on the interface.
Step 5	`Switch(config-if)#` **`dot1x auth-fail vlan`** *`vlan-id`*	Enables authentication-failed VLAN on a particular interface. To disable the authentication-failed VLAN feature on a particular port, use the **no dot1x auth-fail vlan** interface configuration command.
Step 6	`Switch(config-if)#` **`dot1x auth-fail max-attempts`** *`max-attemtps`*	Configure a maximum number of attempts before the port is moved to authentication-failed VLAN. Default is 3 attempts.
Step 7	`Switch(config-if)#` **`end`**	Returns to configuration mode.
Step 8	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 9	`Switch#` **`show dot1x interface`** *`interface-id`* **`details`**	(Optional) Verifies your entries.
Step 10	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

This example shows how to enable a regular VLAN 40 on Fast Ethernet 4/3 as a authentication-failed VLAN on a static access port:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet3/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x auth-fail vlan 40
Switch(config-if)# dot1x auth-fail max-attempts 5
Switch(config-if)# end
Switch(config)# end
Switch# show dot1x all
Dot1x Info for interface GigabitEthernet3/1
----------------------------------------------------
PortStatus        = AUTHORIZED(AUTH-FAIL-VLAN)
MaxReq            = 2
MaxAuthReq        = 2
HostMode          = Single(AUTH-FAIL-VLAN)
PortControl       = Auto
QuietPeriod       = 60 Seconds
Re-authentication = Disabled
ReAuthPeriod      = 3600 Seconds
ServerTimeout     = 30 Seconds
SuppTimeout       = 30 Seconds
TxPeriod          = 30 Seconds
Guest-Vlan        = 6
Switch
```

Configuring 802.1X with Voice VLAN

Note Supervisor Engine 6-E does *not* support this feature.

Note You must configure 802.1X and voice VLAN at the same time.

Note You cannot configure an authentication-failed VLAN and a voice VLAN on the same port. When you try to configure these two features on the same port, a syslog message is generated.

To enable 802.1X with voice VLAN, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode.
Step 3	`Switch(config-if)# switchport access vlan vlan-id`	Sets the VLAN for a switched interface in access mode.
Step 4	`Switch(config-if)# switchport mode access`	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 5	`Switch(config-if)# switchport voice vlan vlan-id`	Sets the voice VLAN for the interface.

	Command	Purpose
Step 6	Switch(config-if)# **dot1x pae authenticator**	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 7	Switch(config-if)# **dot1x port-control auto**	Enables 802.1X authentication on the interface.
Step 8	Switch(config-if)# **end**	Returns to configuration mode.
Step 9	Switch(config)# **end**	Returns to privileged EXEC mode.
Step 10	Switch# **show dot1x interface** *interface-id* **details**	(Optional) Verifies your entries.
Step 11	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to enable 802.1X with voice VLAN feature on Fast Ethernet interface 5/9:

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport access vlan 2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch(config# end
Switch#
```

Enabling Periodic Reauthentication

You can enable periodic 802.1X client reauthentication and specify how often it occurs. If you do not specify a time value before enabling reauthentication, the interval between reauthentication attempts is 3600 seconds.

Automatic 802.1X client reauthentication is a per-interface setting and can be set for clients connected to individual ports. To manually reauthenticate the client connected to a specific port, see the "Changing the Quiet Period" section on page 34-44.

To enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface** *interface-id*	Enters interface configuration mode and specifies the interface to be enabled for periodic reauthentication.
Step 3	Switch(config-if)# **switchport mode access**	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	Switch(config-if)# **dot1x pae authenticator**	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.

	Command	Purpose
Step 5	`Switch(config-if)# dot1x re-authentication`	Enables periodic reauthentication of the client, which is disabled by default. To disable periodic reauthentication, use the **no dot1x re-authentication** interface configuration command.
Step 6	`Switch(config-if)# dot1x timeout reauth-period {seconds | server}`	Specifies the number of seconds between reauthentication attempts or have the switch use a RADIUS-provided session timeout. The range is 1 to 65,535; the default is 3600 seconds. To return to the default number of seconds between reauthentication attempts, use the **no dot1x timeout reauth-period** global configuration command. This command affects the behavior of the switch only if periodic reauthentication is enabled.
Step 7	`Switch(config-if)# dot1x port-control auto`	Enables 802.1X authentication on the interface.
Step 8	`Switch(config-if)# end`	Returns to privileged EXEC mode.

This example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000:

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x re-authentication
Switch(config-if)# dot1x timeout reauth-period 4000
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Enabling Multiple Hosts

You can attach multiple hosts (clients) to a single 802.1X-enabled port as shown in Figure 34-7 on page 34-21. In this mode, when the port is authorized, all other hosts that are 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 for all wireless access point-attached clients.

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	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode and specifies the interface to which multiple hosts are indirectly attached.
Step 3	`Switch(config-if)# switchport mode access`	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.

	Command	Purpose
Step 4	`Switch(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.
Step 5	`Switch(config-if)# dot1x host-mode multiple-hosts`	Allows multiple hosts (clients) on an 802.1X-authorized port. **Note** Ensure that the **dot1x port-control** interface configuration command set is set to **auto** for the specified interface. To disable multiple hosts on the port, use the **no dot1x host-mode multiple-hosts** interface configuration command.
Step 6	`Switch(config-if)# dot1x port-control auto`	Enables 802.1X authentication on the interface.
Step 7	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 8	`Switch# show dot1x all interface interface-id`	Verifies your entries.
Step 9	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to enable 802.1X on Fast Ethernet interface 0/1 and to allow multiple hosts:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x host-mode multiple-hosts
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

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 number smaller than the default.

To change the quiet period, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode and specifies the interface to be enabled for timeout **quiet-period**.
Step 3	`Switch(config-if)# switchport mode access`	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	`Switch(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.

	Command	Purpose
Step 5	`Switch(config-if)# dot1x timeout quiet-period seconds`	Sets the number of seconds that the switch remains in the **quiet-period** following a failed authentication exchange with the client. To return to the default quiet-period, use the **no dot1x timeout quiet-period** configuration command. The range is 0 to 65,535 seconds; the default is 60.
Step 6	`Switch(config-if)# dot1x port-control auto`	Enables 802.1X authentication on the interface.
Step 7	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 8	`Switch# show dot1x all`	Verifies your entries.
Step 9	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to set the **quiet-period** on the switch to 30 seconds:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x timeout quiet-period 30
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

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	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode and specifies the interface to be enabled for timeout tx-period.
Step 3	`Switch(config-if)# switchport mode access`	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	`Switch(config-if)# dot1x pae authenticator`	Enables 802.1X authentication on the port with default parameters. Refer to the "Default 802.1X Configuration" section on page 34-22.

	Command	Purpose
Step 5	`Switch(config-if)# `**`dot1x timeout tx-period`** *`seconds`*	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request. The range is 1 to 65,535 seconds; the default is 30. To return to the default retransmission time, use the **no dot1x timeout tx-period** interface configuration command.
Step 6	`Switch(config-if)# `**`dot1x port-control auto`**	Enables 802.1X authentication on the interface.
Step 7	`Switch(config-if)# `**`end`**	Returns to privileged EXEC mode.
Step 8	`Switch# `**`show dot1x all`**	Verifies your entries.
Step 9	`Switch# `**`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

This example shows how to set the retransmission time to 60 seconds:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x timeout tx-period 60
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Setting the Switch-to-Client Frame-Retransmission Number

In addition to changing the switch-to-client retransmission times, you can change the number of times that the switch sends EAP-Request/Identity and other EAP-Request frames to the client before restarting the authentication process. The number of EAP-Request/Identity retransmissions is controlled by the **dot1x max-reauth-req** command; the number of retransmissions for other EAP-Request frames is controlled by the **dot1x max-req** command.

Note You should change the default values of these commands 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 numbers, perform this task:

	Command	Purpose
Step 1	`Switch# `**`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)# `**`interface`** *`interface-id`*	Enters interface configuration mode and specifies the interface to be enabled for **max-reauth-req** and/or **max-req**.
Step 3	`Switch(config-if)# `**`switchport mode access`**	Specifies a non-trunking, nontagged single VLAN Layer 2 interface.
Step 4	`Switch(config-if)# `**`dot1x pae authenticator`**	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 34-22.

	Command	Purpose

Step 5	`Switch(config-if)# dot1x max-req count`
	or
	`Switch(config-if)# dot1x max-reauth-req count`
		Specifies the number of times EAPOL DATA packets are re-transmitted (if lost, or not replied to). For example, if you have a supplicant in the midst of authenticating and it experiences a problem, the authenticator will re-transmit requests for data 3 times before giving up on the authentication request. The range for *count* is 1 to 10; the default is 2.
		Specifies the timer for EAPOL-Identity-Request frames (only). If you plug in a device incapable of 802.1X, 3 EAPOL-Id-Req frames will go out on the wire before the state machine resets. Alternatively, if you have configured Guest-VLAN, 3 frames will go out on the wire before the port is enabled. This parameter has a default value of 2.The range for *count* is 1 to 10; the default is 2.
		To return to the default retransmission number, use the **no dot1x max-req** and **no dot1x max-reauth-req** global configuration command.

Step 6	`Switch(config-if)# dot1x port-control auto`	Enables 802.1X authentication on the interface.
Step 7	`Switch(config-if)# end`	Returns to privileged EXEC mode.
Step 8	`Switch# show dot1x all`	Verifies your entries.
Step 9	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to set 5 as the number of times that the switch retransmits an EAP-request/identity request before restarting the authentication process:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x max-reauth-req 5
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Manually Reauthenticating a Client Connected to a Port

You can manually reauthenticate a client connected to a specific port at any time by entering the **dot1x re-authenticate interface** privileged EXEC command. If you want to enable or disable periodic reauthentication, see the “Enabling Periodic Reauthentication” section on page 34-42.

This example shows how to manually reauthenticate the client connected to Fast Ethernet port 1/1:

```
Switch# dot1x re-authenticate interface fastethernet1/1
Starting reauthentication on FastEthernet1/1
```

Initializing the 802.1X Authentication State

The **dot1x initialize** command causes the authentication process to be restarted irrespective of the state it is in currently.

This example shows how to restart the authentication process on Fast Ethernet port 1/1:

```
Switch# dot1x initialize interface fastethernet1/1
```

This example shows how to restart the authentication process on all ports of the switch:

```
Switch# dot1x initialize
```

Removing 802.1X Client Information

The **clear dot1x** command causes all existing supplicants to be completely deleted from an interface or from all the interfaces on a switch.

This example shows how to remove 802.1X client information on Fast Ethernet port 1/1:

```
Switch# clear dot1x interface fastethernet1/1
```

This example shows how to remove 802.1X client information on all ports of the switch:

```
Switch# clear dot1x all
```

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	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# dot1x default`	Resets the configurable 802.1X parameters to the default values.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show dot1x all`	Verifies your entries.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Displaying 802.1X Statistics and Status

To display 802.1X statistics for all interfaces, use the **show dot1x all statistics** privileged EXEC command.

To display the 802.1X administrative and operational status for the switch, use the **show dot1x all details** privileged EXEC command. To display the 802.1X administrative and operational status for a specific interface, use the **show dot1x interface details** privileged EXEC command.

CHAPTER 35

Configuring Port Security

This chapter describes how to configure port security on the Catalyst 4500 series switch. It provides an overview of port security on the Catalyst 4500 series switch and details the configuration on various types of ports such as access, voice, trunk and private VLAN.

This chapter consists of these sections:

For information on how to troubleshoot Port Security, refer to the "Troubleshooting Port Security" section on page 51-35.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Command List

This table lists the commands most commonly used with Port Security.

Command	Purpose	Navigation
errdisable recovery cause psecure-violation	Brings a secure port out of error-disabled state	Violation Actions, page 35-6
errdisable recovery interval	Customizes the time to recover from a specified error disable cause	Violation Actions, page 35-6

Command	Purpose	Navigation
port-security mac-address	Configures all secure MAC addresses on each VLAN	Secure MAC Addresses, page 35-3
port-security maximum	Configures a maximum number of MAC addresses on an interface	Configuring Port Security on Access Ports, page 35-7
private-vlan association add	Creates an association between a secondary VLAN and a primary VLAN	Example of Port Security on an Isolated Private VLAN Host Port, page 35-15
private-vlan isolated	Designates the VLAN as a private VLAN	Configuring Port Security on an Isolated Private VLAN Host Port, page 35-14
private-vlan primary	Specifies the VLAN as the primary private VLAN	Configuring Port Security on an Isolated Private VLAN Host Port, page 35-14
switchport mode private-vlan host	Specifies that ports with valid private VLAN trunk association become active host private VLAN trunk ports	Configuring Port Security on an Isolated Private VLAN Host Port, page 35-14
switchport private-vlan host-association	Defines a host association on an isolated host port	Configuring Port Security on an Isolated Private VLAN Host Port, page 35-14
switchport private-vlan mapping	Defines a private VLAN for the promiscuous ports	Configuring Port Security on an Isolated Private VLAN Host Port, page 35-14
switchport port-security	Enables port security	Configuring Port Security on Access Ports, page 35-7
switchport port-security aging static	Configures static aging of MAC address.	Aging Secure MAC Addresses, page 35-5
switchport port-security aging time	Specifies an aging time for a port	Example 3: Setting the Aging Timer, page 35-11
switchport port-security limit rate invalid-source-mac	Sets the rate limit for bad packets	Example 7: Setting a Rate Limit for Bad Packets, page 35-13
switchport port-security mac-address	Configures a secure MAC address for an interface	Example 5: Configuring a Secure MAC Address, page 35-11
switchport port-security mac-address <*mac_address*> **sticky**	Specifies the sticky MAC address for an interface	Configuring Port Security on Access Ports, page 35-7
switchport port-security mac-address sticky	Enables sticky Port Security	Sticky Addresses on a Port, page 35-5
no switchport port-security mac-address sticky	Converts a sticky secure MAC address to a dynamic MAC secure address	Configuring Port Security on Access Ports, page 35-7
switchport port-security maximum	Sets the maximum number of secure MAC addresses for an interface	Example 1: Setting Maximum Number of Secure Addresses, page 35-10
switchport port-security violation	Sets the violation mode	Example 2: Setting a Violation Mode, page 35-10

Command	Purpose	Navigation
no switchport port-security violation	Sets the violation mode	Configuring Port Security on Access Ports, page 35-7
switchport trunk encapsulation dot1q	Sets the encapsulation mode to dot1q	Example 1: Configuring a Maximum Limit of Secure MAC Addresses for all VLANs, page 35-18

Overview of Port Security

Port security enables you to restrict the number of MAC addresses (termed *secure MAC addresses*) on a port, allowing you to prevent access by unauthorized MAC addresses. It also allows you to configure a maximum number of secure MAC addresses on a given port (and optionally for a VLAN for trunk ports). When a secure port exceeds the maximum, a security violation is triggered, and a violation action is performed based on the violation action mode configured on the port.

If you configure the maximum number of secure MAC addresses as 1 on the port, the device attached to the secure port is assured sole access to the port.

If a secure MAC address is secured on a port, that MAC address is not allowed to enter on any other port off that VLAN. If it does, the packet is dropped unnoticed in the hardware. Other than through the interface or port counters, you do not receive a log message reflecting this fact. Be aware that this condition does not trigger a violation. Dropping these packets in the hardware is more efficient and can be done without putting additional load on the CPU.

Port Security has the following characteristics:

- It allows you to age out secure MAC addresses. Two types of aging are supported: inactivity and absolute.
- It supports a sticky feature whereby the secure MAC addresses on a port are retained through switch reboots and link flaps.
- It can be configured on various types of ports such as access, voice, trunk, EtherChannel, and private VLAN ports.

This overview contains the following topics:

- Secure MAC Addresses, page 35-3
- Maximum Number of Secure MAC Addresses, page 35-4
- Aging Secure MAC Addresses, page 35-5
- Sticky Addresses on a Port, page 35-5
- Violation Actions, page 35-6

Secure MAC Addresses

Port Security supports the following types of secure MAC addresses:

- Dynamic or Learned—Dynamic secure MAC addresses are learned when packets are received from the host on the secure port. You might want to use this type if the user’s MAC address is not fixed (laptop).
- Static or Configured—Static secure MAC addresses are configured by the user through CLI or SNMP. You might want to use this type if your MAC address remains fixed (PC).

- Sticky—Sticky addresses are learned like dynamic secure MAC addresses, but persist through switch reboots and link flaps like static secure MAC addresses. You might want to use this type if a large number of fixed MAC addresses exist and you do not want to configure MAC addresses manually (100 PCs secured on their own ports).

If a port has reached its maximum number of secure MAC addresses and you try to configure a static secure MAC address, your configuration is rejected and an error message displays. If a port has reached its maximum number of secure MAC addresses and a new dynamic secure MAC address is added, a violation action is triggered.

You can clear dynamic secure MAC addresses with the **clear port-security** command. You can clear sticky and static secure MAC addresses one at a time with the **no** form of the **switchport port-security mac-address** command.

Maximum Number of Secure MAC Addresses

A secure port has a default of one MAC address. You can change the default to any value between 1 and 3,000. The upper limit of 3,000 guarantees one MAC address per port and an additional 3,000 across all ports in the system.

After you have set the maximum number of secure MAC addresses on a port, you can include the secure addresses in an address table in one of the following ways:

- You can configure the secure MAC addresses with the **switchport port-security mac-address** *mac_address* interface configuration command.
- You can configure all secure MAC addresses on a range of VLANs with the **port-security mac-address** VLAN range configuration command for trunk ports.
- You can allow the port to dynamically configure secure MAC addresses with the MAC addresses of connected devices.
- You can configure some of the addresses and allow the rest to be dynamically configured.

Note If a port's link goes down, all dynamically secured addresses on that port are no longer secure.

- You can configure MAC addresses to be sticky. These can be dynamically learned or manually configured, stored in the address table, and added to the running configuration. After these addresses are saved in the configuration file, the interface does not need to dynamically relearn them when the switch restarts. Although you can manually configure sticky secure addresses, this action is not recommended.

Note On a trunk port, a maximum number of secure MAC addresses can be configured on both the port and port VLAN. The port's maximum value can be greater than or equal to the port VLAN maximum(s) but not less than the port VLAN maximum(s). If the port's maximum value is less than at least one of the port VLAN's maximum (i.e. if we have max set to 3 on VLAN 10 while no "sw port max" is set (defaults to 1)), the port shuts down when dynamic adds reaches 2 on VLAN 10 (see "Port Security Guidelines and Restrictions" on page 31). The port VLAN maximum enforces the maximum allowed on a given port on a given VLAN. If the maximum is exceeded on a given VLAN but the port's maximum is not exceeded, the port still shuts down. The entire port is shut down even if one of the VLANs on the port has actually caused the violation.

Aging Secure MAC Addresses

You might want to age secure MAC addresses when the switch may be receiving more than 3,000 MAC addresses ingress.

Note Aging of sticky addresses is not supported.

By default, port security does not age out the secure MAC addresses. After learned, the MAC addresses remain on the port until either the switch reboots or the link goes down (unless the sticky feature is enabled). However, port security does allow you to configure aging based on the absolute or inactivity mode and aging interval (in minutes, from 1 to n).

- Absolute mode: ages between n and n+1
- Inactivity mode: ages between n+1 and n+2

Use this feature to remove and add PCs on a secure port without manually deleting the existing secure MAC addresses, while still limiting the number of secure addresses on a port.

Unless static aging is explicitly configured with the **switchport port-security aging static** command, static addresses are not aged even if aging is configured on the port.

Note The aging increment is one minute.

Sticky Addresses on a Port

By enabling *sticky* port security, you can configure an interface to convert the dynamic MAC addresses to sticky secure MAC addresses and to add them to the running configuration. You might want to do this if you do not expect the user to move to another port, and you want to avoid statically configuring a MAC address on every port.

Note If you use a different chassis, you might need another MAC address.

To enable sticky port security, enter the **switchport port-security mac-address sticky** command. When you enter this command, the interface converts all the dynamic secure MAC addresses, including those that were dynamically learned before sticky learning was enabled, to sticky secure MAC addresses.

The sticky secure MAC addresses do not automatically become part of the configuration file, which is the startup configuration used each time the switch restarts. If you save the running config file to the configuration file, the interface does not need to relearn these addresses when the switch restarts. If you do not save the configuration, they are lost.

If sticky port security is disabled, the sticky secure MAC addresses are converted to dynamic secure addresses and are removed from the running configuration.

After the maximum number of secure MAC addresses is configured, they are stored in an address table. To ensure that an attached device has sole access of the port, configure the MAC address of the attached device and set the maximum number of addresses to one, which is the default.

A security violation occurs if the maximum number of secure MAC addresses to a port has been added to the address table and a workstation whose MAC address is not in the address table attempts to access the interface.

Violation Actions

A security violation is triggered when the number of secure MAC addresses on the port exceeds the maximum number of secure MAC addresses allowed on the port.

Note A secure violation is not triggered if the host secured on one port shows up on another port. The Catalyst 4500 series switch drops such packets on the new port silently in the hardware and does not overload the CPU.

You can configure the interface for one of following violation modes, which are based on the response to the violation:

- Restrict—A port security violation restricts data (that is, packets are dropped in software), causes the SecurityViolation counter to increment, and causes an SNMP Notification to be generated. You might want to configure this mode in order to provide uninterrupted service/access on a secure port.

  The rate at which SNMP traps are generated can be controlled by the **snmp-server enable traps port-security trap-rate** command. The default value ("0") causes an SNMP trap to be generated for every security violation.

- Shutdown—A port security violation causes the interface to shut down immediately. You might want to configure this mode in a highly secure environment, where you do not want unsecured MAC addresses to be denied in software and service interruption is not an issue.

  When a secure port is in the error-disabled state, you can bring it out of this state automatically by configuring the **errdisable recovery cause psecure-violation** global configuration command or you can manually reenable it by entering the **shutdown** and **no shut down** interface configuration commands. This is the default mode.

  You can also customize the time to recover from the specified error disable cause (default is 300 seconds) by entering the **errdisable recovery interval** *interval* command.

Invalid Packet Handling

You might want to rate limit invalid source MAC address packets on a secure port if you anticipate that a device will send invalid packets (such as traffic generator, sniffer, and bad NICs). Port security considers packets with all zero MAC addresses, as well as multicast or broadcast source MAC address, as invalid packets. You can chose to rate limit these packets, and if the rate is exceeded, trigger a violation action for the port.

Port Security on Access Ports

These sections describe how to configure port security:

- Configuring Port Security on Access Ports, page 35-7
- Examples, page 35-10

Note Port security can be enabled on a Layer 2 port channel interface configured in access mode. The port security configuration on an EtherChannel is kept independent of the configuration of any physical member ports.

Configuring Port Security on Access Ports

To restrict traffic through a port by limiting and identifying MAC addresses of the stations allowed to the port, perform this task:

	Command	Purpose

Step 1	`Switch(config)# interface interface_id`
	`interface port-channel port_channel_number`
		Enters interface configuration mode and specifies the interface to configure.
		Note The interface can be a Layer 2 port channel logical interface.

Step 2	`Switch(config-if)# switchport mode access`	Sets the interface mode. **Note** An interface in the default mode (dynamic desirable) cannot be configured as a secure port.
Step 3	`Switch(config-if)# [no] switchport port-security`	Enables port security on the interface. To return the interface to the default condition as nonsecure port, use the **no switchport port-security** command.
Step 4	`Switch(config-if)# [no] switchport port-security maximum value`	(Optional) Sets the maximum number of secure MAC addresses for the interface. The range is 1 to 3072; the default is 1. To return the interface to the default number of secure MAC addresses, use the no **switchport port-security maximum** *value*.

	Command	Purpose (continued)
Step 5	Switch(config-if)# **switchport port-security** [**aging** {**static** | **time** *aging_time* | **type** {**absolute** | **inactivity**}]	Sets the aging time and aging type for all secure addresses on a port. Use this feature to remove and add PCs on a secure port without manually deleting the existing secure MAC addresses while still limiting the number of secure addresses on a port. The **static** keyword enables aging for statically configured secure addresses on this port. The **time** *aging_tim*e keyword specifies the aging time for this port. Valid range for aging_time is from 0 to 1440 minutes. If the time is equal to 0, aging is disabled for this port. The **type** keyword sets the aging type as **absolute** or **inactive**. • **absolute**—All the secure addresses on this port ago out exactly after the time (minutes) specified and are removed from the secure address list. • **inactive**—The secure addresses on this port age out only if there is no data traffic from the secure source address for the specified time period. To disable port security aging for all secure addresses on a port, use the **no switchport port-security aging time** interface configuration command.
Step 6	Switch(config-if)# [**no**] **switchport port-security violation** {**restrict** | **shutdown**}	(Optional) Sets the violation mode, the action to be taken when a security violation is detected, as one of these: • **restrict**—A port security violation restricts data and causes the SecurityViolation counter to increment and send an SNMP trap notification. • **shutdown**—The interface is error-disabled when a security violation occurs. **Note** When a secure port is in the error-disabled state, you can bring it out of this state by entering the **errdisable recovery cause psecure-violation** global configuration command or you can manually reenable it by entering the **shutdown** and **no shut down** interface configuration commands. To return the violation mode to the default condition (shutdown mode), use the **no switchport port-security violation shutdown** command.
Step 7	Switch(config-if)# **switchport port-security limit rate invalid-source-mac** *packets_per_sec*	Sets the rate limit for bad packets. Default is 10 pps.

	Command	Purpose (continued)
Step 8	Switch(config-if)# [**no**] **switchport port-security mac-address** *mac_address*	(Optional) Enters a secure MAC address for the interface. You can use this command to configure a secure MAC addresses. If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are dynamically learned. To delete a MAC address from the address table, use the **no switchport port-security mac-address** *mac_address* command. **Note** This command only applies to access, PVLAN host, and PVLAN promiscuous mode. For more details on PVLAN, trunk, or regular trunk mode, refer to the "Port Security on Trunk Ports" section on page 35-16.
Step 9	Switch(config-if)# [**no**] **switchport port-security mac-address sticky**	(Optional) Enable sticky learning on the interface. To disable sticky learning on an interface, use the **no switchport port-security mac-address sticky** command. The interface converts the sticky secure MAC addresses to dynamic secure addresses.
Step 10	Switch(config-if)# [**no**] **switchport port-security mac-address** *mac_address* **sticky** [**vlan** [**voice** | **access**]]	Specifies the sticky mac-address for the interface. When you specify the **vlan** keyword, the mac-address becomes sticky in the specified VLAN. To delete a sticky secure MAC addresses from the address table, use the **no switchport port-security mac-address** *mac_address* **sticky** command. To convert sticky to dynamic addresses, use the **no switchport port-security mac-address sticky** command. **Note** This command only applies to access, PVLAN host, and PVLAN promiscuous mode. For more details on PVLAN or trunk or regular trunk mode, refer to the "Port Security on Trunk Ports" section on page 35-16.
Step 11	Switch(config-if)# **end**	Returns to privileged EXEC mode.
Step 12	Switch# **show port-security address interface** *interface_id* Switch# **show port-security address**	Verifies your entries.

Note To clear dynamically learned port security MAC addresses in the CAM table, use the **clear port-security dynamic** command. The **address** keyword enables you to clear a secure MAC addresses. The **interface** keyword enables you to clear all secure addresses on any interface (including any port channel interface). The **VLAN** keyword allows you to clear port security MACs on a per-VLAN per-port basis.

Examples

The following examples are provided:

- Example 1: Setting Maximum Number of Secure Addresses, page 35-10
- Example 2: Setting a Violation Mode, page 35-10
- Example 3: Setting the Aging Timer, page 35-11
- Example 4: Setting the Aging Timer Type, page 35-11
- Example 5: Configuring a Secure MAC Address, page 35-11
- Example 6: Configuring Sticky Port Security, page 35-12
- Example 7: Setting a Rate Limit for Bad Packets, page 35-13
- Example 8: Clearing Dynamic Secure MAC Addresses, page 35-13

Example 1: Setting Maximum Number of Secure Addresses

This example shows how to enable port security on the Fast Ethernet interface 3/12 and how to set the maximum number of secure addresses to 5. The violation mode is the default, and no secure MAC addresses are configured.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 3/12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 5
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# end
Switch# show port-security interface fastethernet 3/12
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Enabled
Maximum MAC Addresses      : 5
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
```

Example 2: Setting a Violation Mode

This example shows how to set the violation mode on the Fast Ethernet interface 3/12 to restrict.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 3/12
Switch(config-if)# switchport port-security violation restrict
Switch(config-if)# end
Switch#
```

SNMP traps can be enabled with a rate-limit to detect port-security violations due to restrict mode. The following example shows how to enable traps for port-security with a rate of 5 traps per second:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# snmp-server enable traps port-security trap-rate 5
Switch(config)# end
Switch#
```

Example 3: Setting the Aging Timer

This example shows how to set the aging time to 2 hours (120 minutes) for the secure addresses on the Fast Ethernet interface 5/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport port-security aging time 120
Switch(config-if)# end
Switch#
```

This example shows how to set the aging time to 2 minutes:

```
Switch(config-if)# switchport port-security aging time 2
```

You can verify the previous commands with the **show port-security interface** command.

Example 4: Setting the Aging Timer Type

This example shows how to set the aging timer type to Inactivity for the secure addresses on the Fast Ethernet interface 3/5:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 3/5
Switch(config-if)# switch port-security aging type inactivity
Switch(config-if)# end
Switch# show port-security interface fastethernet 3/5
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
```

Example 5: Configuring a Secure MAC Address

This example shows how to configure a secure MAC address on Fast Ethernet interface 5/1 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 10
Switch(config-if)# switchport port-security mac-address 0000.0000.0003 (Static secure MAC)
Switch(config-if)# end
Switch#show port address
Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                     Ports   Remaining Age
                                                              (mins)
----    -----------       ----                     -----   -------------
   1    0000.0000.0003    SecureConfigured         Fa5/1         -

------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 2
Max Addresses limit in System (excluding one mac per port) : 3072
```

Example 6: Configuring Sticky Port Security

This example shows how to configure a sticky MAC address on Fast Ethernet interface 5/1 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa5/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# switchport port-security maximum 5
Switch(config-if)# end
```

Note Sending traffic to the ports causes the system to configure the port with sticky secure addresses.

```
Switch# show port-security address
          Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                     Ports   Remaining Age
                                                              (mins)
----    -----------       ----                     -----   -------------
   1    0000.0000.0001    SecureSticky             Fa5/1         -
   1    0000.0000.0002    SecureSticky             Fa5/1         -
   1    0000.0000.0003    SecureSticky             Fa5/1         -
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 2
Max Addresses limit in System (excluding one mac per port) : 3072
Switch# show running-config interface fastEthernet 5/1
Building configuration...

Current configuration : 344 bytes
!
interface FastEthernet5/1
 switchport mode access
 switchport port-security
 switchport port-security maximum 5
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0000.0000.0001
 switchport port-security mac-address sticky 0000.0000.0002
 switchport port-security mac-address sticky 0000.0000.0003
end

Switch#
```

Example 7: Setting a Rate Limit for Bad Packets

The following example shows how to configure rate limit for invalid source packets on Fast Ethernet interface 5/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport port-security limit rate invalid-source-mac 100
Switch(config-if)# end
Switch#
```

The following example shows how to configure rate limit for invalid source packets on Fast Ethernet interface 5/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport port-security limit rate invalid-source-mac none
Switch(config-if)# end
Switch#
```

Example 8: Clearing Dynamic Secure MAC Addresses

The following example shows how to clear a dynamic secure MAC address:

```
Switch# clear port-security dynamic address 0000.0001.0001
```

The following example shows how to clear all dynamic secure MAC addresses on Fast Ethernet interface 2/1:

```
Switch# clear port-security dynamic interface fa2/1
```

The following example shows how to clear all dynamic secure MAC addresses in the system:

```
Switch# clear port-security dynamic
```

Port Security on a Private VLAN Port

You can configure port security on a private VLAN port to take advantage of private VLAN functionality as well as to limit the number of MAC addresses.

Note This section follows the same configuration model that was presented for access ports.

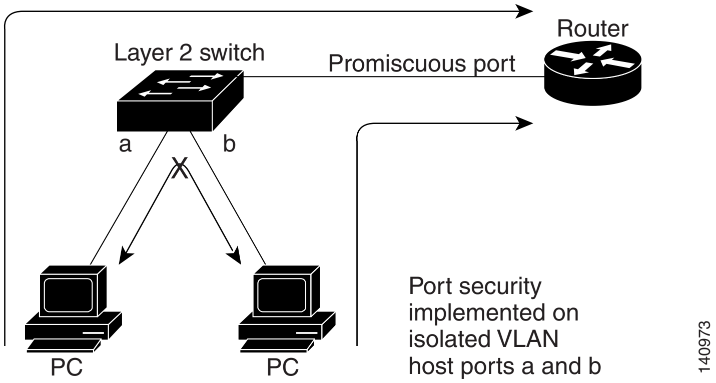
These sections describe how to configure trunk port security on host and promiscuous ports:

- Configuring Port Security on an Isolated Private VLAN Host Port, page 35-14
- Example of Port Security on an Isolated Private VLAN Host Port, page 35-15
- Configuring Port Security on a Private VLAN Promiscous Port, page 35-15
- Example of Port Security on a Private VLAN Promiscous Port, page 35-16

Configuring Port Security on an Isolated Private VLAN Host Port

Figure 35-1 illustrates a typical topology for port security implemented on private VLAN host ports. In this topology, the PC connected through port a on the switch can communicate only with the router connected through the promiscuous port on the switch. The PC connected through port a cannot communicate with the PC connected through port b.

Figure 35-1 Port Security on Isolated Private VLAN Host Ports


Router
Layer 2 switch
Promiscuous port
a
b
X
PC
PC
Port security implemented on isolated VLAN host ports a and b
140973

Note Dynamic addresses secured on an isolated private VLAN host port on private VLANs are secured on the secondary VLANs, and not primary VLANs.

To configure port security on an isolated private VLAN host port, perform this task:

	Command	Purpose
Step 1	`Switch# **configure terminal**`	Enter global configuration mode.
Step 2	`Switch(config)# **vlan** *sec_vlan_id*`	Specifies a secondary VLAN.
Step 3	`Switch(config-vlan)# **private-vlan isolated**`	Sets the private VLAN mode to isolated.
Step 4	`Switch(config-vlan)# **exit**`	Returns to global configuration mode.
Step 5	`Switch(config)# **vlan** *pri_vlan_id*`	Specifies a primary VLAN.
Step 6	`Switch(config-vlan)# **private-vlan primary**`	Specifies the VLAN as the primary private VLAN.
Step 7	`Switch(config-vlan)# **private-vlan association add** *sec_vlan_id*`	Creates an association between a secondary VLAN and a primary VLAN.
Step 8	`Switch(config-vlan)# **exit**`	Returns to global configuration mode.
Step 9	`Switch(config)# **interface** *interface_id*`	Enters interface configuration mode and specifies the physical interface to configure.
Step 10	`Switch(config-if)# **switchport mode private-vlan host**`	Specifies that the ports with a valid private VLAN trunk association become active host private VLAN trunk ports.
Step 11	`Switch(config-if)# **switchport private-vlan host-association** *primary_vlan secondary_vlan*`	Establishes a host association on an isolated host port.
Step 12	`Switch(config-if)# [**no**] **switchport port-security**`	Enables port security on the interface.

	Command	Purpose (continued)
Step 13	`Switch(config-if)# end`	Returns to privileged EXEC mode.

Step 14	`Switch# show port-security address interface interface_id`
	`Switch# show port-security address`
		Verifies your entries.

Example of Port Security on an Isolated Private VLAN Host Port

The following example shows how to configure port security on an isolated private VLAN host port, Fast Ethernet interface 3/12:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vlan 6
Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# exit
Switch(config)# vlan 3
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# private-vlan association add 6
Switch(config-vlan)# exit
Switch(config)# interface fastethernet 3/12
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan association host 3 6
Switch(config-if)# switchport port-security
Switch(config-if)# end
```

Configuring Port Security on a Private VLAN Promiscous Port

To configure port security on a private VLAN promiscuous port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enter global configuration mode.
Step 2	`Switch(config)# vlan sec_vlan_id`	Specifies the VLAN.
Step 3	`Switch(config-vlan)# private-vlan isolated`	Sets the private VLAN mode to isolated.
Step 4	`Switch(config-vlan)# exit`	Returns to global configuration mode.
Step 5	`Switch(config)# vlan pri_vlan_id`	Specifies the VLAN.
Step 6	`Switch(config-vlan)# private-vlan primary`	Designates the VLAN as the primary private VLAN.
Step 7	`Switch(config-vlan)# private-vlan association add sec_vlan_id`	Creates an association between a secondary VLAN and a primary VLAN.
Step 8	`Switch(config-vlan)# exit`	Returns to global configuration mode.
Step 9	`Switch(config)# interface interface_id`	Enters interface configuration mode and specifies the physical interface to configure.
Step 10	`Switch(config-if)# switchport mode private-vlan promiscuous`	Specifies that the ports with a valid PVLAN mapping become active promiscuous ports.
Step 11	`Switch(config-if)# switchport private-vlan mapping primary_vlan secondary_vlan`	Configures a private VLAN for the promiscuous ports
Step 12	`Switch(config-if)# switchport port-security`	Enables port security on the interface.

	Command	Purpose (continued)
Step 13	`Switch(config-if)# end`	Returns to privileged EXEC mode.

Step 14	`Switch# show port-security address interface interface_id`
	`Switch# show port-security address`
		Verifies your entries.

Example of Port Security on a Private VLAN Promiscous Port

The following example shows how to configure port security on a private VLAN promiscuous port, Fast Ethernet interface 3/12:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vlan 6
Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# exit
Switch(config)# vlan 3
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# private-vlan association add 6
Switch(config-vlan)# exit
Switch(config)# interface fastethernet 3/12
Switch(config-if)# switchport mode private-vlan promiscuous
Switch(config-if)# switchport mode private-vlan mapping 3 6
Switch(config-if)# switchport port-security
Switch(config-if)# end
```

Port Security on Trunk Ports

You might want to configure port security on trunk ports in metro aggregation to limit the number of MAC addresses per VLAN. Trunk port security extends port security to trunk ports. It restricts the allowed MAC addresses or the maximum number of MAC addresses to individual VLANs on a trunk port. Trunk port security enables service providers to block the access from a station with a different MAC address than the ones specified for that VLAN on that trunk port. Trunk port security is also supported on private VLAN trunk ports.

Note Port security can be enabled on a Layer 2 port channel interface configured in mode. The port security configuration on an EtherChannel is kept independent of the configuration of any physical member ports.

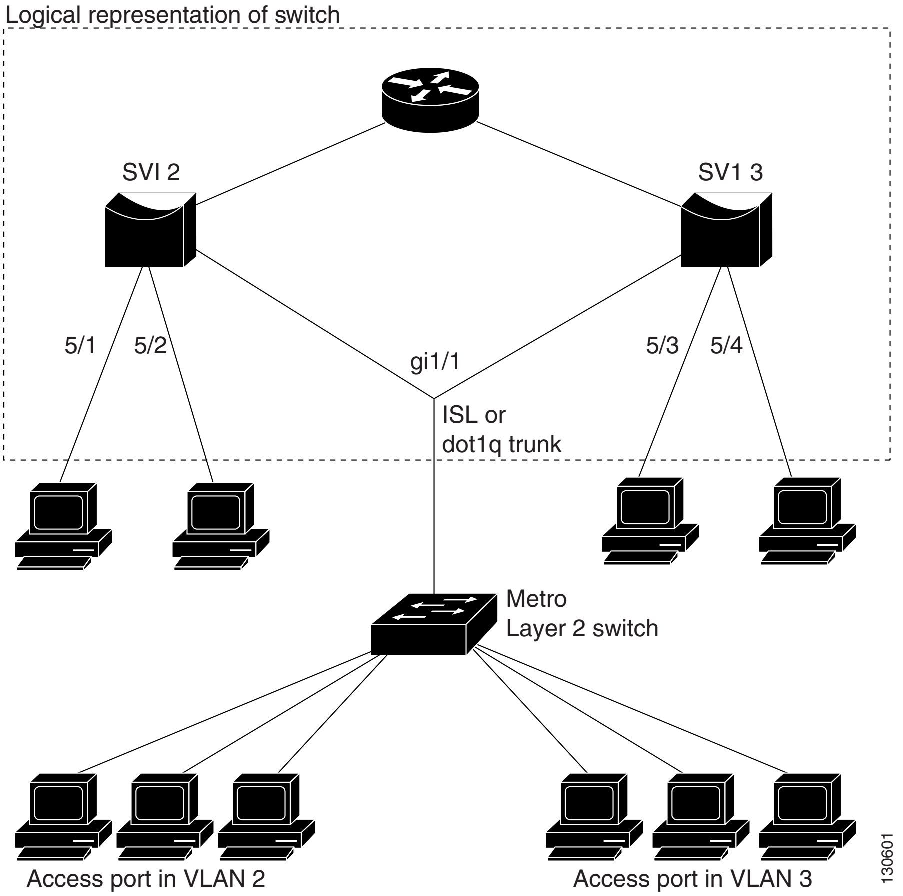
These sections describe how to configure trunk port security:

- Configuring Trunk Port Security, page 35-16
- Examples of Trunk Port Security, page 35-18
- Trunk Port Security Guidelines and Restrictions, page 35-20

Configuring Trunk Port Security

Trunk port security is used when a Catalyst 4500 series switch has a dot1q or isl trunk attached to a neighborhood Layer 2 switch. This may be used, for example, in metro aggregation networks (Figure 35-2).

Figure 35-2 Trunk Port Security



You can configure various port security related parameters on a per-port per-VLAN basis.

Note The steps involved in configuring port security parameters is similar to those for access ports. In addition to those steps, the following per-port per-VLAN configuration steps are supported for trunk ports.

To configure port security related parameters on a per-VLAN per-port basis, perform this task:

	Command	Purpose

Step 1	`Switch(config)# interface interface_id`
	`interface port-channel port_channel_number`
		Enters interface configuration mode and specifies the interface to configure.
		Note The interface can be a Layer 2 port channel logical interface.

Step 2	`Switch(config-if)# switchport trunk encapsulation dot1q`	Sets the trunk encapsulation format to 802.1Q.
Step 3	`Switch(config-if)# switchport mode trunk`	Sets the interface mode. **Note** An interface in the default mode (dynamic desirable) cannot be configured as a secure port.

	Command	Purpose (continued)
Step 4	Switch(config-if)# **switchport port-security maximum** *value* **vlan**	Configures a maximum number of secure mac-addresses for each VLAN on the interface that are not explicitly configured with a maximum mac-address limit. (See the "Maximum Number of Secure MAC Addresses" section on page 35-4.)
Step 5	Switch(config-if)# **vlan-range** *range*	Enters VLAN range sub-mode. **Note** You can specify single or multiple VLANs.
Step 6	Switch(config-if-vlan-range)# **port-security maximum** *value*	Configures a maximum number of secure MAC addresses for each VLAN.
Step 7	Switch(config-if-vlan-range)# **no port-security maximum**	Removes a maximum number of secure MAC addresses configuration for all the VLANs. Subsequently, the maximum value configured on the port will be used for all the VLANs.
Step 8	Switch(config-if-vlan-range)# [**no**] **port-security mac-address** *mac_address*	Configures a secure MAC-address on a range of VLANs.
Step 9	Switch(config-if-vlan-range)# [**no**] **port-security mac-address sticky** *mac_address*	Configures a sticky MAC-address on a range of VLANs.
Step 10	Switch(config-if-vlan-range)# **end**	Returns to interface configuration mode.
Step 11	Switch(config-if)# **end**	Returns to privileged EXEC mode.

Examples of Trunk Port Security

The following examples are provided:

- Example 1: Configuring a Maximum Limit of Secure MAC Addresses for all VLANs, page 35-18
- Example 2: Configuring a Maximum Limit of Secure MAC Addresses for Specific VLANs, page 35-19
- Example 3: Configuring Secure MAC Addresses in a VLAN Range, page 35-19

Example 1: Configuring a Maximum Limit of Secure MAC Addresses for all VLANs

This example shows how to configure a secure MAC-address and a maximum limit of secure MAC addresses on Gigabit Ethernet interface 1/1 for all VLANs:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface g1/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# sw mode trunk
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 3

Switch# show port-security in gi1/1 vlan
Default maximum: 3
VLAN  Maximum    Current
    1         3          0
    2         3          0
    3         3          0
    4          3          0
    5          3          0
    6          3          0
Switch#

Switch# show running interface gi1/1
Building configuration...

Current configuration : 161 bytes
!
interface GigabitEthernet1/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport port-security
 switchport port-security maximum 3 vlan
end
```

Example 2: Configuring a Maximum Limit of Secure MAC Addresses for Specific VLANs

This example shows how to configure a secure MAC-address on interface g1/1 in a specific VLAN or range of VLANs:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface g1/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# sw mode trunk
Switch(config-if)# switchport port-security
Switch(config-if)# vlan-range 2-6
Switch(config-if-vlan-range)# port-security maximum 3
Switch(config-if)# exit

Switch# show port-security interface g1/1 vlan
Default maximum: not set, using 3072
VLAN  Maximum    Current
   2          3          0
   3          3          0
   4          3          0
   5          3          0
   6          3          0
Switch#
```

Example 3: Configuring Secure MAC Addresses in a VLAN Range

This example shows how to configure a secure MAC-address in a VLAN on interface g1/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface g1/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# sw mode trunk
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# vlan-range 2-6
Switch(config-if-vlan-range)# port-security mac-address 1.1.1
Switch(config-if-vlan-range)# port-security mac-address sticky 1.1.2
Switch(config-if-vlan-range)# port-security mac-address sticky 1.1.3
Switch(config-if-vlan-range)# exit
```

```
Switch# show port-security interface g1/1 address vlan 2-4
         Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                     Ports   Remaining Age
                                                             (mins)
----   -----------        ----                     -----   -------------
  2    0001.0001.0001    SecureConfigured         Gi1/1        -
  2    0001.0001.0002    SecureSticky             Gi1/1        -
  2    0001.0001.0003    SecureSticky             Gi1/1        -
  3    0001.0001.0001    SecureConfigured         Gi1/1        -
  3    0001.0001.0002    SecureSticky             Gi1/1        -
  3    0001.0001.0003    SecureSticky             Gi1/1        -
  4    0001.0001.0001    SecureConfigured         Gi1/1        -
  4    0001.0001.0002    SecureSticky             Gi1/1        -
  4    0001.0001.0003    SecureSticky             Gi1/1        -
------------------------------------------------------------------------
Total Addresses: 9

Switch#
```

Trunk Port Security Guidelines and Restrictions

Follow these guidelines when configuring port security related parameters on a per-port per-VLAN basis:

- A secure MAC-address cannot be configured on a VLAN that is not allowed on a regular trunk port.
- The configuration on the primary VLAN on the private VLAN trunk is not allowed. The CLI is rejected and an error message is displayed.
- If a specific VLAN on a port is not configured with a maximum value (directly or indirectly), the maximum configured for the port is used for that VLAN. In this situation, the maximum number of addresses that can be secured on this VLAN is limited to the maximum value configured on the port.

  Each VLAN can be configured with a maximum count that is greater than the value configured on the port. Also, the sum of the maximum configured values for all the VLANs can exceed the maximum configured for the port. In either of these situations, the number of MAC addresses secured on each VLAN is limited to the lesser of the VLAN configuration maximum and the port configuration maximum. Also, the number of addresses secured on the port across all VLANs cannot exceed a maximum that is configured on the port.
- For private VLAN trunk ports, the VLAN on which the configuration is being performed must be in either the allowed VLAN list of the private VLAN trunk or the secondary VLAN list in the association pairs. (The CLI is rejected if this condition is not met.) The allowed VLAN list on a private VLAN trunk is intended to hold the VLAN-IDs of all the regular VLANs that are allowed on the private VLAN trunk.
- Removal of an association pair from a PVLAN trunk causes all static and sticky addresses associated with the secondary VLAN of the pair to be removed from the running configuration. Dynamic addresses associated with the secondary VLAN are deleted from the system.

  Similarly, when a VLAN is removed from the list of allowed PVLAN trunks, the addresses associated with that VLAN are removed.

Note For a regular or private VLAN trunk port, if the VLAN is removed from the allowed VLAN list, all the addresses associated with that VLAN are removed.

Port Mode Changes

Generally, when a port mode changes, all dynamic addresses associated with that port are removed. All static or sticky addresses and other port security parameters configured on the native VLAN are moved to the native VLAN of the port in the new mode. All the addresses on the non-native VLANs are removed.

The native VLAN refers to the following VLAN on the specified port type:

Port Type	Native VLAN
access	access VLAN
trunk	native VLAN
isolated	secondary VLAN (from host association)
promiscuous	primary VLAN (from mapping)
private VLAN trunk	private VLAN trunk native VLAN
.1Q tunnel	access VLAN

For example, when the mode changes from access to private VLAN trunk, all the static or sticky addresses configured on the access VLAN of the access port are moved to the private VLAN native VLAN of the private VLAN trunk port. All other addresses are removed.

Similarly, when the mode changes from private VLAN trunk to access mode, all the static or sticky addresses configured on the private VLAN native VLAN are moved to the access VLAN of the access port. All other addresses are removed.

When a port is changed from trunk to private VLAN trunk, addresses associated with a VLAN on the trunk are retained if that VLAN is present in the allowed list of private VLAN trunk or the secondary VLAN of an association on the private VLAN trunk. If the VLAN is not present in either of them, the address is removed from the running configuration.

When a port is changed from private VLAN trunk to trunk, a static or sticky address is retained if the VLAN associated with the address is present in the allowed VLAN list of the trunk. If the VLAN is not present in the allowed list, the address is removed from running configuration.

Port Security on Voice Ports

You might want to configure port security in an IP Telephony environment when a port is configured with a data VLAN for a PC and a voice VLAN for a Cisco IP Phone.

These sections describe how to configure port security on voice ports:

- Configuring Port Security on Voice Ports, page 35-22
- Examples of Voice Port Security, page 35-24
- Voice Port Security Guidelines and Restrictions, page 35-26

Configuring Port Security on Voice Ports

To configure port security on a voice port, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface interface_id`	Enters interface configuration mode and specifies the physical interface to configure.
Step 2	`Switch(config-if)# switchport mode access`	Sets the interface mode. **Note** An interface in the default mode (dynamic desirable) cannot be configured as a secure port.
Step 3	`Switch(config-if)# [no] switchport port-security`	Enables port security on the interface. To return the interface to the default condition as nonsecure port, use the **no switchport port-security** command.
Step 4	`Switch(config-if)# [no] switchport port-security violation {restrict | shutdown}`	(Optional) Sets the violation mode, the action to be taken when a security violation is detected, as one of these: • **restrict**—A port security violation restricts data and causes the SecurityViolation counter to increment and send an SNMP trap notification. • **shutdown**—The interface is error-disabled when a security violation occurs. **Note** When a secure port is in the error-disabled state, you can bring it out of this state by entering the **errdisable recovery cause psecure-violation** global configuration command or you can manually reenable it by entering the **shutdown** and **no shut down** interface configuration commands. To return the violation mode to the default condition (shutdown mode), use the **no switchport port-security violation shutdown** command.
Step 5	`Switch(config-if)# switchport port-security limit rate invalid-source-mac packets_per_sec`	Sets the rate limit for bad packets. Default is 10 pps.

	Command	Purpose (continued)
Step 6	Switch(config-if)# [**no**] **switchport port-security mac-address** *mac_address* [**vlan** {**voice** | **access**}]	(Optional) Specifies a secure MAC address for the interface. When you specify the **vlan** keyword, addresses are configured in the specified VLAN. • **voice**—MAC address is configured in the voice VLAN. • **access**—MAC address is configured in the access VLAN. You can use this command to configure secure MAC addresses. If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are dynamically learned. To delete a MAC address from the address table, use the **no switchport port-security mac-address** *mac_address* command. **Note** This command only applies to access, PVLAN host, and PVLAN promiscuous mode. For more details on PVLAN, trunk, or regular trunk mode, refer to the "Port Security on Trunk Ports" section on page 35-16.
Step 7	Switch(config-if)# [**no**] **switchport port-security mac-address sticky**	(Optional) Enable sticky learning on the interface. To disable sticky learning on an interface, use the **no switchport port-security mac-address sticky** command. The interface converts the sticky secure MAC addresses to dynamic secure addresses.
Step 8	Switch(config-if)# [**no**] **switchport port-security mac-address** *mac_address* **sticky** [**vlan** {**voice** | **access**}]	Specifies the sticky mac-address for the interface. When you specify the **vlan** keyword, the mac-address becomes sticky in the specified VLAN. • **voice**—MAC address becomes sticky in the voice VLAN. • **access**—MAC address becomes sticky in the access VLAN. To delete a sticky secure MAC addresses from the address table, use the **no switchport port-security mac-address** *mac_address* **sticky** command. To convert sticky to dynamic addresses, use the **no switchport port-security mac-address sticky** command. **Note** This command only applies to access, PVLAN host, and PVLAN promiscuous mode. For more details on PVLAN or trunk or regular trunk mode, refer to the "Port Security on Trunk Ports" section on page 35-16.

	Command	Purpose (continued)
Step 9	Switch(config-if)# **end**	Returns to privileged EXEC mode.
Step 10	Switch# **show port-security address interface** *interface_id* Switch# **show port-security address**	Verifies your entries.

Note To clear dynamically learned port security MAC addresses in the CAM table, use the **clear port-security dynamic** command. The **address** keyword enables you to clear a secure MAC addresses. The **interface** keyword enables you to clear all secure addresses on an interface (including any port channel interface). The **VLAN** keyword allows you to clear port security MACs on a per-VLAN per-port basis.

Examples of Voice Port Security

The following examples are provided:

- Example 1: Configuring Maximum MAC Addresses for Voice and Data VLANs, page 35-24
- Example 2: Configuring Sticky MAC Addresses for Voice and Data VLANs, page 35-25

Example 1: Configuring Maximum MAC Addresses for Voice and Data VLANs

This example shows how to designate a maximum of one MAC address for a voice VLAN (for a Cisco IP Phone, let's say) and one MAC address for the data VLAN (for a PC, let's say) on Fast Ethernet interface 5/1 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa5/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# switchport port-security maximum 1 vlan voice
Switch(config-if)# switchport port-security maximum 1 vlan access
Switch(config-if)# end
```

Note Sending traffic to the ports causes the system to configure the port with sticky secure addresses.

```
Switch# show port-security address
          Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                     Ports   Remaining Age
                                                              (mins)
----    -----------       ----                     -----   -------------
   1    0000.0000.0001    SecureSticky             Fa5/1        -
   3    0000.0000.0004    SecureSticky             Fa5/1        -
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 1
Max Addresses limit in System (excluding one mac per port) : 3072

Switch# show running-config interface fastEthernet 5/1
Building configuration...
```

```
Current configuration : 344 bytes
!
interface FastEthernet5/1
 switchport mode access
 switchport voice vlan 3
 switchport port-security
 switchport port-security maximum 1 vlan voice
 switchport port-security maximum 1 vlan access
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0000.0000.0001
 switchport port-security mac-address sticky 0000.0000.0004 vlan voice
end

Switch#
```

Example 2: Configuring Sticky MAC Addresses for Voice and Data VLANs

This example shows how to configure sticky MAC addresses for voice and data VLANs on Fast Ethernet interface 5/1 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa5/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address sticky 0000.0000.obob vlan voice
Switch(config-if)# switchport port-security mac-address sticky 0000.0000.0005 vlan access
Switch(config-if)# end
```

Note Sending traffic to the ports causes the system to configure the port with sticky secure addresses.

```
Switch# show port-security address
          Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                     Ports   Remaining Age
                                                              (mins)
----    -----------       ----                     -----   -------------
   1    0000.0000.0001    SecureSticky             Fa5/1         -
   1    0000.0000.0002    SecureSticky             Fa5/1         -
   1    0000.0000.0003    SecureSticky             Fa5/1         -
   3    0000.0000.0004    SecureSticky             Fa5/1         -
   1    0000.0000.0005    SecureSticky             Fa5/1         -
   3    0000.0000.0b0b    SecureSticky             Fa5/1         -
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 5
Max Addresses limit in System (excluding one mac per port) : 3072

Switch# show running-config interface fastEthernet 5/1
Building configuration...

Current configuration : 344 bytes
!
interface FastEthernet5/1
 switchport mode access
 switchport voice vlan 3
 switchport port-security
 switchport port-security maximum 5 vlan voice
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0000.0000.0001
 switchport port-security mac-address sticky 0000.0000.0002
 switchport port-security mac-address sticky 0000.0000.0003
 switchport port-security mac-address sticky 0000.0000.0004 vlan voice
 switchport port-security mac-address sticky 0000.0000.0005
 switchport port-security mac-address sticky 0000.0000.0b0b vlan voice
end

Switch#
```

Voice Port Security Guidelines and Restrictions

Port security as implemented on voice ports behaves the same as port security on access ports:

- You can configure sticky port security on voice ports. If sticky port security is enabled on a voice port, addresses secured on data and voice VLANs are secured as sticky addresses.
- You can configure maximum secure addresses per VLAN. You can set a maximum for either the data VLAN or the voice VLAN. You can also set a maximum per-port, just as with access ports.
- You can configure port security MAC addresses on a per-VLAN basis on either the data or voice VLANs.
- Prior to Cisco IOS Release 12.2(31)SG, you required three MAC addresses as the maximum parameter to support an IP Phone and a PC. With Cisco IOS Release 12.2(31)SG and later releases, the maximum parameter must be configured to two, one for the phone and one for the PC.

Displaying Port Security Settings

Use the **show port-security** command to display port-security settings for an interface or for the switch.

To display traffic control information, perform one or more of these tasks:

Command	Purpose
`Switch#` **`show interface status err-disable`**	Displays interfaces that have been error-disabled along with the cause for which they were disabled.
Switch# **show port-security** [**interface** *interface_id* | **interface** *port_channel port_channel_*number]	Displays port security settings for the switch or for the specified interface, including 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, and the violation mode. The interface can be a port channel logical interface.
Switch# **show port-security** [**interface** *interface_id* | **interface** *port_channel port_channel_*number] **address**	Displays all secure MAC addresses configured on all switch interfaces or on a specified interface with aging information for each address.
Switch# **show port-security** [**interface** *interface_id* | **interface** *port_channel port_channel_*number] **vlan** *vlan_list*	Displays the maximum allowed number of secure MAC addresses and the current number of secure MAC addresses on a specific VLAN-list and a specific interface.
Switch# **show port-security** [**interface** *interface_id* | **interface** *port_channel port_channel_*number] [**address** [**vlan** *vlan_list*]]	Displays all secure MAC addresses configured on a specific VLAN-list and a specific interface.

Examples

The following examples are provided:

- Example 1: Displaying Security Settings for the Entire Switch, page 35-27
- Example 2: Displaying Security Settings for an Interface, page 35-27
- Example 3: Displaying all Secure Addresses for the Entire Switch, page 35-28
- Example 4: Displaying a Maximum Number of MAC Addresses on an Interface, page 35-28
- Example 5: Displaying Security Settings on an Interface for a VLAN Range, page 35-28
- Example 6: Displaying Secured MAC Addresses and Aging Information on an Interface, page 35-29
- Example 7: Displaying Secured MAC Addresses for a VLAN Range on an Interface, page 35-29

Example 1: Displaying Security Settings for the Entire Switch

This example shows how to display port security settings for the entire switch:

```
Switch# show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)          (Count)
---------------------------------------------------------------------------
      Fa3/1              2            2                  0         Restrict
      Fa3/2              2            2                  0         Restrict
      Fa3/3              2            2                  0         Shutdown
      Fa3/4              2            2                  0         Shutdown
      Fa3/5              2            2                  0         Shutdown
      Fa3/6              2            2                  0         Shutdown
      Fa3/7              2            2                  0         Shutdown
      Fa3/8              2            2                  0         Shutdown
     Fa3/10              1            0                  0         Shutdown
     Fa3/11              1            0                  0         Shutdown
     Fa3/12              1            0                  0         Restrict
     Fa3/13              1            0                  0         Shutdown
     Fa3/14              1            0                  0         Shutdown
     Fa3/15              1            0                  0         Shutdown
     Fa3/16              1            0                  0         Shutdown
        Po2              3            0                  0         Shutdown
---------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     :8
Max Addresses limit in System (excluding one mac per port) :3072
Global SNMP trap control for port-security                 :20 (traps per second)
```

Example 2: Displaying Security Settings for an Interface

This example shows how to display port security settings for Fast Ethernet interface 5/1:

```
Switch# show port-security interface fastethernet 5/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : 0000.0001.001a:1
Security Violation Count   : 0
```

Example 3: Displaying all Secure Addresses for the Entire Switch

This example shows how to display all secure MAC addresses configured on all switch interfaces:

```
Switch# show port-security address
          Secure Mac Address Table
-------------------------------------------------------------------
Vlan    Mac Address       Type                Ports   Remaining Age
                                                         (mins)
----    -----------       ----                -----   -------------
   1    0000.0001.0000    SecureConfigured    Fa3/1       15 (I)
   1    0000.0001.0001    SecureConfigured    Fa3/1       14 (I)
   1    0000.0001.0100    SecureConfigured    Fa3/2        -
   1    0000.0001.0101    SecureConfigured    Fa3/2        -
   1    0000.0001.0200    SecureConfigured    Fa3/3        -
   1    0000.0001.0201    SecureConfigured    Fa3/3        -
   1    0000.0001.0300    SecureConfigured    Fa3/4        -
   1    0000.0001.0301    SecureConfigured    Fa3/4        -
   1    0000.0001.1000    SecureDynamic    Fa3/5        -
   1    0000.0001.1001    SecureDynamic    Fa3/5        -
   1    0000.0001.1100    SecureDynamic    Fa3/6        -
   1    0000.0001.1101    SecureDynamic    Fa3/6        -
   1    0000.0001.1200    SecureSticky    Fa3/7        -
   1    0000.0001.1201    SecureSticky    Fa3/7        -
   1    0000.0001.1300    SecureSticky    Fa3/8        -
   1    0000.0001.1301    SecureSticky    Fa3/8        -
   1    0000.0001.2000    SecureSticky    Po2        -
-------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     :8
Max Addresses limit in System (excluding one mac per port) :3072
```

Example 4: Displaying a Maximum Number of MAC Addresses on an Interface

This example shows how to display the maximum allowed number of secure MAC addresses and the current number of secure MAC addressees on Gigabit Ethernet interface 1/1:

```
Switch# show port-security interface g1/1 vlan
Default maximum: 22
VLAN  Maximum    Current
2         22          3
3         22          3
4         22          3
5         22          1
6         22          2
```

Example 5: Displaying Security Settings on an Interface for a VLAN Range

This example shows how to display the port security settings on Gigabit Ethernet interface 1/1 for VLANs 2 and 3:

```
Switch# show port-security interface g1/1 vlan 2-3
Default maximum: 22
VLAN  Maximum    Current
    2         22          3
    3         22          3
```

Example 6: Displaying Secured MAC Addresses and Aging Information on an Interface

This example shows how to display all secure MAC addresses configured on Gigabit Ethernet interface 1/1 with aging information for each address.

```
Switch# show port-security interface g1/1 address

          Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                     Ports   Remaining Age(mins)
----    -----------                 ----                        -----   -------------
   2    0001.0001.0001    SecureConfigured         Gi1/1        -
   2    0001.0001.0002    SecureSticky                 Gi1/1        -
   2    0001.0001.0003    SecureSticky                 Gi1/1        -
   3    0001.0001.0001    SecureConfigured         Gi1/1        -
   3    0001.0001.0002    SecureSticky                 Gi1/1        -
   3    0001.0001.0003    SecureSticky                 Gi1/1        -
   4    0001.0001.0001    SecureConfigured         Gi1/1        -
   4    0001.0001.0002    SecureSticky             Gi1/1        -
   4    0001.0001.0003    SecureSticky             Gi1/1        -
   5    0001.0001.0001    SecureConfigured         Gi1/1        -
   6    0001.0001.0001    SecureConfigured         Gi1/1        -
   6    0001.0001.0002    SecureConfigured         Gi1/1        -
------------------------------------------------------------------------
Total Addresses: 12
```

Example 7: Displaying Secured MAC Addresses for a VLAN Range on an Interface

This example shows how to display all secure MAC addresses configured on VLANs 2 and 3 on Gigabit Ethernet interface 1/1 with aging information for each address:

```
Switch# show port-security interface g1/1 address vlan 2-3

          Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                     Ports   Remaining Age(mins)
----    -----------       ----                     -----   ------------
   2    0001.0001.0001    SecureConfigured         Gi1/1        -
   2    0001.0001.0002    SecureSticky             Gi1/1        -
   2    0001.0001.0003    SecureSticky             Gi1/1        -
   3    0001.0001.0001    SecureConfigured         Gi1/1        -
   3    0001.0001.0002    SecureSticky             Gi1/1        -
   3    0001.0001.0003    SecureSticky             Gi1/1        -
------------------------------------------------------------------------
Total Addresses: 12
Switch#
```

Configuring Port Security with Other Features/Environments

The following topics are discussed:

- DHCP and IP Source Guard, page 35-30
- 802.1X Authentication, page 35-30
- Configuring Port Security in a Wireless Environment, page 35-31
- Configuring Port Security over Layer 2 EtherChannel, page 35-31

DHCP and IP Source Guard

You might want to configure port security with DHCP and IP Source Guard to prevent IP spoofing by unsecured MAC addresses. IP Source Guard supports two levels of IP traffic filtering:

- Source IP address filtering
- Source IP and MAC address filtering

When used in source IP and MAC address filtering, IP Source Guard uses private ACLs to filter traffic based on the source IP address, and uses port security to filter traffic based on the source MAC address. So, port security must be enabled on the access port in this mode.

When both features are enabled, the following limitations apply:

- The DHCP packet is not subject to port security dynamic learning.
- If multiple IP clients are connected to a single access port, port security cannot enforce exact binding of source IP and MAC address for each client.

  Let's say that clients reside on an access port with the following IP/MAC address:

  - client1: MAC1 <---> IP1
  - client2: MAC2 <---> IP2

  Then, any combination of the source MAC and IP address traffic is allowed:

  - MAC1 <---> IP1, valid
  - MAC2 <---> IP2, valid
  - MAC1 <---> IP2, invalid
  - MAC2 <---> IP1, invalid

  IP traffic with the correct source IP and MAC address binding will be permitted and port security will dynamically learn its MAC address. IP traffic with source addresses that are not in the binding will be treated as invalid packets and dropped by port security. To prevent a denial of service attack, you must configure port security rate limiting for the invalid source MAC address.

802.1X Authentication

You might want to configure port security with 802.1X authentication to prevent MAC spoofing. 802.1X is not supported on regular or private VLAN trunks. On access ports and PVLAN host or promiscuous ports, both port security and 802.1X can be configured simultaneously. When both are configured, hosts must be 802.1X authenticated before port security can secure the MAC address of the host. Both 802.1X and port security must approve of the host or a security violation will be triggered. The type of security violation will depend on which feature rejects the port: if the host is allowed by 802.1X (for example, because the port is in multi-host mode) but is disallowed by port security, the port-security violation action will be triggered. If the host is allowed by port security but rejected by 802.1X (for example, because the host is non-authorized on a single-host mode port) then the 802.1X security violation action will be triggered.

Note 802.1X, port-security and VVID can all be configured on the same port.

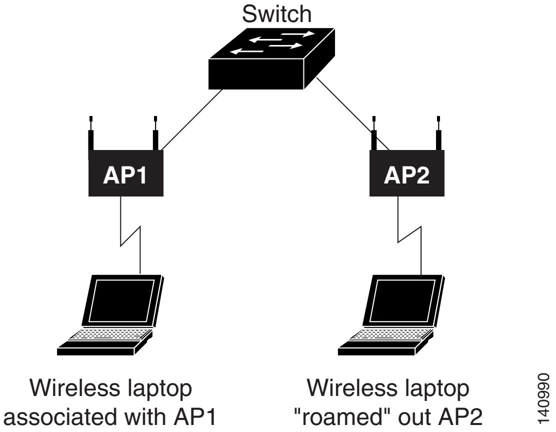
For more information on the interaction between 802.1X and Port Security, see "Using 802.1X with Port Security" on page 15.

Configuring Port Security in a Wireless Environment

If access points are connected to a secure port, do not configure a static MAC address for your users. A MAC address might move from one access point to another and might cause security violations if both the access points are connected on the same switch.

Figure 35-3 illustrates a typical topology of port security in a wireless environment.

Figure 35-3 Port Security in a Wireless Environment


Switch
AP1
AP2
Wireless laptop associated with AP1
Wireless laptop "roamed" out AP2
140990

Configuring Port Security over Layer 2 EtherChannel

Note Supervisor Engine 6-E does not support this feature.

Port security can be enabled on an EtherChannel in either trunk or access mode. (Refer to the "Port Security on Access Ports" section on page 35-6 and the "Port Security on Trunk Ports" section on page 35-16 for configuration steps.) When you do this in trunking mode, the MAC address restrictions apply to the entire port-channel on a per VLAN basis.

In general, be aware of the following:

- Port security on Layer 2 EtherChannel works only on access mode or trunk mode and is independent of the configuration on any physical member ports.
- If at least one member port is secured, port security cannot be disabled on the channel interface; it is rejected by the CLI.
- A secure port cannot join a non-secure EtherChannel; it is rejected by the CLI.
- Port security over EtherChannel is supported in both PAgP and LACP modes. It does not apply to Layer 3 EtherChannels.

Port Security Guidelines and Restrictions

Follow these guidelines when configuring port security:

- A secure port cannot be a destination port for the Switch Port Analyzer (SPAN).

- A secure port and a static MAC address configuration for an interface are mutually exclusive.
- When you enter a maximum secure address value for an interface, and the new value is greater than the previous value, the new value overwrites the previously configured value. If the new value is less than the previous value and the number of configured secure addresses on the interface exceeds the new value, the command is rejected.
- While configuring trunk port security on a trunk port, you do not need to account for the protocol packets (like CDP and BPDU) because they are not learned and secured.
- You cannot enable port security aging on sticky secure MAC addresses.
- To restrict MAC spoofing using port security, you must enable 802.1X authentication.
- You cannot configure port security on dynamic ports. You must change the mode to access before you enable port security.
- When port security is enabled on an EtherChannel, 802.1X cannot be enabled.
- A secure EtherChannel does not work in PVLAN mode.

CHAPTER 36

Configuring Control Plane Policing

Note Control Plane Policing is *not* supported on Supervisor Engine 6-E.

This chapter contains information on how to protect your Catalyst 4000 family switch using control plane policing (CoPP). The information covered in this chapter is unique to the Catalyst 4500 series switches, and it supplements the network security information and procedures in Chapter 39, "Configuring Network Security with ACLs." This information also supplements the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, *Cisco IOS Release 12.4*, at this URL:

  http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a008043360a.html

- *Cisco IOS Security Command Reference*, *Cisco IOS Release 12.4*, at this URL:

  http://www.cisco.com/en/US/products/ps6350/products_command_reference_book09186a008042df75.html

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

This chapter includes the following major sections:

Understanding How Control Plane Policing Works

The control plane policing (CoPP) feature increases security on the Catalyst 4000 family switch by protecting the CPU from unnecessary or DoS traffic and giving priority to important control plane and management traffic. The Classification TCAM and QoS policers provide hardware support for CoPP. CoPP works with all supervisor engines supported by Cisco IOS Release 12.2(31)SG.

The traffic managed by the CPU is divided into three functional components or *planes*:

- Data plane
- Management plane
- Control plane

You can use CoPP to protect most of the CPU bound traffic and ensure routing stability, reachability and packet delivery. Most importantly, CoPP is often used to protect the CPU from the DoS attack. There is a list of pre-defined ACLs matching a selected set of Layer 2 and Layer 3 control plane packets. You can define your preferred policing parameters to each of these control packets but you cannot modify the matching criteria of these pre-defined ACLs. Following is the list of pre-defined ACLs:

Pre-defined Named ACL	**Description**
system-cpp-dot1x	MacDA = 0180.C200.0003
system-cpp-bpdu-range	MacDA = 0180.C200.0000 - 0180.C200.000F
system-cpp-cdp	MacDA = 0100.0CCC.CCCC (UDLD/DTP/VTP/Pagp)
system-cpp-sstp	MacDA = 0100.0CCC.CCCD
system-cpp-cgmp	Mac DA = 01-00-0C-DD-DD-DD
system-cpp-ospf	IP Protocol = OSPF, IPDA matches 224.0.0.0/24
system-cpp-igmp	IP Protocol = IGMP, IPDA matches 224.0.0.0/3
system-cpp-pim	IP Protocol = PIM, IPDA matches 224.0.0.0/24
system-cpp-all-systems-on-subnet	IPDA = 224.0.0.1
system-cpp-all-routers-on-subnet	IPDA = 224.0.0.2
system-cpp-ripv2	IPDA = 224.0.0.9
system-cpp-ip-mcast-linklocal	IP DA = 224.0.0.0/24
system-cpp-dhcp-cs	IP Protocol = UDP, L4SrcPort = 68, L4DstPort = 67
system-cpp-dhcp-sc	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 68
system-cpp-dhcp-ss	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 67

For the Data Plane and Management Plane traffic, you can define your own ACLs to match the traffic class that you want to police.

CoPP uses MQC to define traffic classification criteria and to specify the configurable policy actions for the classified traffic. MQC uses class maps to define 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 policy to be directly attached to the control plane.

The only policy-map that you can attach to the control-plane is *system-cpp-policy*. It must contain the pre-defined class-maps in the pre-defined order at the beginning of the policy map. The best way to create the system-cpp-policy policy-map is through the global macro *system-cpp*.

The system-cpp-policy contains the pre-defined class maps for the control plane traffic. The names of all system defined CoPP class maps and their matching ACLs contain the prefix “system-cpp-”. By default, no action is specified for each traffic class. You can define your own class maps matching CPU bound data plane and management plane traffic. You can add your defined class maps to the system-cpp-policy policy-map.

Caveats for Control Plane Policing

- Port Security might cancel its effect for non-IP control packets.

  Although Source MAC Learning on the Catalyst 4500 series switch is performed in software, learning of source MAC addresses from control packets (e.g.: IEEE BPDU/CDP/SSTP BPDU/GARP/etc) is dis-allowed. Once you configure Port Security on a port where you expect to receive a high rate of such (possibly rogue) control packets, the system generates a copy of the packet to the CPU (until the source address is learned, how Port Security is implemented), rather than forward it.

  The current architecture of the Catalyst 4500 switching engine does not allow you to apply policing on the copy of packets sent to the CPU; policing can only be applied on packets that are forwarded to CPU. So, copies of packets are sent to the CPU at the rate control packets arrive and Port Security is not triggered because learning from control packets is dis-allowed. Furthermore, policing will not be applied because the packet copy, not the original, is sent to the CPU.

- As of Cisco IOS Release 12.2(31)SGA1, the GARP class is no longer part of the CoPP. (Due to the fix associated with CSCsg08775, even though the system-cpp-garp-range entry still appears in the CPP configuration, it is merely idling and will be removed in future releases.) Henceforward, you can manipulate GARP traffic with user ACLs and QoS. If you want to protect CPU against GARP packets, you also can "police down" GARP packets using CoPP after you define the user class for the GARP packet. (This is now possible because GARP is no longer part of the Static CAM area.)

  Due to tight integration of CPP implementation between IOS and platform code, an error message will always appear during boot-up and CPP will not be applied when downgrading IOS software from a version where this caveat is integrated to a previous release (where this fix is not present):

  ```
  %Invalid control plane policy-map; Please unconfigure policy-map attached to
  control-plane, and associated class-maps, and execute config command "macro global
  apply system-cpp" error: failed to install policy map system-cpp-policy
  ```

  As a workaround do the following:

  1. Back-up your configuration when performing software downgrading.

  2. Remove all CPP entries manually from the config and then re-appy the **macro global apply system-cpp** command.

  There should be no problem associated with this caveat while upgrading between releases.

CoPP Default Configuration

CoPP is disabled by default.

Configuring CoPP

This section includes the following tasks:

- Configure CoPP for Control Plan Traffic, page 36-4
- Configure CoPP for Data Plane and Management Plan Traffic, page 36-5

Configure CoPP for Control Plan Traffic

To configure CoPP for Control Plane traffic, perform this task:

	Command	Purpose
Step 1	`Switch#` **`config terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`qos`**	(Optional) Enables QoS globally.

Step 3	`Switch(config)#`
	`macro global apply system-cpp`
		(Optional) Creates the system-cpp-policy policy-map and attaches it to the control-plane.

Step 4	Switch(config)# **policy-map system-cpp-policy**
	Switch(config-pmap)# **class** {**system-cpp-dot1x** | **system-cpp-bpdu-range** | **system-cpp-cdp** | **service** | **system-cpp-sstp** | **system-cpp-cgmp** | **system-cpp-ospf** | **system-cpp-igmp** | **system-cpp-pim** | **system-cpp-all-systems-on-subnet** | **system-cpp-all-routers-on-subnet** | **system-cpp-ripv2** | **system-cpp-ip-mcast-linklocal** | **system-cpp-dhcp-cs** | **system-cpp-dhcp-sc** | **system-cpp-dhcp-ss**}
	Switch(config-pmap-c)# **police** [**aggregate** *name*] *rate burst* [**conform-action** {**drop** | **transmit**}] [{**exceed-action** {**drop** | **transmit**}}]}
		Associates actions to one or multiple system defined control plane traffic in the service policy map. Repeat this step if necessary.

Step 5	`Switch#` **`show policy-map system-cpp-policy`**	(Optional) Verifies the configuration

The following example shows how to police CDP packets:

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class system-cpp-cdp
Switch(config-pmap-c)# police 32000 1000 conform-action transmit exceed-action drop
Switch(config-pmap-c)# end
Switch# show policy-map system-cpp-policy
  Policy Map system-cpp-policy
    Class system-cpp-dot1x
    Class system-cpp-bpdu-range
*    Class system-cpp-cdp
      police 32000 bps 1000 byte conform-action transmit exceed-action drop *
    Class system-cpp-sstp
    Class system-cpp-cgmp
    Class system-cpp-ospf
    Class system-cpp-igmp
    Class system-cpp-pim
    Class system-cpp-all-systems-on-subnet
    Class system-cpp-all-routers-on-subnet
    Class system-cpp-ripv2
    Class system-cpp-ip-mcast-linklocal
    Class system-cpp-dhcp-cs
    Class system-cpp-dhcp-sc
    Class system-cpp-dhcp-ss
Switch#
```

Configure CoPP for Data Plane and Management Plan Traffic

To configure CoPP for Data Plane and Management Plane traffic, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`qos`**	(Optional) Enables QoS globally.

Step 2	`Switch(config)#`
	`macro global apply system-cpp`
		(Optional) Attaches the system-cpp-policy policy-map to the control-plane.

	Command	Purpose

Step 3	`Switch(config)# {`**`ip`**` | `**`mac`**`} `**`access-list extended`**` {`*`access-list-name`*`}`
	
	For an **ip** access list, issue `Switch(config-ext-nacl)#{`**`permit`**`|`**`deny`**`} {`*`protocol`*`} `**`source`**` {`*`source-wildcard`*`} `**`destination`**` {`*`destination-wildcard`*`}`
	
	For a **mac** access list, issue `Switch(config-ext-macl)#{`**`permit`**`|`**`deny`**`} `**`source`**` {`*`source-wildcard`*`} `**`destination`**` {`*`destination-wildcard`*`} [`**`protocol-family`**`]`
	
	OR
	
	`Switch(config)# `**`access-list`**` {`*`access-list-name`*`} {`**`permit`**` | `**`deny`**`} {`**`type-code wild-mask`**` | `**`address mask`**`}`
		Defines ACLs to match traffic:
		• **permit** - sets the conditions under which a packet passes a named ACL
		• **deny** - sets the conditions under which a packet does not pass a name ACL
		Note You must configure ACLs in most cases to identify the important or unimportant traffic.
		• **type-code** - 16-bit hexadecimal number written with a leading 0x; for example, 0x6000. Specify either a Link Service Access Point (LSAP) type code for 802-encapsulated packets or a SNAP type code for SNAP-encapsulated packets. (LSAP, sometimes called SAP, refers to the type codes found in the DSAP and SSAP fields of the 802 header.)
		• **wild-mask** - 16-bit hexadecimal number whose ones bits correspond to bits in the type-code argument. The wild-mask indicates which bits in the type-code argument should be ignored when making a comparison. (A mask for a DSAP/SSAP pair should always be 0x0101 because these two bits are used for purposes other than identifying the SAP code.)
		• **address** - 48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. This field is used for filtering by vendor code.
		• **mask** - 48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. The ones bits in mask are the bits to be ignored in address. This field is used for filtering by vendor code.

Step 4	`Switch(config)# `**`class-map`**` {`*`traffic-class-name`*`}`
	
	`Switch(config-cmap)# `**`match access-group`**` {`**`access-list-number`**` | `**`name`**` {`*`access-list-name`*`}}`
		Defines the packet classification criteria. Use the **match** statements to identify the traffic associated with the class.

Step 5	`Switch(config-cmap)# `**`exit`**	Returns to global configuration mode.

	Command	Purpose

Step 6	`Switch(config)#` **`policy-map system-cpp-policy`**
	`Switch(config-pmap)#` **`class`** *`<class-map-name>`*
	`Switch(config-pmap-c)#` **`police`** `[`**`aggregate`** *`name`*`]` *`rate burst`* `[`**`conform-action`** `{`**`drop`** `|` **`transmit`**`}] [{`**`exceed-action`** `{`**`drop`** `|` **`transmit`**`}}]`
		Adds the traffic classes to the CoPP policy-map. Uses the **police** statement to associate actions to the traffic class.

Step 7	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 8	`Switch#` **`show policy-map system-cpp-policy`**	Verifies your entries.

The following example shows how to configure trusted hosts with source addresses 10.1.1.1 and 10.1.1.2 to forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets to be policed at the specific rate (this example assumes the global qos is enabled and the system-cpp-policy policy-map has been created):

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp

! Allow 10.1.1.1 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.1 any eq telnet

! Allow 10.1.1.2 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.2 any eq telnet

! Rate limit all other Telnet traffic.
Switch(config)# access-list 140 permit tcp any any eq telnet

! Define class-map "telnet-class."
Switch(config)# class-map telnet-class
Switch(config-cmap)# match access-group 140
Switch(config-cmap)# exit

! Add the class-map "telnet-class" to "system-cpp-policy" and define ! the proper action
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class telnet-class
Switch(config-pmap-c)# police 80000 1000 conform transmit exceed drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

! Verify the above configuration steps
Switch# show policy-map system-cpp-policy
  Policy Map system-cpp-policy
    Class system-cpp-dot1x
    Class system-cpp-bpdu-range
    Class system-cpp-cdp
      police 32000 bps 1000 byte conform-action transmit exceed-action drop
    Class system-cpp-sstp
    Class system-cpp-cgmp
    Class system-cpp-ospf
    Class system-cpp-igmp
    Class system-cpp-pim
    Class system-cpp-all-systems-on-subnet
    Class system-cpp-all-routers-on-subnet
    Class system-cpp-ripv2
    Class system-cpp-ip-mcast-linklocal
    Class system-cpp-dhcp-cs
    Class system-cpp-dhcp-sc
    Class system-cpp-dhcp-ss
*    Class telnet-class
      police 8000 bps 1000 byte conform-action drop exceed-action drop
```

CoPP Configuration Guidelines and Restrictions

When configuring CoPP, follow these guidelines and restrictions:

- Only ingress CoPP is supported. So only **input** keyword is supported in control-plane related CLIs.
- Use the system defined class maps for policing control plane traffic.
- Control plane traffic can be policed only using CoPP. Traffic cannot be policed at the input interface or VLAN even though a policy-map containing the control-plane traffic is accepted when the policy-map is attached to an interface or VLAN.
- System-defined class maps cannot be used in policy-maps for regular QoS.
- Use ACLs and class-maps to identify data plane and management plane traffic that are handled by CPU. User-defined class maps should be added to the **system-cpp-policy** policy-map for CoPP.
- The policy-map named **system-cpp-policy** is dedicated for CoPP. Once attached to the **control-plane**, it cannot be detached.
- The default **system-cpp-policy** map does not define actions for the system-defined class maps, which means **no policing**.
- The only action supported in **system-cpp-policy** policy-map is **police**.
- Do not use the **log** keyword in the CoPP policy ACLs.
- Both MAC and IP ACLs can be used to define data plane and management plane traffic classes. But if a packet also matches a pre-defined ACL for the control plane traffic, the **police** action (or no police action) of the control plane class will be taken as the control plane classes appear above user-defined classes in the service policy. This is the same MQC semantic.
- The exceeding action **policed-dscp-transmit** is not supported for CoPP.
- CoPP is not enabled unless the global QoS is enabled and **police** action is specified.

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:

```
Switch# show policy-map control-plane

Control Plane

Service-policy input: system-cpp-policy

    Class-map: system-cpp-dot1x (match-all)
      0 packets
      Match: access-group name system-cpp-dot1x
    Class-map: system-cpp-bpdu-range (match-all)
      0 packets
      Match: access-group name system-cpp-bpdu-range

*    Class-map: system-cpp-cdp (match-all)
      160 packets
      Match: access-group name system-cpp-cdp
**      police: Per-interface
        Conform: 22960 bytes Exceed: 0 bytes
*
    Class-map: system-cpp-sstp (match-all)
      0 packets
      Match: access-group name system-cpp-sstp

    Class-map: system-cpp-cgmp (match-all)
      0 packets
      Match: access-group name system-cpp-cgmp

    Class-map: system-cpp-ospf (match-all)
      0 packets
      Match: access-group name system-cpp-ospf

    Class-map: system-cpp-igmp (match-all)
      0 packets
      Match: access-group name system-cpp-igmp

    Class-map: system-cpp-pim (match-all)
      0 packets
      Match: access-group name system-cpp-pim

    Class-map: system-cpp-all-systems-on-subnet (match-all)
      0 packets
      Match: access-group name system-cpp-all-systems-on-subnet

    Class-map: system-cpp-all-routers-on-subnet (match-all)
      0 packets
      Match: access-group name system-cpp-all-routers-on-subnet

    Class-map: system-cpp-ripv2 (match-all)
      0 packets
      Match: access-group name system-cpp-ripv2

    Class-map: system-cpp-ip-mcast-linklocal (match-all)
      0 packets
      Match: access-group name system-cpp-ip-mcast-linklocal

    Class-map: system-cpp-dhcp-cs (match-all)
      83 packets
      Match: access-group name system-cpp-dhcp-cs

    Class-map: system-cpp-dhcp-sc (match-all)
      0 packets
      Match: access-group name system-cpp-dhcp-sc

    Class-map: system-cpp-dhcp-ss (match-all)
      0 packets
      Match: access-group name system-cpp-dhcp-ss

*    Class-map: telnet-class (match-all)
      0 packets
      Match: access-group 140
**      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes*
    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
Switch#
```

To clear the counters on the control-plane, enter the **clear control-plane *** command:

```
Switch# clear control-plane *
Switch#
```

To display all the CoPP access list information, enter the **show access-lists** command:

```
Switch# show access-lists
Extended IP access list system-cpp-all-routers-on-subnet
10 permit ip any host 224.0.0.2
Extended IP access list system-cpp-all-systems-on-subnet
10 permit ip any host 224.0.0.1
Extended IP access list system-cpp-dhcp-cs
10 permit udp any eq bootpc any eq bootps Extended IP access list
system-cpp-dhcp-sc
10 permit udp any eq bootps any eq bootpc Extended IP access list
system-cpp-dhcp-ss
10 permit udp any eq bootps any eq bootps Extended IP access list
system-cpp-igmp
10 permit igmp any 224.0.0.0 31.255.255.255 Extended IP access list
system-cpp-ip-mcast-linklocal
10 permit ip any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ospf
10 permit ospf any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-pim
10 permit pim any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ripv2
10 permit ip any host 224.0.0.9
Extended MAC access list system-cpp-bpdu-range
permit any 0180.c200.0000 0000.0000.000f Extended MAC access list
system-cpp-cdp
permit any host 0100.0ccc.cccc
Extended MAC access list system-cpp-cgmp
permit any host 0100.0cdd.dddd
Extended MAC access list system-cpp-dot1x
permit any host 0180.c200.0003
system-cpp-sstp
permit any host 0100.0ccc.cccd
```

To display one CoPP access list, enter the **show access-lists system-cpp-cdp** command:

```
Switch# show access-list system-cpp-cdp
Extended MAC access list system-cpp-cdp
permit any host 0100.0ccc.cccc
Switch#
```

CHAPTER 37

Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts

This chapter describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping, IP Source Guard, and IPSG for Static Hosts on Catalyst 4500 series switches. It provides guidelines, procedures, and configuration examples.

This chapter consists of the following major sections:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of DHCP Snooping

DHCP snooping is a DHCP security feature that provides security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding table. An untrusted message is a message that is received from outside the network or firewall and that can cause traffic attacks within your network.

The DHCP snooping binding table contains the MAC address, IP address, lease time, binding type, VLAN number, and interface information that corresponds to the local untrusted interfaces of a switch; it does not contain information regarding hosts interconnected with a trusted interface. An untrusted interface is an interface that is configured to receive messages from outside the network or firewall. A trusted interface is an interface that is configured to receive only messages from within the network.

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. It also gives you a way to differentiate between untrusted interfaces connected to the end-user and trusted interfaces connected to the DHCP server or another switch.

Note In order to enable DHCP snooping on a VLAN, you must enable DHCP snooping on the switch.

You can configure DHCP snooping for switches and VLANs. When you enable DHCP snooping on a switch, the interface acts as a Layer 2 bridge, intercepting and safeguarding DHCP messages going to a Layer 2 VLAN. When you enable DHCP snooping on a VLAN, the switch acts as a Layer 2 bridge within a VLAN domain.

Topics include:

- Trusted and Untrusted Sources, page 37-2
- Overview of the DHCP Snooping Database Agent, page 37-2
- Option-82 Data Insertion, page 37-3

Trusted and Untrusted Sources

The DHCP snooping feature determines whether traffic sources are trusted or untrusted. An untrusted source may initiate traffic attacks or other hostile actions. To prevent such attacks, the DHCP snooping feature filters messages and rate-limits traffic from untrusted sources.

In an enterprise network, devices under your administrative control are trusted sources. These devices include the switches, routers and servers in your network. Any device beyond the firewall or outside your network is an untrusted source. Host ports are generally treated as untrusted sources.

In a service provider environment, any device that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In the Catalyst 4500 series switch, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to devices (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.

Note For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces, as untrusted DHCP messages will be forwarded only to trusted interfaces.

Overview of the DHCP Snooping Database Agent

To retain the bindings across switch reloads, you must use the DHCP snooping database agent. Without this agent, the bindings established by DHCP snooping are lost upon switch reload. Connectivity is lost as well.

The mechanism for 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 accounts for 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, a switch reads entries from the file and adds the bindings to the DHCP snooping database. When 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 situation 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 a 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.

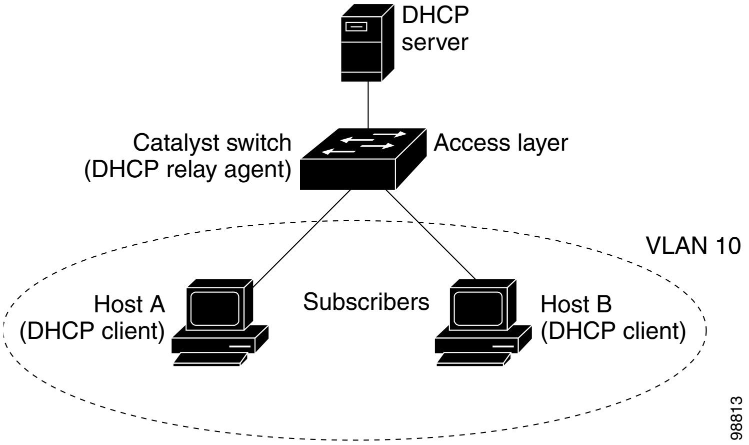
Option-82 Data Insertion

In residential, metropolitan Ethernet-access environments, DHCP can centrally manage the IP address assignments for a large number of subscribers. When the DHCP 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.

Note The DHCP option-82 feature is supported only when DHCP snooping is globally enabled and on the VLANs to which subscriber devices using this feature are assigned.

Figure 37-1 is an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the 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 (the Catalyst switch) is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.

Figure 37-1 DHCP Relay Agent in a Metropolitan Ethernet Network



When you enable the DHCP snooping information option 82 on the 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. By default, theThe remote-ID suboption is the switch MAC address, and the circuit-ID suboption is the port identifier, **vlan-mod-port**, from which the packet is received. Beginning with Cisco IOS Release 12.2(40)SG, you can configure the remote ID and circuit ID. For information on configuring these suboptions, see the "Enabling DHCP Snooping and Option 82" section on page 37-9.
- If the IP address of the relay agent is configured, the switch adds this 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, 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. Then the DHCP server 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. 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.

In the default suboption configuration, whenWhen the described sequence of events occurs, the values in these fields in Figure 37-2 do not change:

- Circuit-ID suboption fields
  - Suboption type
  - Length of the suboption type
  - Circuit-ID type
  - Length of the circuit-ID type
- Remote-ID suboption fields
  - Suboption type
  - Length of the suboption type
  - Remote-ID type
  - Length of the remote-ID type

Figure 37-2 shows the packet formats for the remote-ID suboption and the circuit-ID suboption when the default suboption configuration is used. For the circuit-ID suboption, the module number corresponds to the switch module number. The switch uses the packet formats when you globally enable DHCP snooping and enter the **ip dhcp snooping information option** global configuration command.

Figure 37-2 Suboption Packet Formats

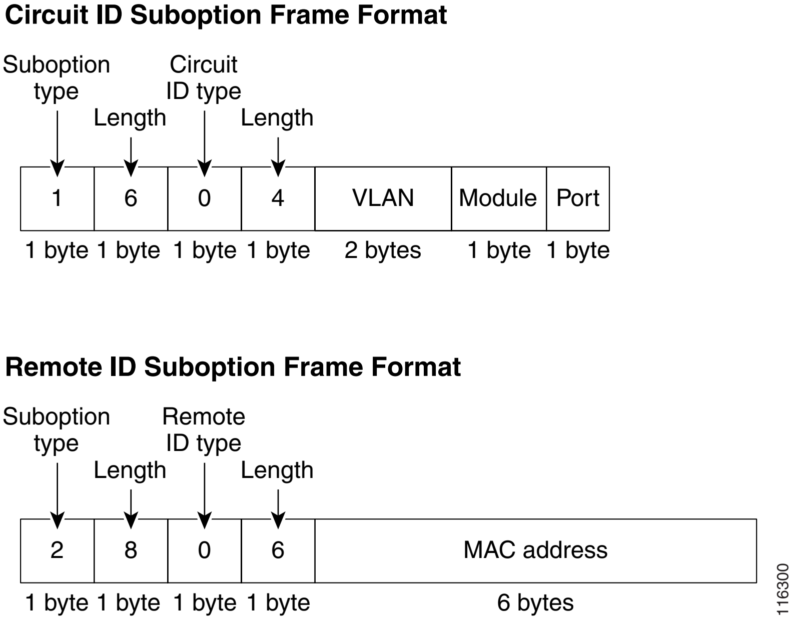
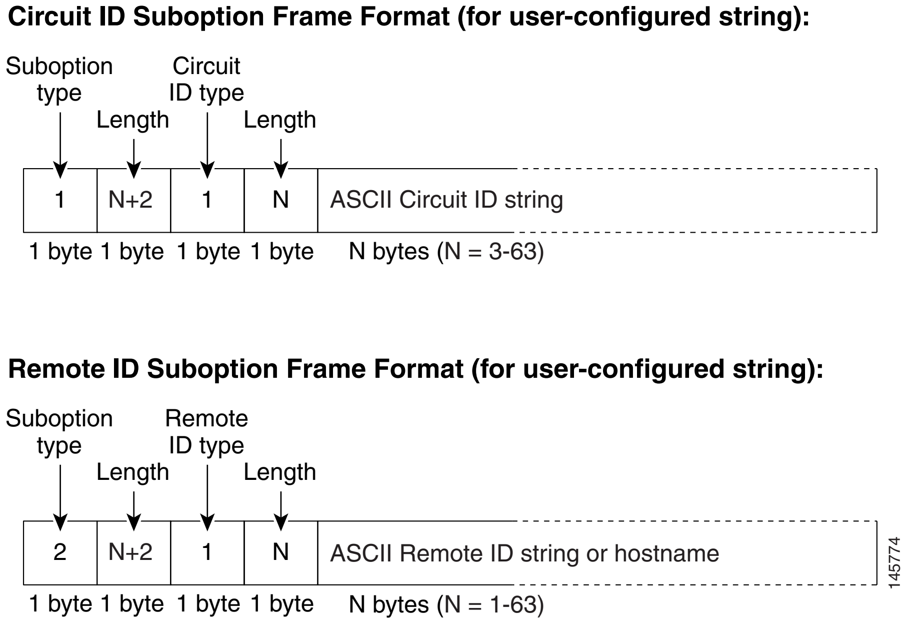


Figure 37-3 shows the packet formats for user-configured remote-ID and circuit-ID suboptions The switch uses these packet formats when DHCP snooping is globally enabled and when the **ip dhcp snooping information option format remote-id** global configuration command and the **ip dhcp snooping vlan information option format-type circuit-id string** interface configuration command are entered.

The values for these fields in the packets change from the default values when you configure the remote-ID and circuit-ID suboptions:

- Circuit-ID suboption fields
  - The circuit-ID type is 1.
  - The length values are variable, depending on the length of the string that you configure.
- Remote-ID suboption fields
  - The remote-ID type is 1.
  - The length values are variable, depending on the length of the string that you configure.

Figure 37-3 User-Configured Suboption Packet Formats


Circuit ID Suboption Frame Format (for user-configured string):
Suboption type
Length
Circuit ID type
Length
1
N+2
1
N
ASCII Circuit ID string
1 byte 1 byte 1 byte 1 byte
N bytes (N = 3-63)
Remote ID Suboption Frame Format (for user-configured string):
Suboption type
Length
Remote ID type
Length
2
N+2
1
N
ASCII Remote ID string or hostname
1 byte 1 byte 1 byte 1 byte
N bytes (N = 1-63)
145774

Configuring DHCP Snooping on the Switch

When you configure DHCP snooping on your switch, you are enabling the switch to differentiate untrusted interfaces from trusted interfaces. You must enable DHCP snooping globally before you can use DHCP snooping on a VLAN. You can enable DHCP snooping independently from other DHCP features.

Once you have enabled DHCP snooping, all the DHCP relay information option configuration commands are disabled; this includes the following commands:

- **ip dhcp relay information check**
- **ip dhcp relay information policy**
- **ip dhcp relay information trusted**
- **ip dhcp relay information trust-all**

These sections describe how to configure DHCP snooping:

- Default Configuration for DHCP Snooping, page 37-7
- Enabling DHCP Snooping, page 37-7
- Enabling DHCP Snooping on the Aggregration Switch, page 37-9
- Enabling DHCP Snooping and Option 82, page 37-9
- Enabling DHCP Snooping on Private VLAN, page 37-11
- Enabling the DHCP Snooping Database Agent, page 37-12
- Configuration Examples for the Database Agent, page 37-12

Note 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/ip_c/ipcprt1/1cddhcp.htm

Default Configuration for DHCP Snooping

DHCP snooping is disabled by default. Table 37-1 shows all the default configuration values for each DHCP snooping option.

Table 37-1 Default Configuration Values for DHCP Snooping

Option	Default Value/State
DHCP snooping	Disabled
DHCP snooping information option	Enabled
DHCP snooping information option allow-untrusted	Disabled
DHCP snooping limit rate	Infinite (functions as if rate limiting were disabled)
DHCP snooping trust	Untrusted
DHCP snooping vlan	Disabled

If you want to change the default configuration values, see the "Enabling DHCP Snooping" section.

Enabling DHCP Snooping

Note When DHCP snooping is enabled globally, DHCP requests are dropped until the ports are configured. Consequently, you should probably configure this feature during a maintenance window and not during production.

To enable DHCP snooping, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`ip dhcp snooping`**	Enables DHCP snooping globally. You can use the **no** keyword to disable DHCP snooping.
Step 2	`Switch(config)#` **`ip dhcp snooping vlan`** *`number`* `[`*`number`*`] |` **`vlan`** `{`*`vlan range`*`}]`	Enables DHCP snooping on your VLAN or VLAN range
Step 3	`Switch(config-if)#` **`ip dhcp snooping trust`**	Configures the interface as trusted or untrusted. You can use the **no** keyword to configure an interface to receive messages from an untrusted client.
Step 4	`Switch(config-if)#` **`ip dhcp snooping limit rate`** *`rate`*	Configures the number of DHCP packets per second (pps) that an interface can receive.[1]
Step 5	`Switch(config)#` **`end`**	Exits configuration mode.
Step 6	`Switch#` **`show ip dhcp snooping`**	Verifies the configuration.

[1] Cisco recommends not configuring the untrusted interface rate limit to more than 100 packets per second. The recommended rate limit for each untrusted client is 15 packets per second. Normally, the rate limit applies to untrusted interfaces. If you want to set up rate limiting for trusted interfaces, keep in mind that trusted interfaces aggregate all DHCP traffic in the switch, and you will need to adjust the rate limit to a higher value. You should fine tune this threshold depending on the network configuration. The CPU should not receive DHCP packets at a sustained rate of more than 1,000 packets per second

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 and range of VLANs.

This example shows how to enable DHCP Snooping on Vlan 500 through 555:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 500 555
Switch(config)# ip dhcp snooping information option format remote-id string switch123
Switch(config)# interface GigabitEthernet 5/1
Switch(config-if)# ip dhcp snooping trust
Switch(config-if)# ip dhcp snooping limit rate 100
Switch(config-if)# ip dhcp snooping vlan 555 information option format-type circuit-id
string customer-555
Switch(config-if)# interface FastEthernet 2/1
Switch(config-if)# ip dhcp snooping vlan 555 information option format-type circuit-id
string customer-500
Switch(config)# end
Switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
500,555
DHCP snooping is operational on following VLANs:
500,555
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is enabled
    circuit-id default format: vlan-mod-port
    remote-id: switch123 (string)
```

```
Option 82 on untrusted port is not allowed Verification of hwaddr field is enabled DHCP
snooping trust/rate is configured on the following Interfaces:

Interface                    Trusted     Rate limit (pps)
 ------------------------     -------     ----------------
FastEthernet5/1              yes         100
  Custom circuit-ids:
    VLAN 555: customer-555
FastEthernet2/1              no          unlimited
  Custom circuit-ids:
    VLAN 500: customer-500

Switch#
```

The following configuration describes the DHCP snooping configuration steps if routing is defined on another Catalyst switch (for example, a Catalyst 6500 series switch):

```
// Trust the uplink gigabit Ethernet trunk port

interface range GigabitEthernet 1/1 – 2
switchport mode trunk
switchport trunk encapsulation dot1q
ip dhcp snooping trust

!

interface VLAN 14
ip address 10.33.234.1 255.255.254.0
ip helper-address 10.5.1.2
```

Note If you are enabling trunking on uplink gigabit interfaces, and the above routing configuration is defined on a Catalyst 6500 series switch, you must configure the “trust” relationship with downstream DHCP Snooping (on a Catalyst 4500 series switch) which adds Option 82. On a Catalyst 6500 series switch, this task is accomplished with **ip dhcp relay information trusted** VLAN configuration command.

Enabling DHCP Snooping on the Aggregration Switch

To enable DHCP Snooping on an aggregation switch, configure the interface connecting to a downstream switch as a snooping untrusted port. If the downstream switch (or a device such as a DSLAM in the path between the aggregation switch and the DHCP clients) adds DHCP information option 82 to the DHCP packets, the DHCP packets would be dropped on arriving on a snooping untrusted port. Configuring the **ip dhcp snooping information option allow-untrusted** global configuration command on the aggregation switch would allow the aggregation switch to accept DHCP requests with option 82 information from any snooping untrusted port.

Enabling DHCP Snooping and Option 82

To enable DHCP snooping and Option 82 on the switch, perform the following steps:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`ip dhcp snooping`**	Enables DHCP snooping globally.

	Command	Purpose
Step 3	Switch(config)# **ip dhcp snooping vlan** *vlan-range*	Enables DHCP snooping on a VLAN or range of VLANs. The range is 1 to 4094. You can enter a single VLAN ID identified by VLAN ID number, a series of VLAN IDs separated by commas, a range of VLAN IDs separated by hyphens, or a range of VLAN IDs separated by entering the starting and ending VLAN IDs separated by a space.
Step 4	Switch(config)# **ip dhcp snooping information option**	Enables the switch to insert and remove DHCP relay information (option-82 field) in forwarded DHCP request messages to the DHCP server. This is the default setting.
Step 5	Switch(config)# **ip dhcp snooping information option format remote-id** [**string** *ASCII-string* | hostname]	(Optional) Configures the remote-ID suboption. You can configure the remote ID to be: • String of up to 63 ASCII characters (no spaces) • Configured hostname for the switch **Note** If the hostname is longer than 63 characters, it is truncated to 63 characters in the remote-ID configuration. The default remote ID is the switch MAC address.
Step 6	Switch(config)# **ip dhcp snooping information option allow-untrusted**	(Optional) If the switch is an aggregation switch connected to an edge switch, enables the switch to accept incoming DHCP snooping packets with option-82 information from the edge switch. The default setting is disabled. **Note** Enter this command only on aggregation switches that are connected to trusted devices.
Step 7	Switch(config)# **interface** *interface-id*	Specifies the interface to be configured, and enter interface configuration mode.
Step 8	Switch(config-if)# **ip dhcp snooping vlan** *vlan* **information option format-type circuit-id string** *ASCII-string*	(Optional) Configures the circuit-ID suboption for the specified interface. Specify the VLAN and port identifier, using a VLAN ID in the range of 1 to 4094. The default circuit ID is the port identifier, in the format **vlan-mod-port**. You can configure the circuit ID to be a string of 3 to 63 ASCII characters (no spaces).
Step 9	Switch(config-if)# **ip dhcp snooping trust**	(Optional) Configures the interface as trusted or untrusted. You can use the **no** keyword to configure an interface to receive messages from an untrusted client. The default setting is untrusted.
Step 10	Switch(config-if)# **ip dhcp snooping limit rate** *rate*	(Optional) Configures the number of DHCP packets per second that an interface can receive. The range is 1 to 2048. By default, no rate limit is configured. **Note** We recommend an untrusted rate limit of not more than 100 packets per second. If you configure rate limiting for trusted interfaces, you might need to increase the rate limit if the port is a trunk port assigned to more than one VLAN on which DHCP snooping is enabled.
Step 11	Switch(config-if)# **exit**	Returns to global configuration mode.

	Command	Purpose
Step 12	`Switch(config)#` **`ip dhcp snooping verify mac-address`**	(Optional) Configures the switch to verify that the source MAC address in a DHCP packet that is received on untrusted ports matches the client hardware address in the packet. The default is to verify that the source MAC address matches the client hardware address in the packet.
Step 13	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 14	`Switch#` **`show running-config`**	Verifies your entries.
Step 15	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To disable DHCP snooping, use the **no ip dhcp snooping** global configuration command. To disable DHCP snooping on a VLAN or range of VLANs, use the **no ip dhcp snooping vlan** *vlan-range* global configuration command. To disable the insertion and removal of the option-82 field, use the **no ip dhcp snooping information option** global configuration command. To configure an aggregation switch to drop incoming DHCP snooping packets with option-82 information from an edge switch, use the **no ip dhcp snooping information option allow-untrusted** global configuration command.

This example shows how to enable DHCP snooping globally and on VLAN 10 and to configure a rate limit of 100 packets per second on a port:

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10
Switch(config)# ip dhcp snooping information option
Switch(config)# interface gigabitethernet2/0/1
Switch(config-if)# ip dhcp snooping limit rate 100
```

Enabling DHCP Snooping on Private VLAN

DHCP snooping can be enabled on private VLANs, which provide isolation between Layer 2 ports within the same VLAN. If DHCP snooping is enabled (or disabled), the configuration is propagated to both the primary VLAN and its associated secondary VLANs. You cannot enable (or disable) DHCP snooping on a primary VLAN without reflecting this configuration change on the secondary VLANs.

Configuring DHCP snooping on a secondary VLAN is still allowed, but it does not take effect if the associated primary VLAN is already configured. If the associated primary VLAN is configured, the effective DHCP snooping mode on the secondary VLAN is derived from the corresponding primary VLAN. Manually configuring DHCP snooping on a secondary VLAN causes the switch to issue this warning message:

```
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.

Enabling the DHCP Snooping Database Agent

To configure the database agent, perform one or more of the following tasks:

Command	Purpose
Switch(config)# **ip dhcp snooping database {** *url* **|** **write-delay** *seconds* **|** **timeout** *seconds* **}** Switch(config)# **no ip dhcp snooping database [write-delay | timeout]**	(Required) Configures a URL for the database agent (or file) and the related timeout values.
Switch# **show ip dhcp snooping database [detail]**	(Optional) Displays the current operating state of the database agent and statistics associated with the transfers.
Switch# **clear ip dhcp snooping database statistics**	(Optional) Clears the statistics associated with the database agent.
Switch# **renew ip dhcp snooping database [validation none] [***url***]**	(Optional) Requests the read entries from a file at the given URL.
Switch# **ip dhcp snooping binding** *mac-addr* **vlan** *vlan ipaddr* **interface** *ifname* **expiry** *lease-in-seconds* Switch# **no ip dhcp snooping binding** *mac-addr* **vlan** *vlan ipaddr* **interface** *ifname*	(Optional) Adds/deletes bindings to the snooping database.

Note Because both NVRAM and bootflash have limited storage capacity, you should use TFTP or network-based files. If you use flash to store the database file, new updates (by the agent) result in the creation of new files (flash fills quickly). Moreover, due to the nature of the filesystem used on the flash, a large number of files can cause slow access. When a file is stored in a remote location accessible through TFTP, an RPR/SSO standby supervisor engine can take over the binding list when a switchover occurs.

Note 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

The following examples show how to use the above commands.

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:

```
Switch# configure terminal
Switch(config)# ip dhcp snooping database tftp://10.1.1.1/directory/file
Switch(config)# end
Switch# show ip dhcp snooping database detail
Agent URL : tftp://10.1.1.1/directory/file
Write delay Timer : 300 seconds
Abort Timer : 300 seconds
```

```
Agent Running : No
Delay Timer Expiry : 7 (00:00:07)
Abort Timer Expiry : Not Running

Last Succeded Time : None
Last Failed Time : 17:14:25 UTC Sat Jul 7 2001
Last Failed Reason : Unable to access URL.

Total Attempts       :       21   Startup Failures :        0
Successful Transfers :        0   Failed Transfers :       21
Successful Reads     :        0   Failed Reads     :        0
Successful Writes    :        0   Failed Writes    :       21
Media Failures       :        0

First successful access: Read

Last ignored bindings counters :
Binding Collisions    :        0   Expired leases    :        0
Invalid interfaces    :        0   Unsupported vlans :        0
Parse failures        :        0
Last Ignored Time : None

Total ignored bindings counters:
Binding Collisions    :        0   Expired leases    :        0
Invalid interfaces    :        0   Unsupported vlans :        0
Parse failures        :        0

Switch#
```

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 the read or create of the file has failed upon bootup.

Note Because the location is based off in the network, you must create a temporary file on the TFTP server. You can create a temporary file on a typical UNIX workstation by creating a 0 byte file "file" in the directory "directory" that can be referenced by the TFTP server daemon. With some server implementations on UNIX workstations, the file should be provided with full (777) permissions for write access to the file.

DHCP snooping bindings are keyed on the MAC address and VLAN combination. Therefore, 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 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	`Switch# show ip dhcp snooping database`	Displays the DHCP snooping database agent statistics.
Step 2	`Switch# renew ip dhcp snoop data url`	Directs the switch to read the file from given URL.
Step 3	`Switch# show ip dhcp snoop data`	Displays the read status.
Step 4	`Switch# 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:

```
Switch# showb ip dhcp snooping database
Agent URL :
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : Not Running
Abort Timer Expiry : Not Running

Last Succeded Time : None
Last Failed Time : None
Last Failed Reason : No failure recorded.

Total Attempts       :        0   Startup Failures :        0
Successful Transfers :        0   Failed Transfers :        0
Successful Reads     :        0   Failed Reads     :        0
Successful Writes    :        0   Failed Writes    :        0
Media Failures       :        0
Switch#
Switch# 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.

Switch#
00:01:29: %DHCP_SNOOPING-6-AGENT_OPERATION_SUCCEEDED: DHCP snooping database Read
succeeded.
Switch#
Switch# show ip dhcp snoop data
Agent URL :
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : Not Running
Abort Timer Expiry : Not Running

Last Succeded Time : 15:24:34 UTC Sun Jul 8 2001
Last Failed Time : None
Last Failed Reason : No failure recorded.
```

```
Total Attempts       :        1   Startup Failures :        0
Successful Transfers :        1   Failed Transfers :        0
Successful Reads     :        1   Failed Reads     :        0
Successful Writes    :        0   Failed Writes    :        0
Media Failures       :        0
Switch#
Switch# 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
Switch#
Switch# clear ip dhcp snoop bind
Switch# show ip dhcp snoop bind
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
Switch#
```

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	`Switch#` **`show ip dhcp snooping binding`**	Views the DHCP snooping database
Step 2	`Switch#` **`ip dhcp snooping binding`** *`binding-id`* **`vlan`** *`vlan-id`* **`interface`** *`interface`* **`expiry`** *`lease-time`*	Adds the binding using the 'ip dhcp snooping' exec command
Step 3	`Switch#` **`show ip dhcp snooping binding`**	Checks the DHCP snooping database

This example shows how to manually add a binding to the DHCP snooping database:

```
Switch# show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
Switch#
Switch# ip dhcp snooping binding 1.1.1 vlan 1 1.1.1.1 interface gi1/1 expiry 1000

Switch# 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
Switch#
```

Displaying DHCP Snooping Information

You can display a DHCP snooping binding table and configuration information for all interfaces on a switch.

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 has its own DHCP snooping binding table.

This example shows how to display the DHCP snooping binding information for a switch:

```
Switch# 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
Switch#
```

Table 37-2 describes the fields in the **show ip dhcp snooping binding** command output.

Table 37-2 show ip dhcp snooping binding Command Output

Field	Description
MAC Address	Client hardware MAC address
IP Address	Client IP address assigned from the DHCP server
Lease (seconds)	IP address lease time
Type	Binding type; dynamic binding learned by dhcp-snooping or statically-configured binding.
VLAN	VLAN number of the client interface
Interface	Interface that connects to the DHCP client host

Displaying the DHCP Snooping Configuration

This example shows how to display the DHCP snooping configuration for a switch.

```
Switch# show ip dhcp snooping
Switch DHCP snooping is enabled.
DHCP Snooping is configured on the following VLANs:
    10 30-40 100 200-220
Insertion of option 82 is enabled
Option82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Interface           Trusted        Rate limit (pps)
---------           -------        ----------------
FastEthernet2/1     yes            10
FastEthernet3/1     yes            none
GigabitEthernet1/1  no             20
Switch#
```

Overview of IP Source Guard

Similar to DHCP snooping, this feature is enabled on a DHCP snooping untrusted Layer 2 port. Initially, all IP traffic on the port is blocked except for DHCP packets that are captured by the DHCP snooping process. When a client receives a valid IP address from the DHCP server, or when a static IP source binding is configured by the user, a per-port and VLAN Access Control List (PVACL) is installed on the port. This process restricts the client IP traffic to those source IP addresses configured in the binding; any IP traffic with a source IP address other than that in the IP source binding is filtered out. This filtering limits a host's ability to attack the network by claiming a neighbor host's IP address.

Note If IP Source Guard is enabled on a trunk port with a large number of VLANs that have DHCP snooping enabled, you might run out of ACL hardware resources, and some packets might be switched in software instead.

Note When IP Source Guard is enabled, you might want to designate an alternative scheme for ACL hardware programming. For more information, see the "TCAM Programming and ACLs" section in the "Configuring Network Security with ACLs" chapter.

IP Source Guard supports the Layer 2 port only, including both access and trunk. For each untrusted Layer 2 port, there are two levels of IP traffic security filtering:

- Source IP address filter

  IP traffic is filtered based on its source IP address. Only IP traffic with a source IP address that matches the IP source binding entry is permitted.

  An IP source address filter is changed when a new IP source entry binding is created or deleted on the port. The port PVACL is recalculated and reapplied in the hardware to reflect the IP source binding change. By default, if the IP filter is enabled without any IP source binding on the port, a default PVACL that denies all IP traffic is installed on the port. Similarly, when the IP filter is disabled, any IP source filter PVACL is removed from the interface.

- Source IP and MAC address filter

  IP traffic is filtered based on its source IP address as well as its MAC address; only IP traffic with source IP and MAC addresses matching the IP source binding entry are permitted.

Note When IP source guard is enabled in IP and MAC filtering mode, the DHCP snooping option 82 must be enabled to ensure that the DHCP protocol works properly. Without option 82 data, the switch cannot locate the client host port to forward the DHCP server reply. Instead, the DHCP server reply is dropped, and the client cannot obtain an IP address.

Configuring IP Source Guard on the Switch

To enable IP Source Guard, perform this task:

	Command	Purpose
Step 1	`Switch(config)# ip dhcp snooping`	Enables DHCP snooping globally. You can use the **no** keyword to disable DHCP snooping.
Step 2	`Switch(config)# ip dhcp snooping vlan number [number]`	Enables DHCP snooping on your VLANs.
Step 3	`Switch(config-if)# no ip dhcp snooping trust`	Configures the interface as trusted or untrusted. You can use the **no** keyword of to configure an interface to receive only messages from within the network.

	Command	Purpose
Step 4	`Switch(config-if)#` **`ip verify source vlan dhcp-snooping port-security`**	Enables IP source guard, source IP, and source MAC address filtering on the port.
Step 5	`Switch(config-if)#` **`switchport port-security limit rate invalid-source-mac N`**	Enables security rate limiting for learned source MAC addresses on the port. **Note** This limit only applies to the port where IP Source Guard is enabled as filtering both IP and MAC addresses.
Step 6	`Switch(config)#` **`ip source binding`** *`mac-address`* **`Vlan`** *`vlan-id ip-address`* **`interface`** *`interface-name`*	Configures a static IP binding on the port.
Step 7	`Switch(config)#` **`end`**	Exits configuration mode.
Step 8	`Switch#` **`show ip verify source interface`** *`interface-name`*	Verifies the configuration.

If you want to stop IP Source Guard with Static Hosts on an interface, use the following commands in interface configuration submode:

```
Switch(config-if)# no ip verify source
Switch(config-if)# no ip device tracking max"
```

If "no ip device tracking" is used in the interface configuration submode, this command will be interpreted and run in the global configuration mode in fact and it causes IP device tracking to be disabled globally. For all the interfaces with the following command - "ip verify source tracking [port-security]", disabling IP device tracking globally will cause the IP Source Guard with Static Hosts denies all the IP traffic from those interfaces.

Note The static IP source binding can only be configured on switch port. If you issue the **ip source binding vlan interface** command on a Layer 3 port, you receive this error message: `Static IP source binding can only be configured on switch port.`

This example shows how to enable per-Layer 2-port IP source guard on VLANs 10 through 20:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10 20
Switch(config)# interface fa6/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk native vlan 10
Switch(config-if)# switchport trunk allowed vlan 11-20
Switch(config-if)# no ip dhcp snooping trust
Switch(config-if)# ip verify source vlan dhcp-snooping
Switch(config)# end
Switch# show ip verify source interface f6/1
Interface  Filter-type  Filter-mode  IP-address       Mac-address        Vlan
---------  -----------  -----------  ---------------  -----------------  ----------
Fa6/1      ip-mac       active       10.0.0.1                            10
Fa6/1      ip-mac       active       deny-all                            11-20
Switch#
```

The output shows that there is one valid DHCP binding to VLAN 10.

Configuring IP Source Guard on Private VLANs

For private VLAN ports, you must enable DHCP snooping on primary VLANs in order for IP source guard to be effective. IP source guard on a primary VLAN is automatically propagate to a secondary VLAN. Configuring a static IP source binding on a secondary VLAN is allowed, but it does not take effect. When manually configuring a static IP source binding on a secondary VLAN, you receive the following warning:

Warning **IP source filter may not take effect on secondary vlan where IP source binding is configured. If private vlan feature is enabled, IP source filter on primary vlan will automatically propagate to all secondary vlans.**

Displaying IP Source Guard Information

You can display IP Source Guard PVACL information for all interfaces on a switch using the **show ip verify source** command.

- This example shows displayed PVACLs if DHCP snooping is enabled on VLAN 10 through 20, if interface fa6/1 is configured for IP filtering, and if there is an existing IP address binding 10.0.01 on VLAN 10:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/1      ip           active       10.0.0.1                         10
fa6/1      ip           active       deny-all                         11-20
```

Note The second entry shows that a default PVACL (deny all IP traffic) is installed on the port for those snooping-enabled VLANs that do not have a valid IP source binding.

- This example shows displayed PVACL for a trusted port:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/2      ip           inactive-trust-port
```

- This example shows displayed PVACL for a port in a VLAN not configured for DHCP snooping:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/3      ip           inactive-no-snooping-vlan
```

- This example shows displayed PVACLs for a port with multiple bindings configured for an IP/MAC filtering:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/4      ip-mac       active       10.0.0.2         aaaa.bbbb.cccc  10
fa6/4      ip-mac       active       11.0.0.1         aaaa.bbbb.cccd  11
fa6/4      ip-mac       active       deny-all         deny-all        12-20
```

- This example shows displayed PVACLs for a port configured for IP/MAC filtering but not for port security:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/5      ip-mac       active       10.0.0.3         permit-all      10
fa6/5      ip-mac       active       deny-all         permit-all      11-20
```

Note The MAC filter shows permit-all because port security is not enabled, so the MAC filter cannot apply to the port/VLAN and is effectively disabled. Always enable port security first.

- This example shows displayed error message when issuing the **show ip verify source** command on a port that does not have an IP source filter mode configured:

```
IP Source Guard is not configured on the interface fa6/6.
```

You can also use the **show ip verify source** command to display all interfaces on the switch that have IP source guard enabled:

```
Interface  Filter-type  Filter-mode  IP-address       Mac-address     Vlan
---------  -----------  -----------  ---------------  --------------  ---------
fa6/1      ip           active       10.0.0.1                         10
fa6/1      ip           active       deny-all                         11-20
fa6/2      ip           inactive-trust-port
fa6/3      ip           inactive-no-snooping-vlan
fa6/4      ip-mac       active       10.0.0.2         aaaa.bbbb.cccc  10
fa6/4      ip-mac       active       11.0.0.1         aaaa.bbbb.cccd  11
fa6/4      ip-mac       active       deny-all         deny-all        12-20
fa6/5      ip-mac       active       10.0.0.3         permit-all      10
fa6/5      ip-mac       active       deny-all         permit-all      11-20
```

Displaying IP Source Binding Information

You can display all IP source bindings configured on all interfaces on a switch using the **show ip source binding** command.

```
Switch# show ip source binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:02:B3:3F:3B:99   55.5.5.2         6522        dhcp-snooping  10    FastEthernet6/10
00:00:00:0A:00:0B   11.0.0.1         infinite    static         10    FastEthernet6/10
Switch#
```

Table 37-3 describes the fields in the **show ip source binding** command output.

Table 37-3 show ip source binding Command Output

Field	Description
MAC Address	Client hardware MAC address
IP Address	Client IP address assigned from the DHCP server
Lease (seconds)	IP address lease time
Type	Binding type; static bindings configured from CLI to dynamic binding learned from DHCP Snooping

Table 37-3 show ip source binding Command Output (continued)

Field	Description
VLAN	VLAN number of the client interface
Interface	Interface that connects to the DHCP client host

Configuring IP Source Guard for Static Hosts

Note Supervisor Engine 6-E does *not* support this feature.

Note IPSG for Static Hosts should not be used on uplink ports.

IP Source Guard (IPSG) for static hosts extends the IPSG capability to non-DHCP and static environments. The existing IP Source Guard (IPSG) feature uses the entries created by the DHCP snooping feature to validate the hosts connected to a switch. Any traffic received from a host without a valid DHCP binding entry is dropped. In essence, a DHCP environment is a prerequisite for IPSG to work. The IPSG for static hosts feature removes IPSG's dependency on DHCP. The switch creates static entries based on ARP requests or other IP packets and uses them to maintain the list of valid hosts for a given port. In addition, the user can specify the number of hosts that would be allowed to send traffic to a given port. This is equivalent to port-security at Layer 3.

Note Some IP hosts with multiple network interfaces may inject some invalid packets into a network interface. Those invalid packets contain the IP/MAC address for another network interface of that host as the source address. It may cause IIPSG for static hosts in the switch, which connects to the host, to learn the invalid IP/MAC address bindings and reject the valid bindings. You should consult the vender of the corresponding OS and/or the network device of that host to prevent it from injecting invalid packets.

IPSG for Static Hosts initially learns IP/MAC bindings dynamically through an ACL-based snooping mechanism. IP/MAC bindings are learned from static hosts via ARP and IP packets and are stored using the device tracking database. Once the number of IP addresses that have been dynamically learned or statically configured on a given port reaches a maximum limit, any packet with a new IP address is dropped in hardware. To handle hosts that have moved or gone away for any reason, the IPSG for Static Hosts feature leverages IP device tracking functionality to age out dynamically learned IP address bindings. This feature can be used in conjunction with DHCP snooping. Multiple bindings will be established on a port that is connected to both DHCP and static hosts (i.e. bindings will be stored in both the device tracking database as well as the DHCP snooping binding database).

Topics include:

- IPSG for Static Hosts on a Layer 2 Access Port, page 37-21
- IPSG for Static Hosts on a PVLAN Host Port, page 37-24

IPSG for Static Hosts on a Layer 2 Access Port

You can configure IPSG for Static Hosts on a Layer 2 Access Port.

To enable IPSG for Static Hosts with IP filters on a Layer 2 access port, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`ip device tracking`**	Turns on the IP host table.
Step 2	`Switch(config)#` **`interface fastEthernet`** *`<a/b>`*	Enters IP configuration mode.
Step 3	`Switch(config-if)#` **`switchport mode access`**	Configures a port as access.
Step 4	`Switch(config-if)#` **`switchport access vlan`** *`<n>`*	Configures the VLAN for this port.
Step 5	`Switch(config-if)#` **`ip device tracking maximum`** *`<n>`*	Establishes a maximum limit for the bindings on this port. Upper bound for the maximum is 10.
Step 6	`Switch(config-if)#` **`switchport port-security`**	(Optional) Activates Port Security for this port.
Step 7	`Switch(config-if)#` **`switchport port-security maximum`** *`<n>`*	(Optional) Establishes a maximum number of MAC addresses for this port.
Step 8	`Switch(config-if)#` **`ip verify source tracking`** `[`**`port-security`**`]`	Activates IPSG for Static Hosts on this port.
Step 9	`Switch(config-if)#` **`end`**	Exits configuration interface mode.
Step 10	`Switch#` **`show ip verify source`** *`interface-name`*	Verifies the configuration.
Step 11	`Switch#` **`show ip device track all`** `[`**`active`** `|` **`inactive`**`]` **`count`**	Verifies the configuration by displaying the IP-to-MAC binding for a given host on the switch interface. • **all active** - displays only the active IP/MAC binding entries • **all inactive** - displays only the inactive IP/MAC binding entries • **all** - displays the active and inactive IP/MAC binding entries

To stop IPSG with Static Hosts on an interface, use the following commands in interface configuration submode:

```
Switch(config-if)# no ip verify source
Switch(config-if)# no ip device tracking max"
```

To enable IPSG with Static Hosts on a port, issue the following commands:

```
Switch(config)# ip device tracking ****enable IP device tracking globally
Switch(config)# ip device tracking max <n> ****set an IP device tracking maximum on int
Switch(config-if)# ip verify source tracking [port-security] ****activate IPSG on the port
```

Caution If you only configure the **ip verify source tracking [port-security]** interface configuration command on a port without enabling IP device tracking globally or setting an IP device tracking maximum on that interface, IPSG with Static Hosts will reject all the IP traffic from that interface.

Note The issue above also applies to IPSG with Statis Hosts on a PVLAN Host port.

This example shows how to enable IPSG for Static Hosts with IP filters on a Layer 2 access port and to verify the three valid IP bindings on the interface Fa4/3:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip device tracking
Switch(config)# interface fastEthernet 4/3
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# ip device tracking maximum 5
Switch(config-if)# ip verify source tracking
Switch(config-if)# end

Switch# show ip verify source
Interface  Filter-type  Filter-mode  IP-address       Mac-address        Vlan
---------  -----------  -----------  ---------------  -----------------  ----
Fa4/3      ip trk       active       40.1.1.24                           10
Fa4/3      ip trk       active       40.1.1.20                           10
Fa4/3      ip trk       active       40.1.1.21                           10
```

The following example shows how to enable IPSG for Static Hosts with IP-Mac filters on a Layer 2 access port, to verify the five valid IP-MAC bindings on the interface Fa4/3, and to verify that the number of bindings on this interface has reached the maximum limit:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip device tracking
Switch(config)# interface fastEthernet 4/3
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 1
Switch(config-if)# ip device tracking maximum 5
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 5
Switch(config-if)# ip verify source tracking port-security
Switch(config-if)# end

Switch# show ip verify source
Interface  Filter-type  Filter-mode  IP-address       Mac-address        Vlan
---------  -----------  -----------  ---------------  -----------------  ----
Fa4/3      ip-mac trk   active       40.1.1.24        00:00:00:00:03:04  1
Fa4/3      ip-mac trk   active       40.1.1.20        00:00:00:00:03:05  1
Fa4/3      ip-mac trk   active       40.1.1.21        00:00:00:00:03:06  1
Fa4/3      ip-mac trk   active       40.1.1.22        00:00:00:00:03:07  1
Fa4/3      ip-mac trk   active       40.1.1.23        00:00:00:00:03:08  1
```

The following example displays all IP/MAC binding entries for all interfaces. Observe that the CLI displays all active as well as inactive entries. When a host is learned on a interface, the new entry is marked as active. When the same host is disconnected from the current interface and connected to a different interface, a new IP/AC binding entry is displayed as active as soon as the host is detected. The old entry for this host on the previous interface is now marked as inactive.

```
Switch# show ip device tracking all
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
---------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface              STATE
---------------------------------------------------------------------
200.1.1.8       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.9       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.10      0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
```

```
200.1.1.1       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.1       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.2       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.2       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.3       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.3       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.4       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.4       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.5       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.5       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.6       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.7       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
```

The following example displays all active IP/MAC binding entries for all interfaces:

```
Switch# show ip device tracking all active
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
---------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface              STATE
---------------------------------------------------------------------
200.1.1.1       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.2       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.3       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.4       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
200.1.1.5       0001.0600.0000  9    GigabitEthernet4/1     ACTIVE
```

The following example displays all inactive IP/MAC binding entries for all interfaces. The host was first learned on GigabitEthernet 3/1 then moved to GigabitEthernet 4/1. So the IP/MAC binding entries learned on GigabitEthernet 3/1 are marked as inactive.

```
Switch# show ip device tracking all inactive
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
---------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface              STATE
---------------------------------------------------------------------
200.1.1.8       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.9       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.10      0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.1       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.2       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.3       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.4       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.5       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.6       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
200.1.1.7       0001.0600.0000  8    GigabitEthernet3/1     INACTIVE
```

The following example display the count of all IP device tracking host entries for all interfaces:

```
Switch# show ip device tracking all count
Total IP Device Tracking Host entries: 5
---------------------------------------------------------------------
  Interface          Maximum Limit          Number of Entries
---------------------------------------------------------------------
Fa4/3                 5
```

IPSG for Static Hosts on a PVLAN Host Port

You can configure IPSG for Static Hosts on a PVLAN host port.

To enable IPSG for Static Hosts with IP filters on a PVLAN host port, perform this task:

	Command	Purpose
Step 1	`Switch(config)# vlan <n1>`	Enters configuration VLAN mode.
Step 2	`Switch(config-vlan)# private-vlan primary`	Establishes a primary VLAN on a PVLAN port.
Step 3	`Switch(config-vlan)# exit`	Exits VLAN configuration mode.
Step 4	`Switch(config)# vlan <n2>`	Enters configuration VLAN mode.
Step 5	`Switch(config-vlan)# private-vlan isolated`	Establishes an isolated VLAN on a PVLAN port.
Step 6	`Switch(config-vlan)# exit`	Exits VLAN configuration mode.
Step 7	`Switch(config)# vlan <n1>`	Enters configuration VLAN mode.
Step 8	`Switch(config-vlan)# private-vlan association 201`	Associates the VLAN on an isolated PVLAN port.
Step 9	`Switch(config-vlan)# exit`	Exits VLAN configuration mode.
Step 10	`Switch(config)# interface fastEthernet <a/b>`	Enters interface configuration mode.
Step 11	`SSwitch(config-if)# switchport mode private-vlan host`	(Optional) Establishes a port as a PVLAN host.
Step 12	`SSwitch(config-if)# switchport private-vlan host-association <a> <b>`	(Optional)Associates this port with the corresponding PVLAN.
Step 13	`Switch(config-if)# ip device tracking maximum <n>`	Establishes a maximum limit for the bindings on this port.
Step 14	`Switch(config-if)# ip verify source tracking [port-security]`	Activiates IPSG for Static Hosts on this port.
Step 15	`Switch(config-if)# end`	Exits configuration interface mode.
Step 16	`Switch# show ip device tracking all`	Verifies the configuration.
Step 17	`Switch# show ip verify source interface-name`	Verifies the configuration.

This example shows how to enable IPSG for Static Hosts with IP filters on a PVLAN host port:

```
Switch(config)# vlan 200
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# exit
Switch(config)# vlan 201
Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# exit
Switch(config)# vlan 200
Switch(config-vlan)# private-vlan association 201
Switch(config-vlan)# exit
Switch(config)# int fastEthernet 4/3
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 200 201
Switch(config-if)# ip device tracking maximum 8
Switch(config-if)# ip verify source tracking

Switch# show ip device tracking all
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
---------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface             STATE
---------------------------------------------------------------------
40.1.1.24       0000.0000.0304  200  FastEthernet4/3       ACTIVE
40.1.1.20       0000.0000.0305  200  FastEthernet4/3       ACTIVE
40.1.1.21       0000.0000.0306  200  FastEthernet4/3       ACTIVE
40.1.1.22       0000.0000.0307  200  FastEthernet4/3       ACTIVE
40.1.1.23       0000.0000.0308  200  FastEthernet4/3       ACTIVE
```

The output shows the five valid IP-MAC bindings that have been learned on the interface Fa4/3. For the PVLAN cases, the bindings are associated with primary VLAN ID. So, in this example, the primary VLAN ID, 200, is shown in the table.

```
Switch# show ip verify source
Interface  Filter-type  Filter-mode  IP-address       Mac-address        Vlan
---------  -----------  -----------  ---------------  -----------------  ----
Fa4/3      ip trk       active       40.1.1.23                           200
Fa4/3      ip trk       active       40.1.1.24                           200
Fa4/3      ip trk       active       40.1.1.20                           200
Fa4/3      ip trk       active       40.1.1.21                           200
Fa4/3      ip trk       active       40.1.1.22                           200
Fa4/3      ip trk       active       40.1.1.23                           201
Fa4/3      ip trk       active       40.1.1.24                           201
Fa4/3      ip trk       active       40.1.1.20                           201
Fa4/3      ip trk       active       40.1.1.21                           201
Fa4/3      ip trk       active       40.1.1.22                           201
```

The output shows that the five valid IP-MAC bindings are on both the primary and secondary VLAN.

CHAPTER 38

Configuring Dynamic ARP Inspection

This chapter describes how to configure Dynamic ARP Inspection (DAI) on the Catalyst 4000 family switch.

This chapter includes the following major sections:

- Overview of Dynamic ARP Inspection, page 38-1
- Configuring Dynamic ARP Inspection, page 38-5

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) is a security feature that validates Address Resolution Protocol (ARP) packets in a network. DAI allows a network administrator to intercept, log, and discard ARP packets with invalid MAC-IP pairs. This capability protects the network from certain "man-in-the-middle" attacks.

This section contains the following subsections:

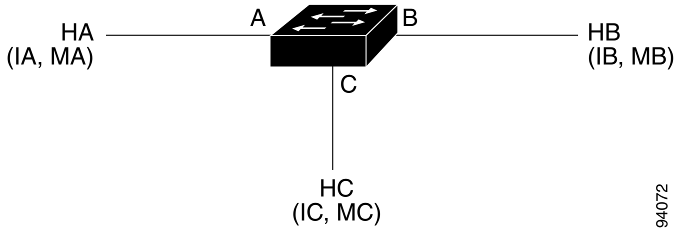
- ARP Cache Poisoning, page 38-2
- Purpose of Dynamic ARP Inspection, page 38-2
- Interface Trust State, Security Coverage and Network Configuration, page 38-3
- Relative Priority of Static Bindings and DHCP Snooping Entries, page 38-4
- Logging of Dropped Packets, page 38-4
- Rate Limiting of ARP Packets, page 38-4
- Port Channels and Their Behavior, page 38-4

ARP Cache Poisoning

You can attack hosts, switches, and routers connected to your Layer 2 network by “poisoning” their ARP caches. For example, a malicious user might intercept traffic intended for other hosts on the subnet by poisoning the ARP caches of systems connected to the subnet.

Consider the following configuration:

Figure 38-1 ARP Cache Poisoning



Hosts HA, HB, and HC 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 HA uses IP address IA and MAC address MA. When HA needs to communicate to HB at the IP Layer, HA broadcasts an ARP request for the MAC address associated with IB. As soon as HB receives the ARP request, the ARP cache on HB is populated with an ARP binding for a host with the IP address IA and a MAC address MA. When HB responds to HA, the ARP cache on HA is populated with a binding for a host with the IP address IB and a MAC address MB.

Host HC can “poison” the ARP caches of HA and HB 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 HC intercepts that traffic. Because HC knows the true MAC addresses associated with IA and IB, HC can forward the intercepted traffic to those hosts using the correct MAC address as the destination. HC has inserted itself into the traffic stream from HA to HB, the classic “man in the middle” attack.

Purpose of Dynamic ARP Inspection

To prevent ARP poisoning attacks, a switch must ensure that only valid ARP requests and responses are relayed. DAI prevents these attacks by intercepting all ARP requests and responses. Each of these intercepted packets is verified for valid MAC address to IP address bindings before the local ARP cache is updated or the packet is forwarded to the appropriate destination. Invalid ARP packets are dropped.

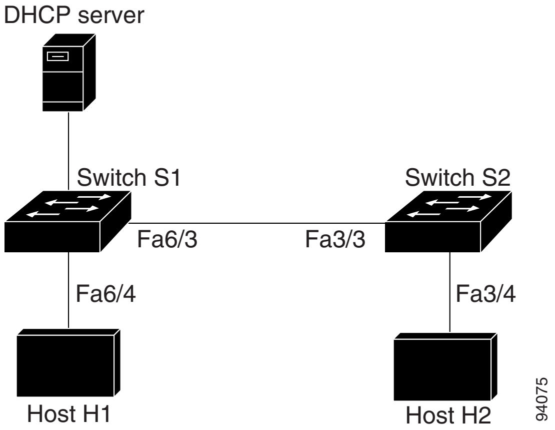
DAI determines the validity of an ARP packet based on valid MAC address to IP address bindings stored in a trusted database. This database is built at runtime by DHCP snooping, provided this feature is enabled on VLANs and on the switch. In addition, in order to handle hosts that use statically configured IP addresses, DAI can also validate ARP packets against user-configured ARP ACLs.

DAI can also be configured to drop ARP packets when the IP addresses in the packet are invalid or when the MAC addresses in the body of the ARP packet do not match the addresses specified in the Ethernet header.

Interface Trust State, Security Coverage and Network Configuration

DAI associates a trust state with each interface on the system. Packets arriving on trusted interfaces bypass all DAI validation checks, while those arriving on untrusted interfaces go through the DAI validation process. In a typical network configuration for DAI, all ports connected to host ports are configured as untrusted, while all ports connected to switches are configured as trusted. With this configuration, all ARP packets entering the network from a given switch pass the security check.

Figure 38-2 Validation of ARP Packets on a DAI-enabled VLAN



Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity. If we assume that both S1 and S2 (in Figure 38-2) run DAI on the VLAN ports that contains H1 and H2, and if H1 and H2 were to acquire their IP addresses from the DHCP server connected to S1, then only S1 binds the IP to MAC address of H1. Therefore, if the interface between S1 and S2 is untrusted, the ARP packets from H1 get dropped on S2. This condition would result in a loss of connectivity between H1 and H2.

Configuring interfaces to be trusted when they are actually untrusted leaves a security hole in the network. If S1 were not running DAI, then H1 can easily poison the ARP of S2 (and H2, if the inter-switch link is configured as trusted). This condition can occur even though S2 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. It does not, however, ensure that hosts from other portions of the network do not poison the caches of the hosts connected to it.

To handle cases in which some switches in a VLAN run DAI and other switches do not, the interfaces connecting such switches should be configured as untrusted. To validate the bindings of packets from non-DAI switches, however, the switch running DAI should be configured with ARP ACLs. When it is not feasible to determine such bindings, switches running DAI should be isolated from non-DAI switches at Layer 3.

Note Depending on the setup of the DHCP server and the network, it may not be possible to perform validation of a given ARP packet on all switches in the VLAN.

Relative Priority of Static Bindings and DHCP Snooping Entries

As mentioned previously, DAI populates its database of valid MAC address to IP address bindings through DHCP snooping. It also validates ARP packets against statically configured ARP ACLs. It is important to note that ARP ACLs have precedence over entries in the DHCP snooping database. ARP packets are first compared to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, then the packet is denied 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 the Log Buffer” section on page 38-14.

Rate Limiting of ARP Packets

DAI performs validation checks in the CPU, so the number of incoming ARP packets is rate-limited to prevent a denial of service attack. By default, the rate for untrusted interfaces is set to 15 pps second, whereas trusted interfaces have no rate limit. When the rate of incoming ARP packets exceeds the configured limit, the port is placed in the errdisable state. The port remains in that state until an administrator intervenes. With the **errdisable recovery** global configuration command, you can enable errdisable recovery so that ports emerge from this state automatically after a specified timeout period.

You use the **ip arp inspection limit** global configuration command to limit the rate of incoming ARP requests and responses on the interface. Unless a rate limit is explicitly configured on an interface, changing the trust state of the interface also changes its rate limit to the default value for that trust state; that is, 15 packets per second for untrusted interfaces and unlimited for trusted interfaces. Once a rate limit is configured explicitly, the interface retains the rate limit even when its trust state is changed. At any time, the interface reverts to its default rate limit if the *no* form of the **rate limit** command is applied. For configuration information, see the “Limiting the Rate of Incoming ARP Packets” section on page 38-16.

Port Channels and Their Behavior

A given physical port can join a channel only when the trust state of the physical port and of the channel match. Otherwise, the physical port remains suspended in the channel. A channel inherits its trust state from the first physical port that joined the channel. Consequently, the trust state of the first physical port need not match the trust state of the channel.

Conversely, when the trust state is changed on the channel, the new trust state is configured on all the physical ports that comprise the channel.

The rate limit check on port channels is unique. 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.

The rate limit is cumulative across all physical ports; that is, the rate of incoming packets on a port channel equals the sum of rates across all physical ports.

When you configure rate limits for ARP packets on trunks, you must account for VLAN aggregation because a high rate limit on one VLAN can cause a “denial of service” attack to other VLANs when the port is errdisabled by software. Similarly, when a port channel is errdisabled, a high rate limit on one physical port can cause other ports in the channel to go down.

Configuring Dynamic ARP Inspection

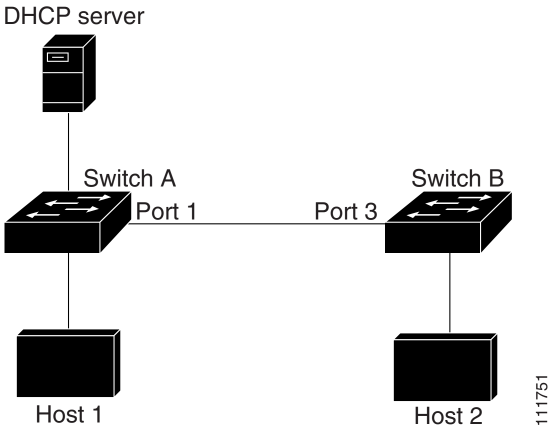
These sections describe how to configure dynamic ARP inspection on your switch:

- Configuring Dynamic ARP Inspection in DHCP Environments, page 38-5 (required)
- Configuring ARP ACLs for Non-DHCP Environments, page 38-10 (optional)
- Configuring the Log Buffer, page 38-14 (optional)
- Limiting the Rate of Incoming ARP Packets, page 38-16 (optional)
- Performing Validation Checks, page 38-19 (optional)

Configuring Dynamic ARP Inspection in DHCP Environments

This procedure shows how to configure dynamic ARP inspection 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 38-3. Both switches are running dynamic ARP inspection on VLAN 100 where the hosts are located. A DHCP server is connected to Switch A. Both hosts acquire their IP addresses from the same DHCP server. Therefore, Switch A has the bindings for Host 1, and Switch B has the bindings for Host 2.

Figure 38-3 ARP Packet Validation on a VLAN Enabled for Dynamic ARP Inspection



Note Dynamic ARP inspection depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses. For configuration information, see Chapter 37, "Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts."

For information on how to configure dynamic ARP inspection when only one switch supports the feature, see the "Configuring ARP ACLs for Non-DHCP Environments" section on page 38-10.

To configure dynamic ARP inspection, perform this task on both switches:

	Command	Purpose
Step 1	`Switch# show cdp neighbors`	Verifies the connection between the switches.
Step 2	`Switch# configure terminal`	Enters global configuration mode.
Step 3	`Switch(config)# [no] ip arp inspection vlan vlan-range`	Enables dynamic ARP inspection on a per-VLAN basis. By default, dynamic ARP inspection is disabled on all VLANs. To disable dynamic ARP inspection, use the **no ip arp inspection vlan** *vlan-range* global configuration command. For *vlan-range*, specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. Specify the same VLAN ID for both switches.
Step 4	`Switch(config)# interface interface-id`	Specifies the interface connected to the other switch, and enter interface configuration mode.
Step 5	`Switch(config-if)# ip arp inspection trust`	Configures the connection between the switches as trusted. To return the interfaces to an untrusted state, use the **no ip arp inspection trust** interface configuration command. By default, all interfaces are untrusted. The switch does not check ARP packets that it receives from the other switch on the trusted interface. It simply forwards the packets. For 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 the Log Buffer" section on page 38-14.
Step 6	`Switch(config-if)# end`	Returns to privileged EXEC mode.

Step 7	`Switch# show ip arp inspection interfaces`
	`Switch# show ip arp inspection vlan vlan-range`
		Verifies the dynamic ARP inspection configuration.

	Command	Purpose
Step 8	`Switch#` **`show ip dhcp snooping binding`**	Verifies the DHCP bindings.
Step 9	`Switch#` **`show ip arp inspection statistics vlan`** *`vlan-range`*	Checks the dynamic ARP inspection statistics.
Step 10	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

This example shows how to configure dynamic ARP inspection on Switch A in VLAN 100. You would perform a similar procedure on Switch B.

On Switch A

```
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             Gig 3/48              179           R S I     WS-C4506  Gig 3/46
SwitchA# configure terminal
SwitchA(config)# ip arp inspection vlan 100
SwitchA(config)# interface g3/48
SwitchA(config-if)# ip arp inspection trust
SwitchA(config-if)# end
SwitchA# show ip arp inspection interfaces

 Interface        Trust State     Rate (pps)    Burst Interval
 ---------------  -----------     ----------    --------------
 Gi1/1            Untrusted               15                 1
 Gi1/2            Untrusted               15                 1
 Gi3/1            Untrusted               15                 1
 Gi3/2            Untrusted               15                 1
 Gi3/3            Untrusted               15                 1
 Gi3/4            Untrusted               15                 1
 Gi3/5            Untrusted               15                 1
 Gi3/6            Untrusted               15                 1
 Gi3/7            Untrusted               15                 1
 Gi3/8            Untrusted               15                 1
 Gi3/9            Untrusted               15                 1
 Gi3/10           Untrusted               15                 1
 Gi3/11           Untrusted               15                 1
 Gi3/12           Untrusted               15                 1
 Gi3/13           Untrusted               15                 1
 Gi3/14           Untrusted               15                 1
 Gi3/15           Untrusted               15                 1
 Gi3/16           Untrusted               15                 1
 Gi3/17           Untrusted               15                 1
 Gi3/18           Untrusted               15                 1
 Gi3/19           Untrusted               15                 1
 Gi3/20           Untrusted               15                 1
 Gi3/21           Untrusted               15                 1
 Gi3/22           Untrusted               15                 1
 Gi3/23           Untrusted               15                 1
 Gi3/24           Untrusted               15                 1
 Gi3/25           Untrusted               15                 1
 Gi3/26           Untrusted               15                 1
 Gi3/27           Untrusted               15                 1
 Gi3/28           Untrusted               15                 1
 Gi3/29           Untrusted               15                 1
 Gi3/30           Untrusted               15                 1
 Gi3/31           Untrusted               15                 1
 Gi3/32           Untrusted               15                 1
 Gi3/33           Untrusted               15                 1
 Gi3/34           Untrusted               15                 1
 Gi3/35           Untrusted               15                 1
 Gi3/36           Untrusted               15                 1
 Gi3/37           Untrusted               15                 1
 Gi3/38           Untrusted               15                 1
 Gi3/39           Untrusted               15                 1
 Gi3/40           Untrusted               15                 1
 Gi3/41           Untrusted               15                 1
 Gi3/42           Untrusted               15                 1
 Gi3/43           Untrusted               15                 1
 Gi3/44           Untrusted               15                 1
 Gi3/45           Untrusted               15                 1
 Gi3/46           Untrusted               15                 1
 Gi3/47           Untrusted               15                 1
 Gi3/48           Trusted               None               N/A

SwitchA# show ip arp inspection vlan 100
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
  100     Enabled          Active

 Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
  100     Deny             Deny

SwitchA# show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:01:00:01:00:01   170.1.1.1        3597        dhcp-snooping  100   GigabitEthernet3/27
Total number of bindings: 1

SwitchA# show ip arp inspection statistics vlan 100

 Vlan      Forwarded        Dropped     DHCP Drops      ACL Drops
 ----      ---------        -------     ----------      ---------
  100             15              0              0              0

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
  100              0              0                     0

 Vlan   Dest MAC Failures   IP Validation Failures   Invalid Protocol Data
 ----   -----------------   ----------------------   ---------------------
  100                   0                        0                       0
SwitchA#
```

On Switch B

```
SwitchB# 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
SwitchA             Gig 3/46              163           R S I     WS-C4507R Gig 3/48
SwitchB#
SwitchB# configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchB(config)# ip arp inspection vlan 100
SwitchB(config)# interface g3/46
SwitchB(config-if)# ip arp inspection trust
SwitchB(config-if)# end
SwitchB#
SwitchB# show ip arp inspection interfaces

 Interface        Trust State     Rate (pps)    Burst Interval
 ---------------  -----------     ----------    --------------
 Gi1/1            Untrusted               15                 1
 Gi1/2            Untrusted               15                 1
 Gi3/1            Untrusted               15                 1
 Gi3/2            Untrusted               15                 1
 Gi3/3            Untrusted               15                 1
 Gi3/4            Untrusted               15                 1
 Gi3/5            Untrusted               15                 1
 Gi3/6            Untrusted               15                 1
 Gi3/7            Untrusted               15                 1
 Gi3/8            Untrusted               15                 1
 Gi3/9            Untrusted               15                 1
 Gi3/10           Untrusted               15                 1
 Gi3/11           Untrusted               15                 1
 Gi3/12           Untrusted               15                 1
 Gi3/13           Untrusted               15                 1
 Gi3/14           Untrusted               15                 1
 Gi3/15           Untrusted               15                 1
 Gi3/16           Untrusted               15                 1
 Gi3/17           Untrusted               15                 1
 Gi3/18           Untrusted               15                 1
 Gi3/19           Untrusted               15                 1
 Gi3/20           Untrusted               15                 1
 Gi3/21           Untrusted               15                 1
 Gi3/22           Untrusted               15                 1
 Gi3/23           Untrusted               15                 1
 Gi3/24           Untrusted               15                 1
 Gi3/25           Untrusted               15                 1
 Gi3/26           Untrusted               15                 1
 Gi3/27           Untrusted               15                 1
 Gi3/28           Untrusted               15                 1
 Gi3/29           Untrusted               15                 1
 Gi3/30           Untrusted               15                 1
 Gi3/31           Untrusted               15                 1
 Gi3/32           Untrusted               15                 1
 Gi3/33           Untrusted               15                 1
 Gi3/34           Untrusted               15                 1
 Gi3/35           Untrusted               15                 1
 Gi3/36           Untrusted               15                 1
 Gi3/37           Untrusted               15                 1
 Gi3/38           Untrusted               15                 1
 Gi3/39           Untrusted               15                 1
 Gi3/40           Untrusted               15                 1
 Gi3/41           Untrusted               15                 1
 Gi3/42           Untrusted               15                 1
 Gi3/43           Untrusted               15                 1
 Gi3/44           Untrusted               15                 1
 Gi3/45           Untrusted               15                 1
 Gi3/46           Trusted               None               N/A
 Gi3/47           Untrusted               15                 1
 Gi3/48           Untrusted               15                 1

SwitchB# show ip arp inspection vlan 100
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
```

```
IP Address Validation      : Disabled

 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
  100     Enabled          Active

 Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
  100     Deny             Deny#

SwitchB# show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:02:00:02:00:02   170.1.1.2        3492        dhcp-snooping  100   GigabitEthernet3/31
Total number of bindings: 1

SwitchB# show ip arp insp statistics vlan 100

 Vlan      Forwarded        Dropped     DHCP Drops      ACL Drops
 ----      ---------        -------     ----------      ---------
  100           2398              0              0              0

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
  100           2398              0                     0

 Vlan   Dest MAC Failures   IP Validation Failures   Invalid Protocol Data
 ----   -----------------   ----------------------   ---------------------
  100                   0                        0                       0
SwitchB#
```

Configuring ARP ACLs for Non-DHCP Environments

This procedure shows how to configure dynamic ARP inspection when Switch B shown in Figure 38-3 on page 38-5 does not support dynamic ARP inspection or DHCP snooping.

If you configure port 1 on Switch A as trusted, a security hole is created because both Switch A and Host 1 could be attacked by either Switch B or Host 2. To prevent this possibility, you must configure port 1 on Switch A as untrusted. To permit ARP packets from Host 2, you must set up an ARP ACL and apply it to VLAN 100. If the IP address of Host 2 is not static, such that it is 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 configure an ARP ACL (on switch A in a non-DHCP environment), perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# arp access-list acl-name`	Defines an ARP ACL, and enter ARP access-list configuration mode. By default, no ARP access lists are defined. **Note** At the end of the ARP access list, there is an implicit **deny ip any mac any** command.

	Command	Purpose
Step 3	`Switch(config-arp-nac)# permit ip host sender-ip mac host sender-mac [log]`	Permits ARP packets from the specified host (Host 2). • For *sender-ip*, enter the IP address of Host 2. • For *sender-mac*, enter the MAC address of Host 2. • (Optional) Specify **log** to log a packet in the log buffer when it matches the access control entry (ACE). Matches are logged if you also configure the **matchlog** keyword in the **ip arp inspection vlan logging** global configuration command. For more information, see the "Configuring the Log Buffer" section on page 38-14.
Step 4	`Switch(config-arp-nac)# exit`	Returns to global configuration mode.
Step 5	`Switch(config)# ip arp inspection filter arp-acl-name vlan vlan-range [static]`	Applies the ARP ACL to the VLAN. By default, no defined ARP ACLs are applied to any VLAN. • For *arp-acl-name*, specify the name of the ACL created in Step 2. • For *vlan-range*, specify the VLAN that the switches and hosts are in. You can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. • (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.
Step 6	`Switch(config)# interface interface-id`	Specifies the Switch A interface that is connected to Switch B, and enter interface configuration mode.

	Command	Purpose
Step 7	Switch(config-if)# **no ip arp inspection trust**	Configures the Switch A interface that is connected to Switch B as untrusted. By default, all interfaces are untrusted. For 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 the Log Buffer" section on page 38-14.
Step 8	Switch(config-if)# **end**	Returns to privileged EXEC mode.
Step 9	Switch# **show arp access-list** [*acl-name*] Switch# **show ip arp inspection vlan** *vlan-range* Switch# **show ip arp inspection interfaces**	Verifies the dynamic ARP inspection configuration.
Step 10	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To remove the ARP ACL, use the **no arp access-list** global configuration command. To remove the ARP ACL attached to a VLAN, use the **no ip arp inspection filter** *arp-acl-name* **vlan** *vlan-range* global configuration command.

This example shows how to configure an ARP ACL called *host2* on Switch A, to permit ARP packets from HostB (IP address 170.1.1.2 and MAC address 2.2.2), to apply the ACL to VLAN 100, and to configure port 1 on Switch A as untrusted:

```
SwitchA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)# arp access-list hostB
SwitchA(config-arp-nacl)# permit ip host 170.1.1.2 mac host 2.2.2 log
SwitchA(config-arp-nacl)# exit
SwitchA(config)# ip arp inspection filter hostB vlan 100 static
SwitchA(config)# interface g3/48
SwitchA(config-if)# no ip arp inspection trust
SwitchA(config-if)# end
SwitchA# show arp access-list hostB
ARP access list hostB
    permit ip host 170.1.1.2 mac host 0002.0002.0002 log

SwitchA# show ip arp inspection interfaces

 Interface        Trust State     Rate (pps)    Burst Interval
 ---------------  -----------     ----------    --------------
 Gi1/1            Untrusted               15                 1
 Gi1/2            Untrusted               15                 1
 Gi3/1            Untrusted               15                 1
 Gi3/2            Untrusted               15                 1
 Gi3/3            Untrusted               15                 1
 Gi3/4            Untrusted               15                 1
 Gi3/5            Untrusted               15                 1
 Gi3/6            Untrusted               15                 1
 Gi3/7            Untrusted               15                 1
 Gi3/8            Untrusted               15                 1
 Gi3/9            Untrusted               15                 1
 Gi3/10           Untrusted               15                 1
 Gi3/11           Untrusted               15                 1
 Gi3/12           Untrusted               15                 1
 Gi3/13           Untrusted               15                 1
 Gi3/14           Untrusted               15                 1
 Gi3/15           Untrusted               15                 1
 Gi3/16           Untrusted               15                 1
 Gi3/17           Untrusted               15                 1
 Gi3/18           Untrusted               15                 1
 Gi3/19           Untrusted               15                 1
 Gi3/20           Untrusted               15                 1
 Gi3/21           Untrusted               15                 1
 Gi3/22           Untrusted               15                 1
 Gi3/23           Untrusted               15                 1
 Gi3/24           Untrusted               15                 1
 Gi3/25           Untrusted               15                 1
 Gi3/26           Untrusted               15                 1
 Gi3/27           Untrusted               15                 1
 Gi3/28           Untrusted               15                 1
 Gi3/29           Untrusted               15                 1
 Gi3/30           Untrusted               15                 1
 Gi3/31           Untrusted               15                 1
 Gi3/32           Untrusted               15                 1
 Gi3/33           Untrusted               15                 1
 Gi3/34           Untrusted               15                 1
 Gi3/35           Untrusted               15                 1
 Gi3/36           Untrusted               15                 1
 Gi3/37           Untrusted               15                 1
 Gi3/38           Untrusted               15                 1
 Gi3/39           Untrusted               15                 1
 Gi3/40           Untrusted               15                 1
 Gi3/41           Untrusted               15                 1
 Gi3/42           Untrusted               15                 1
 Gi3/43           Untrusted               15                 1
 Gi3/44           Untrusted               15                 1
 Gi3/45           Untrusted               15                 1
 Gi3/46           Untrusted               15                 1
 Gi3/47           Untrusted               15                 1
 Gi3/48           Untrusted               15                 1

SwitchA# show ip arp inspection statistics vlan 100

 Vlan      Forwarded        Dropped     DHCP Drops      ACL Drops
 ----      ---------        -------     ----------      ---------
  100             15            169            160              9

 Vlan   DHCP Permits    ACL Permits   Source MAC Failures
 ----   ------------    -----------   -------------------
  100              0              0                     0

 Vlan   Dest MAC Failures   IP Validation Failures   Invalid Protocol Data
 ----   -----------------   ----------------------   ---------------------
  100                   0                        0                       0
SwitchA#
```

Configuring the Log Buffer

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.

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, the switch 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. No other statistics are provided for the entry.

To configure the log buffer, perform this task beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Switch# `**`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)# `**`ip arp inspection log-buffer`** `{`**`entries`** *`number`* `|` **`logs`** *`number`* **`interval`** *`seconds`*`}`	Configures the dynamic ARP inspection logging buffer. By default, when dynamic ARP inspection is enabled, denied or dropped ARP packets are logged. The number of log entries is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second. The keywords have these meanings: • For **entries** *number*, specify the number of entries to be logged in the buffer. The range is 0 to 1024. • For **logs** *number* **interval** *seconds*, specify the number of entries to generate system messages in the specified interval. For **logs** *number*, 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** *seconds*, 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. The **logs** and **interval** settings interact. If the **logs** *number* X is greater than **interval** *seconds* Y, X divided by Y (X/Y) system messages are sent every second. Otherwise, one system message is sent every Y divided by X (Y/X) seconds.

	Command	Purpose
Step 3	`Switch(config)# [`**`no`**`]` **`ip arp inspection vlan`** *vlan-range* **`logging`** {**`acl-match`** {**`matchlog`** | **`none`**} | **`dhcp-bindings`** {**`all`** | **`none`** | **`permit`**}}	Controls the type of packets that are logged per VLAN. By default, all denied or all dropped packets are logged. The term *logged* means the entry is placed in the log buffer and a system message is generated. The keywords have these meanings: • For *vlan-range*, specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. • For **acl-match matchlog**, log packets based on the ACE logging 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 ACEs with log keyword are logged. • For **acl-match none**, do not log packets that match ACLs. • For **dhcp-bindings all**, log all packets that match DHCP bindings. • For **dhcp-bindings none**, do not log packets that match DHCP bindings. • For **dhcp-bindings permit**, log DHCP-binding permitted packets.
Step 4	`Switch(config)#` **`exit`**	Returns to privileged EXEC mode.
Step 5	`Switch#` **`show ip arp inspection log`**	Verifies your settings.
Step 6	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To return to the default log buffer settings, use the **no ip arp inspection log-buffer** global configuration command. To return to the default VLAN log settings, use the **no ip arp inspection vlan** *vlan-range* **logging** {**acl-match** | **dhcp-bindings**} global configuration command. To clear the log buffer, use the **clear ip arp inspection log** privileged EXEC command.

This example shows how to configure the number of entries for the log buffer to 1024. It also shows how to configure your Catalyst 4500 series switch so that the logs must be generated from the buffer at the rate of 100 per 10 seconds.

```
SwitchB# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchB(config)# ip arp inspection log-buffer entries 1024
SwitchB(config)# ip arp inspection log-buffer logs 100 interval 10
SwitchB(config)# end
SwitchB# show ip arp inspection log
Total Log Buffer Size : 1024
Syslog rate : 100 entries per 10 seconds.


Interface   Vlan  Sender MAC      Sender IP        Num Pkts   Reason       Time
----------  ----  --------------  ---------------  ---------  -----------  ----
Gi3/31      100   0002.0002.0003  170.1.1.2                5  DHCP Deny    02:05:45 UTC
Fri Feb 4 2005
SwitchB#
```

Limiting the Rate of Incoming ARP Packets

The switch CPU performs dynamic ARP inspection validation checks; therefore, the number of incoming ARP packets is rate-limited to prevent a denial-of-service attack.

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 or you enable error-disable recovery so that ports automatically emerge from this state after a specified timeout period.

Note Unless you explicitly configure a rate limit on an interface, changing the trust state of the interface also changes its rate limit to the default value for that trust state. After you configure the rate limit, the interface retains the rate limit even when its trust state is changed. If you enter the **no ip arp inspection limit** interface configuration command, the interface reverts to its default rate limit.

To limit the rate of incoming ARP packets, perform this task beginning in privileged EXEC mode.

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Specifies the interface to be rate-limited, and enter interface configuration mode.
Step 3	`Switch(config-if)# [no] ip arp inspection limit {rate pps [burst interval seconds] | none}`	Limits the rate of incoming ARP requests and responses on the interface. The default rate is 15 pps on untrusted interfaces and unlimited on trusted interfaces. The burst interval is 1 second. The keywords have these meanings: • For **rate** *pps*, specify an upper limit for the number of incoming packets processed per second. The range is 0 to 2048 pps. • (Optional) For **burst interval** *seconds*, 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. • For **rate none**, specify no upper limit for the rate of incoming ARP packets that can be processed.
Step 4	`Switch(config-if)# exit`	Returns to global configuration mode.
Step 5	`Switch(config)# errdisable recovery {cause arp-inspection | interval interval}`	(Optional) Enables error recovery from the dynamic ARP inspection error-disable state. By default, recovery is disabled, and the recovery interval is 300 seconds. For **interval** *interval*, specify the time in seconds to recover from the error-disable state. The range is 30 to 86400.
Step 6	`Switch(config)# exit`	Returns to privileged EXEC mode.
Step 7	`Switch# show ip arp inspection interfaces`	Verifies your settings.

	Command	Purpose
Step 8	`Switch# show errdisable recovery`	Verifies your settings.
Step 9	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return to the default rate-limit configuration, use the **no ip arp inspection limit** interface configuration command. To disable error recovery for dynamic ARP inspection, use the **no errdisable recovery cause arp-inspection** global configuration command.

This example shows how to set an upper limit for the number of incoming packets (100 pps) and to specify a burst interval (1 second):

```
SwitchB# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchB(config)# interface g3/31
SwitchB(config-if)# ip arp inspection limit rate 100 burst interval 1
SwitchB(config-if)# exit
SwitchB(config)# errdisable recovery cause arp-inspection
SwitchB(config)# exit
SwitchB# show ip arp inspection interfaces

 Interface        Trust State     Rate (pps)    Burst Interval
 ---------------  -----------     ----------    --------------
 Gi1/1            Untrusted               15                 1
 Gi1/2            Untrusted               15                 1
 Gi3/1            Untrusted               15                 1
 Gi3/2            Untrusted               15                 1
 Gi3/3            Untrusted               15                 1
 Gi3/4            Untrusted               15                 1
 Gi3/5            Untrusted               15                 1
 Gi3/6            Untrusted               15                 1
 Gi3/7            Untrusted               15                 1
 Gi3/8            Untrusted               15                 1
 Gi3/9            Untrusted               15                 1
 Gi3/10           Untrusted               15                 1
 Gi3/11           Untrusted               15                 1
 Gi3/12           Untrusted               15                 1
 Gi3/13           Untrusted               15                 1
 Gi3/14           Untrusted               15                 1
 Gi3/15           Untrusted               15                 1
 Gi3/16           Untrusted               15                 1
 Gi3/17           Untrusted               15                 1
 Gi3/18           Untrusted               15                 1
 Gi3/19           Untrusted               15                 1
 Gi3/20           Untrusted               15                 1
 Gi3/21           Untrusted               15                 1
 Gi3/22           Untrusted               15                 1
 Gi3/23           Untrusted               15                 1
 Gi3/24           Untrusted               15                 1
 Gi3/25           Untrusted               15                 1
 Gi3/26           Untrusted               15                 1
 Gi3/27           Untrusted               15                 1
 Gi3/28           Untrusted               15                 1
 Gi3/29           Untrusted               15                 1
 Gi3/30           Untrusted               15                 1
 Gi3/31           Untrusted              100                 1
 Gi3/32           Untrusted               15                 1
 Gi3/33           Untrusted               15                 1
 Gi3/34           Untrusted               15                 1
 Gi3/35           Untrusted               15                 1
 Gi3/36           Untrusted               15                 1
 Gi3/37           Untrusted               15                1
 Gi3/38           Untrusted               15                1
 Gi3/39           Untrusted               15                1
 Gi3/40           Untrusted               15                1
 Gi3/41           Untrusted               15                1
 Gi3/42           Untrusted               15                1
 Gi3/43           Untrusted               15                1
 Gi3/44           Untrusted               15                1
 Gi3/45           Untrusted               15                1
 Gi3/46           Trusted               None              N/A
 Gi3/47           Untrusted               15                1
 Gi3/48           Untrusted               15                1

SwitchB# show errdisable recovery
ErrDisable Reason    Timer Status
-----------------    --------------
udld                 Disabled
bpduguard            Disabled
security-violatio    Disabled
channel-misconfig    Disabled
vmps                 Disabled
pagp-flap            Disabled
dtp-flap             Disabled
link-flap            Disabled
l2ptguard            Disabled
psecure-violation    Disabled
gbic-invalid         Disabled
dhcp-rate-limit      Disabled
unicast-flood        Disabled
storm-control        Disabled
arp-inspection       Enabled

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

SwitchB#
1w2d: %SW_DAI-4-PACKET_RATE_EXCEEDED: 101 packets received in 739 milliseconds on Gi3/31.
1w2d: %PM-4-ERR_DISABLE: arp-inspection error detected on Gi3/31, putting Gi3/31 in
err-disable state
SwitchB# show clock
*02:21:43.556 UTC Fri Feb 4 2005
SwitchB#
SwitchB# show interface g3/31 status

Port      Name               Status       Vlan       Duplex  Speed Type
Gi3/31                       err-disabled 100          auto   auto 10/100/1000-TX
SwitchB#
SwitchB#
1w2d: %PM-4-ERR_RECOVER: Attempting to recover from arp-inspection err-disable state on
Gi3/31
SwitchB# show interface g3/31 status

Port      Name               Status       Vlan       Duplex  Speed Type
Gi3/31                       connected    100        a-full  a-100 10/100/1000-TX
SwitchB# show clock
*02:27:40.336 UTC Fri Feb 4 2005
SwitchB#
```

Performing Validation Checks

Dynamic ARP inspection intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can configure the switch to perform additional checks on the destination MAC address, the sender and target IP addresses, and the source MAC address.

To perform specific checks on incoming ARP packets, perform this task.

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# ip arp inspection validate {[src-mac] [dst-mac] [ip]}`	Performs a specific check on incoming ARP packets. By default, no additional checks are performed. The keywords have these meanings: • For **src-mac**, check 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. • For **dst-mac**, check 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. • For **ip**, check 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. You must specify at least one of the keywords. Each command overrides the configuration of the previous command; that is, if a command enables **src** and **dst mac** validations, and a second command enables IP validation only, the **src** and **dst mac** validations are disabled as a result of the second command.
Step 3	`Switch(config)# exit`	Returns to privileged EXEC mode.
Step 4	`Switch# show ip arp inspection vlan vlan-range`	Verifies your settings.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To disable checking, use the **no ip arp inspection validate** [**src-mac**] [**dst-mac**] [**ip**] global configuration command. To display statistics for forwarded, dropped, MAC validation failure, and IP validation failure packets, use the **show ip arp inspection statistics** privileged EXEC command.

This example shows how to configure source mac validation. Packets are dropped and an error message may be generated when the source address in the Ethernet header does not match the sender hardware address in the ARP body.

```
SwitchB# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchB(config)# ip arp inspection validate src-mac
SwitchB(config)# exit
SwitchB# show ip arp inspection vlan 100

Source Mac Validation      : Enabled
```

```
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
  100     Enabled          Active

 Vlan     ACL Logging      DHCP Logging
 ----     -----------      ------------
  100     Deny             Deny
SwitchB#
1w2d: %SW_DAI-4-INVALID_ARP: 9 Invalid ARPs (Req) on Gi3/31, vlan
100.([0002.0002.0002/170.1.1.2/0001.0001.0001/170.1.1.1/02:30:24 UTC Fri Feb 4 2005])
```

CHAPTER 39

Configuring Network Security with ACLs

This chapter describes how to use access control lists (ACLs) to configure network security on the Catalyst 4500 series switches.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

This chapter consists of the following major sections:

Note The following discussions applies to both Supervisor Engine 6E and non-Supervisor Engine 6E configurations unless noted otherwise.

Understanding ACLs

This section contains the following subsections:

- ACL Overview, page 39-2
- Supported Features That Use ACLs, page 39-3
- Router ACLs, page 39-3
- Port ACLs, page 39-4
- VLAN Maps, page 39-5

ACL Overview

An ACL is a collection of sequential permit and deny conditions that applies to packets. When a packet is received on an interface, the switch compares the fields in the packet against any applied ACLs to verify that the packet has the permissions required to be forwarded, based on the conditions specified in the access lists. It tests the packets against the conditions in an access list one-by-one. The first match determines whether the switch accepts or rejects the packets. Because the switch stops testing conditions after the first match, the order of conditions in the list is critical. If no conditions match, the switch drops the packet. If there are no restrictions, the switch forwards the packet; otherwise, the switch drops the packet.

Switches traditionally operate at Layer 2, switching traffic within a VLAN, whereas routers route traffic between VLANs at Layer 3. The Catalyst 4500 series switch can accelerate packet routing between VLANs by using Layer 3 switching. The Layer 3 switch bridges the packet, and then routes the packet internally without going to an external router. The packet is then bridged again and sent to its destination. During this process, the switch can control all packets, including packets bridged within a VLAN.

You configure access lists on a router or switch to filter traffic and provide basic security for your network. If you do not configure ACLs, all packets passing through the switch could be allowed on all parts of the network. You can use ACLs to control which hosts can access different parts of a network or to decide which types of traffic are forwarded or blocked at router interfaces. For example, you can allow e-mail traffic to be forwarded but not Telnet traffic. ACLs can be configured to block inbound traffic, outbound traffic, or both. However, on Layer 2 interfaces, you can apply ACLs only in the inbound direction.

An ACL contains an ordered list of access control entries (ACEs). Each ACE specifies permit or deny and a set of conditions the packet must satisfy in order to match the ACE. The meaning of permit or deny depends on the context in which the ACL is used.

The Catalyst 4500 series switch supports three types of ACLs:

- IP ACLs, which filter IP traffic, including TCP, the User Datagram Protocol (UDP), Internet Group Management Protocol (IGMP), and Internet Control Message Protocol (ICMP).
- IPv6 ACLs (applies only to Supervisor Engine 6E).

Supported Features That Use ACLs

The switch supports two applications of ACLs to filter traffic:

- Router ACLs are applied to Layer 3 interfaces. They control the access of routed traffic between VLANs. All Catalyst 4500 series switches can create router ACLs, but you must have a Cisco IOS software image on your switch to apply an ACL to a Layer 3 interface and filter packets routed between VLANs.
- Port ACLs perform access control on traffic entering a Layer 2 interface. If there are not enough hardware CAM entries, the output port ACL is not applied to the port and a warning message is given to user. (This restriction applies to all access group modes for output port ACLs.) When there are enough CAM entries, the output port ACL might be reapplied.

  If there is any output port ACL configured on a Layer 2 port, then no VACL or router ACL can be configured on the VLANs that the Layer 2 port belongs to. Also, the reverse is true: port ACLs and VLAN-based ACLs (VACLs and router ACLs) are mutually exclusive on a Layer 2 port. This restriction applies to all access group modes. On the input direction, port ACLs, VLAN-based ACLs, and router ACLs can co-exist.

  You can apply only one IP access list, one MAC access list for a Layer 2 interface.
- VLAN ACLs or VLAN maps control the access of all packets (bridged and routed). You can use VLAN maps to filter traffic between devices in the same VLAN. You do not need the enhanced image to create or apply VLAN maps. VLAN maps are configured to control access based on Layer 3 addresses for IP. MAC addresses using Ethernet ACEs control the access of unsupported protocols. After you apply a VLAN map to a VLAN, all packets (routed or bridged) entering the VLAN are checked against that map. Packets can either enter the VLAN through a switch port or through a routed port after being routed.

You can use both router ACLs and VLAN maps on the same switch.

Router ACLs

You can apply one access-list of each supported type to an interface.

Note Catalyst 4500 series switches running Cisco IOS Release 12.2(40)SG do *not* support IPv6 Port ACLs (PACLs).

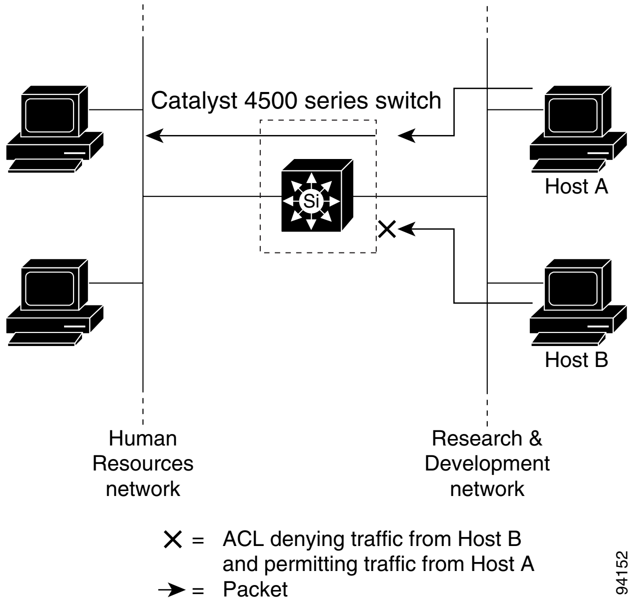
Multiple features can use one ACL for a given interface, and one feature can use multiple ACLs. When a single router ACL is used by multiple features, it is examined multiple times. The access list type determines the input to the matching operation:

- Standard IP access lists use source addresses for matching operations.
- Extended IP access lists use source and destination addresses and optional protocol type information for matching operations.

The switch examines ACLs associated with features configured on a given interface and a direction. As packets enter the switch on an interface, ACLs associated with all inbound features configured on that interface are examined. After packets are routed and before they are forwarded to the next hop, all ACLs associated with outbound features configured on the egress interface are examined.

ACLs permit or deny packet forwarding based on how the packet matches the entries in the ACL. For example, you can use access lists to allow one host to access a part of a network, but prevent another host from accessing the same part. In Figure 39-1, ACLs applied at the router input allow Host A to access the Human Resources network, but prevent Host B from accessing the same network.

Figure 39-1 Using ACLs to Control Traffic to a Network


Catalyst 4500 series switch
Si
Host A
Host B
Human Resources network
Research & Development network
× = ACL denying traffic from Host B and permitting traffic from Host A
→ = Packet
94152

Port ACLs

You can also apply ACLs to Layer 2 interfaces on a switch. Port ACLs are supported on physical interfaces and EtherChannel interfaces.

The following access lists are supported on Layer 2 interfaces:

- Standard IP access lists using source addresses
- Extended IP access lists using source and destination addresses and optional protocol type information
- MAC extended access lists using source and destination MAC addresses and optional protocol type information

As with router ACLs, the switch examines ACLs associated with features configured on a given interface and permits or denies packet forwarding based on how the packet matches the entries in the ACL. In the example in Figure 39-1, if all workstations were in the same VLAN, ACLs applied at the Layer 2 input would allow Host A to access the Human Resources network, but prevent Host B from accessing the same network.

When you apply a port ACL to a trunk port, the ACL filters traffic on all VLANs present on the trunk port. When you apply a port ACL to a port with voice VLAN, the ACL filters traffic on both data and voice VLANs.

With port ACLs, you can filter IP traffic by using IP access lists and non-IP traffic by using MAC addresses. You can filter both IP and non-IP traffic on the same Layer 2 interface by applying both an IP access list and a MAC access list to the interface.

Note You cannot apply more than one IP access list and one MAC access list to a Layer 2 interface. If an IP access list or MAC access list is already configured on a Layer 2 interface and you apply a new IP access list or MAC access list to the interface, the new ACL replaces the previously configured one.

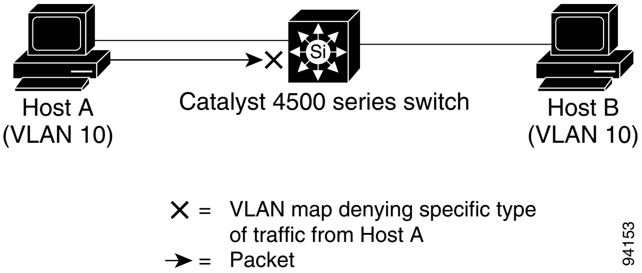
VLAN Maps

VLAN maps can control the access of all traffic in a VLAN. You can apply VLAN maps on the switch to all packets that are routed into or out of a VLAN or are bridged within a VLAN. Unlike router ACLs, VLAN maps are not defined by direction (input or output).

You can configure VLAN maps to match Layer 3 addresses for IP traffic. Access of all non-IP protocols is controlled with a MAC address and an Ethertype using MAC ACLs in VLAN maps. (IP traffic is not controlled by MAC ACLs in VLAN maps.) You can enforce VLAN maps only on packets going through the switch; you cannot enforce VLAN maps on traffic between hosts on a hub or on another switch connected to this switch.

With VLAN maps, forwarding packets is permitted or denied, based on the action specified in the map. Figure 39-2 illustrates how a VLAN map is applied to deny a specific type of traffic from Host A in VLAN 10 from being forwarded.

Figure 39-2 Using VLAN Maps to Control Traffic



Hardware and Software ACL Support

This section describes how to determine whether ACLs are processed in hardware or in software:

- Flows that match a deny statement in standard and extended ACLs are dropped in hardware if ICMP unreachable messages are disabled.
- Flows that match a permit statement in standard ACLs are processed in hardware.
- The following ACL types are not supported in software:
  - Standard Xerox Network Systems (XNS) Protocol access list
  - Extended XNS access list
  - DECnet access list

- Protocol type-code access list
- Standard Internet Packet Exchange (IPX) access list
- Extended IPX access list

Note Packets that require logging are processed in software. A copy of the packets is sent to the CPU for logging while the actual packets are forwarded in hardware so that non-logged packet processing is not impacted.

By default, the Catalyst 4500 series switch sends ICMP unreachable messages when a packet is denied by an access list; these packets are not dropped in hardware but are forwarded to the switch so that it can generate the ICMP unreachable message.

To drop access-list denied packets in hardware on the input interface, you must disable ICMP unreachable messages using the **no ip unreachables** interface configuration command. The **ip unreachables** command is enabled by default.

Note Cisco IOS Release 12.2(40)SG does not support disabling ip unreachables on interfaces routing IPv6 traffic.

Note If you set the **no ip unreachable** command on all Layer 3 interfaces, output ACL denied packets do not come to the CPU.

TCAM Programming and ACLs for Supervisor Engine II-Plus, Supervisor Engine IV, Supervisor Engine V, and Supervisor Engine V-10GE

TCAM entry and mask utilization on the Catalyst 4500 series switch is based on the following elements:

- ACL configuration
- Supervisor model
- IOS software version

For Supervisor Engine II-Plus-10GE, Supervisor Engine V-10GE, and the Catalyst 4948-10GE switch, the entry and mask utilization equals the number of ACEs in the ACL configuration divided by the number of entries in the TCAM region, regardless of IOS software version. Optimzed TCAM utilization is not required.

For Supervisor Engine II-Plus-TS, Supervisor Engines IV, Supervisor Engines V, and the Catalyst 4948 switch, up to eight entries share a single mask in the TCAM regardless of the IOS software release. So, TCAM utilization depends on the ACL configuration. It also depends on the order of configuring each ACL; TCAM utilization may differ if one ACL is configured before another and vice versa. Copying the same ACL configuration to the running-config may also cause TCAM utilization to change.

TCAM utilization on Supervisors II-Plus-TS, IV, V, and the Catalyst 4948 switch may be optimized depending on the ACL configuration and IOS software version. For instance, Cisco IOS Release 12.2(31)SGA and later releases automatically reorder order-independent ACL entries to preserve masks. Two ACEs are order-independent if a single packet can match only one of them. For example, the following two ACEs are order-independent:

```
permit ip host 10.1.1.10 any
permit ip host 10.1.1.20 any
```

Any packet that would match the first ACE would not match the second, and vice versa. In contrast, the following two ACEs are not order-independent:

```
permit ip host 10.1.1.10 any
permit ip any host 10.1.1.20
```

A packet with source IP address 10.1.1.10 and destination IP address 10.1.1.20 would be able to match both ACEs, so their order matters.

When estimating TCAM utilization for Supervisor Engines II-Plus-TS, IV, V, and the Catalyst 4948 switch prior to deployment, start with the default configuration. Because of the dynamic nature of programming ACEs that share masks, estimating TCAM utilization is unpredictable when ACLs are already programmed.

For Cisco IOS Release 12.2(31)SGA and later, you can estimate TCAM utilization for IP ACL provided the TCAM is empty. For each IP ACL, four ACEs are automatically added to the ACL: two static ACEs, an appended IP deny-all ACE, and an appended permit-all ACE. So, the minimum number of masks for an IP ACL is five. To find the number of masks utilized by the remaining ACEs, count the number of different masks, adding one for every different mask with more than eight ACEs.

For Supervisor Engines II-Plus-TS, IV, V, and the Catalyst 4948 switch running IOS software prior to release 12.2(31)SGA, ACLs are not automatically optimized before the TCAM is programmed. Grouping ACEs with similar masks prior to configuring the ACL may improve mask utilization.

Note After upgrading to Cisco IOS Release 12.2(31)SGA or later on Supervisor Engines II-Plus-TS, IV, V, and the Catalyst 4948 switch, TCAM ACL utilization may decrease because of independent ACE reordering. Conversely, downgrading to Cisco IOS Release 12.2(31)SG or earlier may cause TCAM utilization to increase.

TCAM Programming Algorithms

Note The TCAM programming algorithm is *NOT* available on Supervisor Engine 6-E.

Beginning with Cisco IOS Release 12.2(25)EWA, two TCAM programming algorithms are supported on Catalyst 4500 and 4900 series switches: packed and scattered. The *packed* mode algorithm programs the entries in the same 8-entry TCAM block provided the entries' masks match. If the current entry's mask differs from previous entries', the switch software programs the entry in a new 8-entry block. If the mask does not change, or if the mask changes every eight entries across ACLs from the beginning to the end of the configuration, the TCAM may be fully utilized in packed mode for Supervisor Engines II-Plus-TS, IV, V, and the Catalyst 4948 series switch.

In *scattered* mode, a single ACL's entries are distributed across different 8-entry blocks until the ACL is fully programmed. If successive ACLs have the same mask pattern as the first ACL, the TCAM on Supervisor Engines II-Plus-TS, IV, V, and the Catalyst 4948 series switch may be fully utilized.

Scattered mode is recommended for IP Source Guard configurations on Supervisor Engines II-Plus-TS, IV, V, and the Catalyst 4948 switch. This is because the mask pattern for per-VLAN ACLs is the same for all ports configured for IP Source Guard: permit ARP packets, permit Layer 2 traffic if port security is not configured, permit IP traffic from a particular source IP address with a 32-bit mask, deny unknown, and permit all.

Note The TCAM programming algorithm can be configured on Supervisor Engine V-10GE and the Catalyst 4948-10GE switch running Cisco IOS Release 12.2(25)EWA or its subsequent maintenance releases. On Supervisor Engine V-10GE and the Catalyst 4948-10GE switch, however, because ACL masks are not shared among ACEs, TCAM utilization will be the same regardless of the programming algorithm is configured.

Note The TCAM programming algorithm cannot be configured on Supervisor Engines II-Plus-10GE or V-10GE or the Catalyst 4948-10GE switch running Cisco IOS Release 12.2(25)SG and later.

Note The TCAM utilization should not change after you successively configure the same TCAM programming algorithm. For example, configuring access-list hardware entries packed twice should not affect TCAM utilization. However, TCAM utilization may change if one or more commands are issued between successive configurations of the same TCAM programming algorithm.

Below is a summary of scenarios that cause TCAM utilization to change:

- Addition or deletion of ACLs or ACEs in the running-config
- Copying or recopying the ACL configuration from bootflash, a TFTP server, or compact flash memory to the running-config
- Changing the TCAM programming algorithm
- Saving the running-config to NVRAM and reloading the switch
- Resizing the feature ACL or QoS regions of the TCAM with the **access-list hardware region <feature | qos> <input | output> balance** <*percent*> command on Cisco IOS Release 12.2(31)SGA and beyond.
- Upgrading from images based on Cisco IOS Release 12.2(25)EWA to images based on Cisco IOS Release 12.2(31)SGA

As discussed earlier, two types of hardware resources are consumed when you program ACLs: entries and masks. If either one of these resources is exhausted, no additional ACLs can be programmed into hardware.

If you run out of resources, you refer to the folowing sections:

- Change the Programming Algorithm, page 39-9
- Resize the TCAM Regions, page 39-10
- Selecting Mode of Capturing Control Packets, page 39-12

Change the Programming Algorithm

If the masks on a system are exhausted, but entries are available, changing the programming scheme from packed to scattered might free up masks, allowing additional ACLs to be programmed into hardware.

Note Changing the ACL programming algorithm or resizing the TCAM regions causes all ACLs to be temporarily unloaded from the hardware and then reloaded in accordance with the new TCAM parameters. ACLs are inoperative until the reloading process is complete.

The goal is to use TCAM resources more efficiently by minimizing the number of masks per ACL entry.

To...	Use the Command...
Compare TCAM utilization when employing the scattered or packed algorithms.	`Switch#` **`show platform hardware acl statistics utilization brief`**
Change the algorithm from packed to scattered.	`Switch(config)#` **`access-list hardware entries scattered`**
Change the algorithm from scattered to packed.	`Switch(config)#` **`access-list hardware entries packed`**

Note To determine whether the scattered algorithm is configured, use the **show running-config** command. If scattered is configured, the line **access-list hardware entries scattered** appears.

Note The default TCAM programming algorithm is packed.

The following output was collected from a switch running in packed mode. Observe that 89 percent of the masks are required to program only 49 percent of the ACL entries.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# access-list hardware entries packed
Switch(config)# end
Switch#
01:15:34: %SYS-5-CONFIG_I: Configured from console by console
Switch#
Switch# show platform hardware acl statistics utilization brief
                                       Entries/Total(%)  Masks/Total(%)
                                       ----------------- ---------------
            Input  Acl(PortAndVlan)  2016 / 4096 ( 49)   460 /  512 ( 89)
            Input  Acl(PortOrVlan)      6 / 4096 (  0)     4 /  512 (  0)
            Input  Qos(PortAndVlan)     0 / 4096 (  0)     0 /  512 (  0)
            Input  Qos(PortOrVlan)      0 / 4096 (  0)     0 /  512 (  0)
            Output Acl(PortAndVlan)     0 / 4096 (  0)     0 /  512 (  0)
            Output Acl(PortOrVlan)      0 / 4096 (  0)     0 /  512 (  0)
            Output Qos(PortAndVlan)     0 / 4096 (  0)     0 /  512 (  0)
            Output Qos(PortOrVlan)      0 / 4096 (  0)     0 /  512 (  0)

            L4Ops: used 2 out of 64
```

The following output was collected after the algorithm was switched to scattered. Observe that the number of masks required to program 49 percent of the entries has decreased to 49 percent.

Note When you enable DHCP snooping and IP Source Guard on all ports on a chassis, you must use the scattered keyword.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# access-list hardware entries scattered
Switch(config)# end
Switch#
01:39:37: %SYS-5-CONFIG_I: Configured from console by console
Switch#
Switch# show platform hardware acl statistics utilization brief
Entries/Total(%)  Masks/Total(%)
                                       -----------------  ---------------
            Input  Acl(PortAndVlan)  2016 / 4096 ( 49)   252 /  512 ( 49)
            Input  Acl(PortOrVlan)      6 / 4096 (  0)     5 /  512 (  0)
            Input  Qos(PortAndVlan)     0 / 4096 (  0)     0 /  512 (  0)
            Input  Qos(PortOrVlan)      0 / 4096 (  0)     0 /  512 (  0)
            Output Acl(PortAndVlan)     0 / 4096 (  0)     0 /  512 (  0)
            Output Acl(PortOrVlan)      0 / 4096 (  0)     0 /  512 (  0)
            Output Qos(PortAndVlan)     0 / 4096 (  0)     0 /  512 (  0)
            Output Qos(PortOrVlan)      0 / 4096 (  0)     0 /  512 (  0)

            L4Ops: used 2 out of 64
Switch#
```

Resize the TCAM Regions

The TCAM is divided into regions, each of which holds different kinds of entries. There are four main TCAM types: Input ACL, Output ACL, Input QoS, and Output QoS. Each is divided into a PortAndVlan region and a PortOrVlan region. By default the PortAndVlan region and PortOrVlan region are of equal size.

The following table lists the entries and mask counts for each of the supported supervisor engines. The entry/mask counts for the various supervisor engines are for each TCAM type; for example, the input feature TCAM has 16,000 entries, and the output feature TCAM has 16,000 entries.

Supervisor Engine	Entries	Masks
Supervisor Engine III	16,000	2,000
Supervisor Engine IV	16,000	2,000
Supervisor Engine V	16,000	2,000
Supervisor Engine II-Plus	8,000	1,000
Supervisor Engine II-Plus-TS	8,000	1,000
Supervisor Engine V-10GE	16,000	16,000
Supervisor Engine II-Plus-10GE	TBP	TBP

Note Because of the ratio of entries to masks on Supervisor Engines II-Plus-TS, IV, V, and the Catalyst 4948 switch is 8:1, TCAM space for masks may be exhausted before space for entries is exhausted.

Note One region in a TCAM type might be filled while the other region still has free space. When this happens, the regions can be resized to move the free entries from the region where they are not needed to the region where they are needed. Regions are resized using the **access-list hardware region {feature | qos} {input | output} balance** command. Each TCAM type has its own independent region balance.

Note Higher balance values indicate more entries to the PortAndVlan region and less entries to the PortOrVlan region. Lower balance values indicate less entries to the PortAndVlan region and more entries to the PortOrVlan region. A balance of 50 indicates equal allocation to both PortAndVlan and PortOrVlan regions.

Note You can only shift entries between the PortAndVlan region and the PortOrVlan region for a specific TCAM type (for example, from the Input ACL TCAM PortOrVlan region to the Input ACL TCAM PortAndVlan region). You cannot shift entries between TCAM types.

To determine whether region resizing would be beneficial, use the **show platform hardware acl statistics utilization brief** command:

```
Switch# show platform hardware acl statistics utilization brief

Input  Acl(PortAndVlan)     2346 / 8112 ( 29)     1014 / 1014 (100)
Input  Acl(PortOrVlan)         0 / 8112 (  0)        0 / 1014 (  0)
Input  Qos(PortOrVlan)         0 / 8128 (  0)        0 / 1016 (  0)
Input  Qos(PortOrVlan)         0 / 8128 (  0)        0 / 1016 (  0)
Output Acl(PortOrVlan)         0 / 8112 (  0)        0 / 1014 (  0)
Output Acl(PortOrVlan)         0 / 8112 (  0)        0 / 1014 (  0)
Output Qos(PortOrVlan)         0 / 8128 (  0)        0 / 1016 (  0)
Output Qos(PortOrVlan)         0 / 8128 (  0)        0 / 1016 (  0)

L4Ops: used 2 out of 64
```

The above output indicates that the Input ACL PortAndVlan region is out of masks, but there is free space in the Input ACL PortOrVlan region that could be repurposed. The following example shows how to change the region balance of the Input ACL TCAM so that 75 per cent of the entries are allocated to the PortAndVlan region and only 25 per cent to the PortOrVlan region.

```
Switch# configure terminal
Switch(config)# access-list hardware region feature input balance 75
```

After adjusting the region balance, the PortAndVlan region has more resources allocated to it, and the PortOrVlan region has fewer resources.

```
Switch# show platform hardware acl statistics utilization brief

Input  Acl(PortAndVlan)     2346 / 12160 ( 19)     1014 / 1520 ( 67)
Input  Acl(PortOrVlan)         0 / 4064 (  0)        0 /  508 (  0)
Input  Qos(PortOrVlan)         0 / 8128 (  0)        0 / 1016 (  0)
Input  Qos(PortOrVlan)         0 / 8128 (  0)        0 / 1016 (  0)
```

```
Output Acl(PortOrVlan)         0 / 8112 (  0)        0 / 1014 (  0)
Output Acl(PortOrVlan)         0 / 8112 (  0)        0 / 1014 (  0)
Output Qos(PortOrVlan)         0 / 8128 (  0)        0 / 1016 (  0)
Output Qos(PortOrVlan)         0 / 8128 (  0)        0 / 1016 (  0)

L4Ops: used 2 out of 64
Switch#
```

Note The **no** form of the **access-list hardware region** {**feature** | **qos**} {**input** | **output**} **balance** command or a balance of 50 forces the configuration to the default values. A similar configuration can also be performed for QoS.

Troubleshooting High CPU Due to ACLs

Packets that match entries in fully programmed ACLs are processed in hardware. However, large ACL and IPSG configurations may exhaust TCAM masks on Supervisor Engines II-Plus-TS, IV, V, and the Catalyst 4948 switch before the ACLs are fully programmed.

Packets that match entries in partially programmed ACLs are processed in software using the CPU. This may cause high CPU utilization and packets to be dropped. To determine whether packets are being dropped due to high CPU utilization, reference the following:

http://www.cisco.com/en/US/products/hw/switches/ps663/products_tech_note09186a00804cef15.shtml

If the ACL and/or IPSG configuration is partially programmed in hardware, upgrading to Cisco IOS Release 12.2(31)SGA or later and resizing the TCAM regions may enable the ACLs to be fully programmed.

Note Removal of obsolete TCAM entries may take several CPU process review cycles to complete. This may cause some packets to be switched in software if the TCAM entry or mask utilization is at or near 100%.

Selecting Mode of Capturing Control Packets

Note Supervisor Engine 6-E does *not* support this feature.

In some deployments, you might want to bridge control packets in hardware rather than globally capture and forwarding them in software (at the expense of the CPU). The per-VLAN capture mode feature allowing a Catalyst 4500 series switch to capture control packets only on selected VLANs and bridge traffic in hardware on all other VLANs.

When you employ per-VLAN capture mode on your switch, it partially disables the global TCAM capture entries internally and attaches feature-specific capture ACLs on those VLANs that are enabled for snooping or routing features. (All IP capture entries, CGMP, and other non-IP entries are still captured through global TCAM.) Because this feature controls specific control packets, they are captured only on the VLANs on which the internal ACLs are installed. On all other VLANs, the control traffic is bridged in hardware rather than forwarded to CPU.

The per-VLAN capture mode allows you to apply user-defined ACLs and QoS policers (in hardware) on control packets. Furthermore, you can subject the aggregate control traffic ingressing the CPU to Control Plane Policing.

When you use per-VLAN capture mode, the following four protocol groups are selectable per VLAN. Observe the breakdown of protocols intercepted by each group.

- IGMP Snooping - Cgmp, Ospf, Igmp, Pim, 224.0.0.1, 224.0.0.2, 224.0.0.*
- DHCP Snooping - Client to Server, Server to Client, Serv erto Server
- Unicast Routing - Ospf, Rip v2, 224.0.0.1, 224.0.0.2, 224.0.0.*
- Multicast Routing - Ospf, Rip v2, Igmp, Pim, 224.0.0.1, 224.0.0.2, 224.0.0.*

Because some of the groups have multiple overlapping ACEs (for example, 224.0.0.* is present in all the groups except for DHCP Snooping), turning on a certain group will also trigger the interception of some protocols from other groups.

Following are the programming triggers for the four protocol groups per VLAN:

- IGMP Snooping should be enabled globally on a given VLAN.
- DHCP Snooping should be enabled globally on a given VLAN.
- Unicast Routing should be enabled and SVI (or a Layer 3 physical) interface should be up and configured with an IP protocol address. This is because interfaces immediately become part of the routing process once the SVI interface comes up and the protocol family address is configured.
- Multicast Routing should be enabled and one of the multicast routing protocols should be configured on the interface (IGMP, PIMv1, PIMv2, MBGP, MOSPF, DVMRP, and IGMP snooping).

Caveats and Restriction

Note Before configuring per-VLAN capture mode, you should examine your configuration to ensure that only the necessary features are enabled on the desired VLANs.

The following caveats and restrictions apply to per-VLAN capture mode:

- Enabling per-VLAN capture mode consumes additional entries in the ACL/feature TCAM.

  The number of available TCAM entries depends on the type of supervisor engine. The entry/mask count further limits the utilization of the ACL/feature TCAM.

- Certain configurations can exhaust TCAM resource earlier in per-VLAN capture mode than in global capture mode (such as, IP Source Guard is enabled on several interfaces, or on a user configured PACL).

  You can re-size TCAM regions to make more entries available to the PortAndVlan or PortOrVlan region based on the configuration. This allows more entries to be programmed in hardware before reaching the limit. When TCAM resources are exhausted, the packets are forwarded in software.

- In per-VLAN capture mode, you can configure ACLs to permit or deny control traffic on a VLAN or port.

  Because security ACLs are terminated by an *implicit deny*, you must ensure that the ACLs are configured to permit the control packets necessary for the feature (protocol) to operate. However, this rule does not differ from the default behavior.

Configuration

To select the mode of capturing control packets, perform this task:

	Command	Purpose
Step 1	`Switch# conf terminal`	Enter configuration mode.
Step 2	`Switch(config)# [no] access-list hardware capture mode [vlan | global]`	Select mode of capturing control packets. The **no** form of the **access-list hardware capture mode** command restores the capture mode to the default, global.
Step 3	`Switch(config)# end`	Return to enable mode.

This example shows how to configure a Catalyst 4500 series switch to capture control packets only on VLANs where features are enabled:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# access-list hardware capture mode vlan
Switch(config)# end
Switch#
```

This example shows how to configure a Catalyst 4500 series switch to capture control packets globally across all VLANs (using *static ACL*, the default mode):

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# access-list hardware capture mode global
Switch(config)# end
Switch#
```

When the capture mode changes from global to path-managed, the static CAM entries are invalidated. This creates a window during which control packets may pass through a Catalyst 4500 series switch without being intercepted to the CPU. This temporary situation is restored as soon as the new per-VLAN capture entries are programmed in the hardware.

Once you configure per-VLAN capture mode, you should examine the show commands for individual features to verify the appropriate behavior. In per-VLAN capture mode, the invalidated static CAM entries will appear as inactive in the output of the **show platform hardware acl entries static all** command. For example, the hit count for inactive entries will remain frozen because those entries are invalidated and applied per VLAN where the feature is enabled:

CamIndex Entry Type	Active	Hit Count	CamRegion
50 PermitSharedStp	Y	3344	ControlPktsTwo
51 PermitLoopbackTest	Y	0	ControlPktsTwo
52 PermitProtTunnel	Y	0	ControlPktsTwo
53 CaptureCgmp	N	440	ControlPktsTwo
54 CaptureOspf	N	4321	ControlPktsTwo
55 CaptureIgmp	N	0	ControlPktsTwo

TCAM Programming and ACLs for Supervisor Engine 6-E

You apply three types of hardware resources when you program ACLs and ACL-based features on the Supervisor Engine 6-E: mapping table entries (MTEs), profiles, and TCAM value/mask entries. If any of these resources are exhausted, packets are sent to the CPU for software-based processing.

Note Unlike the Supervisor Engine II-Plus through V-10GE, the Supervisor Engine 6-E automatically manages the available resources. Because masks are not shared on the Supervisor Engine 6-E, there is only one programming algorithm. Because there are no regions, there is no need for region resizing. Because the per-VLAN packet capture mode is implemented differently, it cannot be disabled.

If you exhaust resources on the Supervisor Engine 6-E, you should try reducing the complexity of your configuration.

Layer 4 Operators in ACLs

The following sections describe guidelines and restrictions for configuring ACLs that include Layer 4 port operations:

- Restrictions for Layer 4 Operations, page 39-15
- Configuration Guidelines for Layer 4 Operations, page 39-16
- How ACL Processing Impacts CPU, page 39-17

Restrictions for Layer 4 Operations

You can specify these operator types, each of which uses one Layer 4 operation in the hardware:

- gt (greater than)
- lt (less than)
- neq (not equal)
- range (inclusive range)

For Supervisor Engine 2-Plus to V-10GE, you should not specify more than six different operations on the same ACL. If you exceed this number, each new operation might cause the affected ACE (access control entry) to be translated into multiple ACEs in hardware. If you exceed this number, the affected ACE might be processed in software.

For Supervisor Engine 6-E, the limits on the number of Layer 4 operations differ for each type of ACL, and can also vary based on other factors: whether an ACL is applied to incoming or outgoing traffic, whether the ACL is a security ACL or is used as a match condition for a QoS policy, and whether IPv6 ACLs are being programmed using the compressed flowlabel format.

Note The IPv6 compressed flowlabel format uses the Layer 2 Address Table to compress a portion of the IPv6 source address of each ACE in the ACL. The extra space freed in the flowlabel can then be used to support more Layer 4 operations. For this compression to be used, the IPv6 ACL cannot contain any ACEs which mask in only a portion of the bottom 48 bits of the source IPv6 address.

Generally, you will receive at most the following number of Layer 4 operations on the same ACL:

```
Direction  Protocol           Type      Operations
------------------------------------------------
Input      IPv4               Security  16
Input      IPv6 Compressed    Security  16
Input      IPv6 Uncompressed  Security   7
Input      IPv4               QoS        5
Input      IPv6 Compressed    QoS       12
Input      IPv6 Uncompressed  QoS        8
Output     IPv4               Security  17
Output     IPv6 Compressed    Security  17
Output     IPv6 Uncompressed  Security   8
Output     IPv4               QoS        5
Output     IPv6 Compressed    QoS       12
Output     IPv6 Uncompressed  QoS        8
```

Note Cases where up to 16 Operations are supported; the 17th will trigger an expansion.

If you exceed the number of available Layer 4 operations, each new operation might cause the affected ACE to be translated into multiple ACEs in the hardware. If this translation fails, packets are sent to the CPU for software processing.

Configuration Guidelines for Layer 4 Operations

Keep the following guidelines in mind when using Layer 4 operators:

- Layer 4 operations are considered different if the operator or operand differ. For example, the following ACL contains three different Layer 4 operations because gt 10 and gt 11 are considered two different Layer 4 operations:

  ```
  ... gt 10 permit
  ... lt 9 deny
  ... gt 11 deny
  ```

  Note The eq operator can be used an unlimited number of times because eq does not use a Layer 4 operation in hardware.

- Layer 4 operations are considered different if the same operator/operand couple applies once to a source port and once to a destination port, as in the following example:

  ```
  ... Src gt 10....
  ... Dst gt 10
  ```

  A more detailed example follows:

  ```
  access-list 101
  ... (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
  ```

```
access-list 102
... (dst port) gt 20 deny
... (src port) lt 9 deny
... (src port) range 11 13 deny
... (dst port) neq 6 permit
```

Access lists 101 and 102 use the following Layer 4 operations:

- Access list 101 Layer 4 operations: 5
  - gt 10 permit and gt 10 deny both use the same operation because they are identical and both operate on the destination port.
- Access list 102 Layer 4 operations: 4
- Total Layer 4 operations: 8 (due to sharing between the two access lists)
  - neq6 permit is shared between the two ACLs because they are identical and both operate on the same destination port.
- A description of the Layer 4 operations usage is as follows:
  - Layer 4 operation 1 stores gt 10 permit and gt 10 deny from ACL 101
  - Layer 4 operation 2 stores lt 9 deny from ACL 101
  - Layer 4 operation 3 stores gt 11 deny from ACL 101
  - Layer 4 operation 4 stores neg 6 permit from ACL 101 and 102
  - Layer 4 operation 5 stores neg 6 deny from ACL 101
  - Layer 4 operation 6 stores gt 20 deny from ACL 102
  - Layer 4 operation 7 stores lt 9 deny from ACL 102
  - Layer 4 operation 8 stores range 11 13 deny from ACL 102

How ACL Processing Impacts CPU

ACL processing can impact the CPU in two ways:

- For some packets, when the hardware runs out of resources, the software must perform the ACL matches:
  - TCP flag combinations other than “rst ack” and “syn fin rst,” “urq,” and “psh” are processed in hardware. “*rst ack”* is equivalent to the keyword **established**. Other TCP flag combinations are supported in software.
  - For *Supervisor Engine 2-Plus to V-10GE*, you can specify up to six Layer 4 operations (lt, gt, neq, and range) in an ACL in order for all operations to be guaranteed to be processed in hardware. More than six Layer 4 operations trigger an attempt to translate the excess operations into multiple ACEs in hardware. If this attempt fails, packets are processed in software. The translation process is less likely to succeed on large ACLs with a great number of Layer 4 operations, and on switches with large numbers of ACLs configured. The precise limit depends on how many other ACLs are configured and which specific Layer 4 operations are used by the ACLs being translated. The eq operator does not require any Layer 4 operations and can be used any number of times.
  - For *Supervisor Engine 6-E*, refer to the “Restrictions for Layer 4 Operations” section on page 39-15.
  - If the total number of Layer 4 operations in an ACL is less than six, you can distribute the operations in any way you choose.

Examples:

The following access lists are processed completely in hardware:

```
access-list 104 permit tcp any any established
access-list 105 permit tcp any any rst ack
access-list 107 permit tcp any synfin rst
```

Access lists 104 and 105 are identical; established is shorthand for rst and ack.

Access list 101, below, is processed completely in software:

```
access-list 101 permit tcp any any syn
```

Because four source and two destination operations exist, access list 106, below, is processed in hardware:

```
access-list 106 permit tcp any range 100 120 any range 120 140
access-list 106 permit tcp any range 140 160 any range 180 200
access-list 106 permit tcp any range 200 220
access-list 106 deny tcp any range 220 240
```

In the following code, the Layer 4 operations for the third ACE trigger an attempt to translate dst lt 1023 into multiple ACEs in hardware, because three source and three destination operations exist. If the translation attempt fails, the third ACE is processed in software.

```
access-list 102 permit tcp any lt 80 any gt 100
access-list 102 permit tcp any range 100 120 any range 120 1024
access-list 102 permit tcp any gt 1024 any lt 1023
```

Similarly, for access list 103, below, the third ACE triggers an attempt to translate dst gt 1023 into multiple ACEs in hardware. If the attempt fails, the third ACE is processed in software. Although the operations for source and destination ports look similar, they are considered different Layer 4 operations.)

```
access-list 103 permit tcp any lt 80 any lt 80
access-list 103 permit tcp any range 100 120 any range 100 120
access-list 103 permit tcp any gt 1024 any gt 1023
```

Note Remember that source port lt 80 and destination port lt 80 are considered different operations.

- Some packets must be sent to the CPU for accounting purposes, but the action is still performed by the hardware. For example, if a packet must be logged, a copy is sent to the CPU for logging, but the forwarding (or dropping) is performed in the hardware. Although logging slows the CPU, it does not affect the forwarding rate. This sequence of events would happen under the following conditions:
  - When a log keyword is used
  - When an output ACL denies a packet
  - When an input ACL denies a packet, and on the interface where the ACL is applied, **ip unreachable** is enabled (**ip unreachable** is enabled by default on all the interfaces)

Configuring Unicast MAC Address Filtering

To block all unicast traffic to or from a MAC address in a specified VLAN, perform this task:

Command	Purpose
`Switch(config)#` **`mac-address-table static`** *`mac_address`* **`vlan`** *`vlan_ID`* **`drop`**	Blocks all traffic to or from the configured unicast MAC address in the specified VLAN. To clear MAC address-based blocking, use the **no** form of this command without the **drop** keyword.

This example shows how to block all unicast 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 Named MAC Extended ACLs

Note This section applies to Supervisor Engines II-Plus to 6-E.

You can filter non-IP traffic on a VLAN and on a physical Layer 2 port by using MAC addresses and named MAC extended ACLs. The procedure is similar to that of configuring other extended named ACLs. You can use a number to name the access list, but MAC access list numbers from 700 to 799 are not supported.

Note Named MAC extended ACLs cannot be applied to Layer 3 interfaces.

For more information about the supported non-IP protocols in the **mac access-list extended** command, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

To create a named MAC extended ACL, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`mac access-list extended`** *`name`*	Defines an extended MAC access list using a name.

	Command	Purpose
Step 3	`Switch(config-ext-macl)#` {**deny** | **permit**} {**any** | **host** *source MAC address* | *source MAC address mask*} {**any** | **host** *destination MAC address* | *destination MAC address mask*} [**protocol-family** {**appletalk** | **arp-non-ipv4** | **decnet** | **ipx** | **ipv6** | **rarp-ipv4** | **rarp-non-ipv4** | **vines** | **xns**}]	In extended MAC access-list configuration mode, specify to **permit** or **deny any** source MAC address, a source MAC address with a mask, or a specific **host** source MAC address and **any** destination MAC address, destination MAC address with a mask, or a specific destination MAC address. (Optional) • [**protocol-family** {**appletalk** | **arp-non-ipv4** | **decnet** | **ipx** | **ipv6** | **rarp-ipv4** | **rarp-non-ipv4** | **vines** | **xns**}] **Note** On Supervisor Engine 6-E, IPv6 packets do *not* generate Layer 2 ACL lookup keys, and thus dol not match on MAC ACLs in the same way that IPv4 packets would not match against MAC ACLs in Supervisor Engines II-Plus to V-10GE. Therefore, the **ipv6** keyword is applicable to MAC ACLs on Supervisor Engines II-Plus to V-10GE, but not on Supervisor Engine 6-E.
Step 4	`Switch(config-ext-macl)#` **end**	Returns to privileged EXEC mode.
Step 5	`Switch#` **show access-lists** [*number* | *name*]	Shows the access list configuration.
Step 6	`Switch(config)#` **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

You can use the **no mac access-list extended** *name* global configuration command to delete the entire ACL. You can also delete individual ACEs from named MAC extended ACLs.

This example shows how to create and display an access list named mac1, denying only EtherType DECnet Phase IV traffic, but permitting all other types of traffic:

```
Switch(config)# mac access-list extended mac1
Switch(config-ext-macl)# deny any any decnet-iv (old) protocol-family decnet (new)
Switch(config-ext-macl)# permit any any
Switch(config-ext-macl)# end
Switch # show access-lists
Extended MAC access list mac1
    deny   any any decnet-iv (old) protocol-family decnet (new)
    permit any any
```

To enable or disable hardware statistics, enter the following commands while configuring ACEs in the access list:

```
Switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# mac access-list extended mac1
Switch(config-ext-nacl)# hardware statistics
Switch(config-ext-nacl)# end
```

Configuring Named IPv6 ACLs

Note This section applies to Supervisor Engine 6-E.

Supervisor 6E also support hardware based IPv6 Acl to filter unicast, multicast, and broadcast Ipv6 traffic on Layer 3 interface. Such access list can only be configured on l3 interface which has ipv6 address configured.

To create a named IPv6 ACLs, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`ipv6 access-list`** *`name`*	Defines an IPv6 access list using a name.
Step 3	`Switch(config-ipv6-acl)# {`**`deny`** `|` **`permit}`** **`{any`** `|` *`proto`*`}` **`{host`** *`ipv6-addr`* `|` *`ipv6-prefix`*`}` **`host`** *`ipv6-addr`* `|` *`ipv6-prefix`*`}`	Specifies each IPv6 ACE **Note** This step may be repeated to define multiple ACEs in the ACL.
Step 4	`Switch(config-ipv6-acl)#` **`hardware statistics`**	(Optional) Enables hardware statistics for the IPv6 ACL.
Step 5	`Switch(config-ipv6-acl)#` **`end`**	Returns to privileged EXEC mode.
Step 6	`Switch#` **`show ipv6 access-list`**	Display the IPv6 access list configuration.

You can use the **no ipv6 access-list name** global configuration command to delete the IPv6 ACL. You can also delete individual ACEs from IPv6 access-list.

The following example shows how to create and display an IPv6 access list named *v6test*, denying only one ipv6 traffic with one particular source/destination address, but permitting all other types of ipv6 traffic:

```
Switch(config)# ipv6 access-list v6test
Switch(config-ipv6-acl)# deny ipv6 host 2020::10 host 2040::10
Switch(config-ipv6-acl)# permit any any
Switch(config-ipv6-acl)# end
Switch# show ipv6 access-list
IPv6 access list v6test
    deny ipv6 host 2020::10 host 2040::10 sequence 10
    permit ipv6 any any sequence 20
```

To enable hardware statistics, enter the following commands while configuring the access list ACE:

```
Switch(config)# ipv6 access-list v6test
Switch(config-ipv6-acl)# hardware statistics
Switch(config-ipv6-acl)# end
```

Note *Hardware statistics* is disabled by default.

Applying IPv6 ACLs to a Layer 3 Interface

To apply an IPv6 ACL to a Layer 3 interface, perform the following task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.

	Command	Purpose
Step 2	`Switch(config)# interface interface-type slot/interface`	Specifies the interface to be configured. **Note** *interface-type* must be a Layer 3 interface.
Step 3	`Switch(config-if)# ipv6 traffic-filter ipv6-acl {in|out}`	Apply the ipv6 ACL to a Layer 3 interface.

Note IPv6 ACLs are only supported in hardware on Supervisor VI-E.

Note IPv6 ACLs are only supported on Layer 3 interfaces.

The following example applies the extended-named IPv6 ACL *simple-ipv6-acl* to SVI 300 routed ingress traffic:

```
Switch# configure terminal
Switch(config)# interface vlan 300
Switch(config-if)# ipv6 traffic-filter simple-ipv6-acl in
```

Configuring VLAN Maps

This section contains the following subsections:

- VLAN Map Configuration Guidelines, page 39-23
- Creating and Deleting VLAN Maps, page 39-23
- Applying a VLAN Map to a VLAN, page 39-26
- Using VLAN Maps in Your Network, page 39-26

This section describes how to configure VLAN maps, which is the only way to control filtering within a VLAN. VLAN maps have no direction. To filter traffic in a specific direction by using a VLAN map, you need to include an ACL with specific source or destination addresses. If there is a match clause for that type of packet (IP or MAC) in the VLAN map, the default action is to drop the packet if the packet does not match any of the entries within the map. If there is no match clause for that type of packet, the default is to forward the packet.

To create a VLAN map and apply it to one or more VLANs, perform this task:

Step 1 Create the standard or extended IP ACLs or named MAC extended ACLs that you want to apply to the VLAN.

Step 2 Enter the **vlan access-map** global configuration command to create a VLAN ACL map entry.

Step 3 In access map configuration mode, you have the optional to enter an **action** (**forward** [the default] or **drop**) and enter the **match** command to specify an IP packet or a non-IP packet and to match the packet against one or more ACLs (standard or extended). If a match clause is not specified, the action is applied to all packets. The match clause can be used to match against multiple ACLs. If a packet matches any of the specified ACLs, the action is applied.

Note If the VLAN map has a match clause for the type of packet (IP or MAC) and the packet does not match the type, the default is to drop the packet. If there is no match clause in the VLAN map for that type of packet, and no action specified, the packet is forwarded.

Step 4 Use the **vlan filter** global configuration command to apply a VLAN map to one or more VLANs.

Note You cannot apply a VLAN map to a VLAN on a switch that has ACLs applied to Layer 2 interfaces (port ACLs).

VLAN Map Configuration Guidelines

Keep the following guidelines in mind when configuring VLAN maps:

- VLAN maps do not filter IPv4 ARP packets.
- If there is no router ACL configured to deny traffic on a routed VLAN interface (input or output), and no VLAN map configured, all traffic is permitted.
- Each VLAN map consists of a series of entries. The order of entries in a VLAN map is important. A packet that comes into the switch is tested against the first entry in the VLAN map. If it matches, the action specified for that part of the VLAN map is taken. If there is no match, the packet is tested against the next entry in the map.
- If the VLAN map has at least one match clause for the type of packet (IP or MAC) and the packet does not match any of these match clauses, the default is to drop the packet. If there is no match clause for that type of packet in the VLAN map, the default is to forward the packet.
- The system might take longer to boot if you have configured a very large number of ACLs.

Creating and Deleting VLAN Maps

Each VLAN map consists of an ordered series of entries. To create, add to, or delete a VLAN map entry, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# vlan access-map name [number]`	Creates a VLAN map, and give it a name and (optionally) a number. The number is the sequence number of the entry within the map. When you create VLAN maps with the same name, numbers are assigned sequentially in increments of 10. When modifying or deleting maps, you can enter the number of the map entry that you want to modify or delete. This command enables access-map configuration mode.
Step 3	`Switch(config-access-map)# action {drop | forward}`	(Optional) Sets the action for the map entry. The default is to forward.

	Command	Purpose
Step 4	`Switch(config-access-map)# match {ip | mac} address {name | number} [name | number]`	Matches the packet (using either the IP or MAC address) against one or more standard or extended access lists. Note that packets are matched only against access lists of the correct protocol type. IP packets are compared with standard or extended IP access lists. Non-IP packets are only compared with named MAC extended access lists. If a match clause is not specified, the action is taken on all packets.
Step 5	`Switch(config-access-map)# end`	Returns to global configuration mode.
Step 6	`Switch(config)# show running-config`	Displays the access list configuration.
Step 7	`Switch(config)# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

You can use the **no vlan access-map** *name* global configuration command to delete a map. You can use the **no vlan access-map** *name number* global configuration command to delete a single sequence entry from within the map. You can use the **no action** access-map configuration command to enforce the default action, which is to forward.

VLAN maps do not use the specific **permit** or **deny** keywords. To deny a packet by using VLAN maps, create an ACL that would match the packet, and then set the action to drop. A permit in the ACL is the same as a match. A deny in the ACL means no match.

Examples of ACLs and VLAN Maps

These examples show how to create ACLs and VLAN maps that for specific purposes.

Example 1

This example shows how to create an ACL and a VLAN map to deny a packet. In the first map, any packets that match the ip1 ACL (TCP packets) would be dropped. You first create the ip1 ACL to permit any TCP packet and no other packets. Because there is a match clause for IP packets in the VLAN map, the default action is to drop any IP packet that does not match any of the match clauses.

```
Switch(config)# ip access-list extended ip1
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit

Switch(config)# vlan access-map map_1 10
Switch(config-access-map)# match ip address ip1
Switch(config-access-map)# action drop
```

This example shows how to create a VLAN map to permit a packet. ACL ip2 permits UDP packets; and any packets that match the ip2 ACL are forwarded.

```
Switch(config)# ip access-list extended ip2
Switch(config-ext-nacl)# permit udp any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map map_1 20
Switch(config-access-map)# match ip address ip2
Switch(config-access-map)# action forward
```

In this map, any IP packets that did not match any of the previous ACLs (that is, packets that are not TCP packets or UDP packets) would get dropped.

Example 2

In this example, the VLAN map is configured to drop IP packets and to forward MAC packets by default. By applying standard ACL 101 and the extended named access lists **igmp-match** and **tcp-match**, the VLAN map is configured to do the following:

- Forward all UDP packets
- Drop all IGMP packets
- Forward all TCP packets
- Drop all other IP packets
- Forward all non-IP packets

```
Switch(config)# access-list 101 permit udp any any
Switch(config)# ip access-list extended igmp-match
Switch(config-ext-nacl)# permit igmp any any
Switch(config)# ip access-list extended tcp-match
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map drop-ip-default 10
Switch(config-access-map)# match ip address 101
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-ip-default 20
Switch(config-access-map)# match ip address igmp-match
Switch(config-access-map)# action drop
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-ip-default 30
Switch(config-access-map)# match ip address tcp-match
Switch(config-access-map)# action forward
```

Example 3

In this example, the VLAN map is configured to drop MAC packets and forward IP packets by default. By applying MAC extended access lists, **good-hosts** and **good-protocols**, the VLAN map is configured to do the following:

- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211
- Forward MAC packets of DECnet or VINES (Virtual Integrated Network Service) protocol-family
- Drop all other non-IP packets
- Forward all IP packets

```
Switch(config)# mac access-list extended good-hosts
Switch(config-ext-macl)# permit host 000.0c00.0111 any
Switch(config-ext-macl)# permit host 000.0c00.0211 any
Switch(config-ext-nacl)# exit
Switch(config)# mac access-list extended good-protocols
Switch(config-ext-macl)# permit any any protocol-family decnet
Switch(config-ext-macl)# permit any any protocol-family vines
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map drop-mac-default 10
Switch(config-access-map)# match mac address good-hosts
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-mac-default 20
Switch(config-access-map)# match mac address good-protocols
Switch(config-access-map)# action forward
```

Example 4

In this example, the VLAN map is configured to drop all packets (IP and non-IP). By applying access lists **tcp-match** and **good-hosts,** the VLAN map is configured to do the following:

- Forward all TCP packets
- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211
- Drop all other IP packets
- Drop all other MAC packets

```
Switch(config)# vlan access-map drop-all-default 10
Switch(config-access-map)# match ip address tcp-match
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-all-default 20
Switch(config-access-map)# match mac address good-hosts
Switch(config-access-map)# action forward
```

Applying a VLAN Map to a VLAN

To apply a VLAN map to one or more VLANs, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`vlan filter`** *`mapname`* **`vlan-list`** *`list`*	Applies the VLAN map to one or more VLAN IDs. The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around comma, and dash, are optional.
Step 3	`Switch(config)#` **`show running-config`**	Displays the access list configuration.
Step 4	**`c`**`Switch(config)#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Note You cannot apply a VLAN map to a VLAN on a switch that has ACLs applied to Layer 2 interfaces (port ACLs).

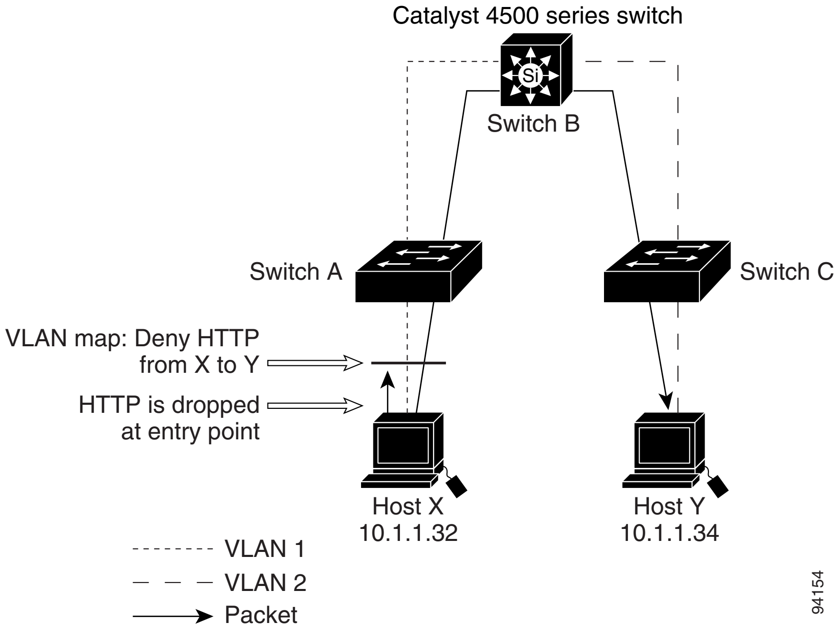
This example shows how to apply VLAN map 1 to VLANs 20 through 22:

```
Switch(config)# vlan filter map 1 vlan-list 20-22
```

Using VLAN Maps in Your Network

Figure 39-3 shows a typical wiring closet configuration. Host X and Host Y are in different VLANs, connected to wiring closet switches A and C. Traffic moving from Host X to Host Y is routed by Switch B. Access to traffic moving from Host X to Host Y can be controlled at the entry point of Switch A. In the following configuration, the switch can support a VLAN map and a QoS classification ACL.

Figure 39-3 Wiring Closet Configuration


Catalyst 4500 series switch
Si
Switch B
Switch A
Switch C
VLAN map: Deny HTTP from X to Y
HTTP is dropped at entry point
Host X
10.1.1.32
Host Y
10.1.1.34
VLAN 1
VLAN 2
Packet
94154

For example, if you do not want HTTP traffic to be switched from Host X to Host Y, you could apply a VLAN map on Switch A to drop all HTTP traffic moving from Host X (IP address 10.1.1.32) to Host Y (IP address 10.1.1.34) at Switch A and not bridge the traffic to Switch B. To configure this scenario, you would do the following:

First, define an IP access list http to permit (match) any TCP traffic on the HTTP port, as follows:

```
Switch(config)# ip access-list extended http
Switch(config-ext-nacl)# permit tcp host 10.1.1.32 host 10.1.1.34 eq www
Switch(config-ext-nacl)# exit
```

Next, create a VLAN access map named map2 so that traffic that matches the http access list is dropped and all other IP traffic is forwarded, as follows:

```
Switch(config)# vlan access-map map2 10
Switch(config-access-map)# match ip address http
Switch(config-access-map)# action drop
Switch(config-access-map)# exit

Switch(config)# ip access-list extended match_all
Switch(config-ext-nacl)# permit ip any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map map2 20
Switch(config-access-map)# match ip address match_all
Switch(config-access-map)# action forward
```

Then, apply the VLAN access map named map2 to VLAN 1, as follows:

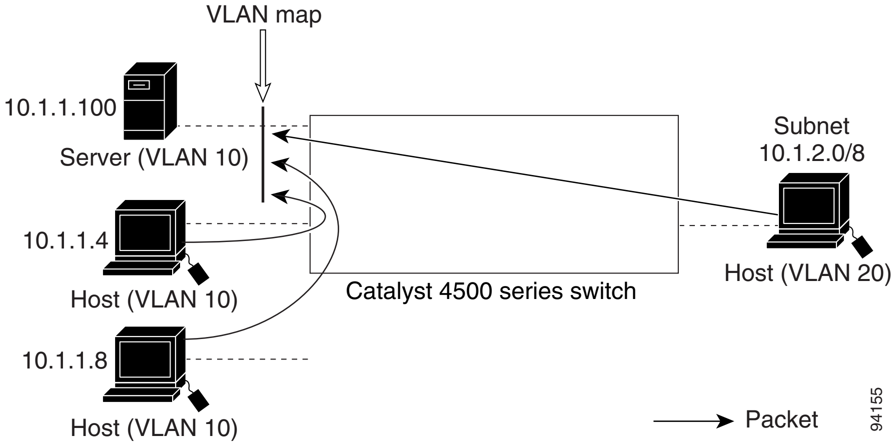
```
Switch(config)# vlan filter map2 vlan 1
```

Denying Access to a Server on Another VLAN

Figure 39-4 shows how to restrict access to a server on another VLAN. In this example, server 10.1.1.100 in VLAN 10 has the following access restrictions:

- Hosts in subnet 10.1.2.0/8 in VLAN 20 should not have access.
- Hosts 10.1.1.4 and 10.1.1.8 in VLAN 10 should not have access.

Figure 39-4 Deny Access to a Server on Another VLAN



This procedure configures ACLs with VLAN maps to deny access to a server on another VLAN. The VLAN map SERVER 1_ACL denies access to hosts in subnet 10.1.2.0/8, host 10.1.1.4, and host 10.1.1.8. Then it permits all other IP traffic. In Step 3, VLAN map SERVER1 is applied to VLAN 10.

To configure this scenario, you could take the following steps:

Step 1 Define the IP ACL to match and permit the correct packets.

```
Switch(config)# ip access-list extended SERVER1_ACL
Switch(config-ext-nacl))# permit ip 10.1.2.0 0.0.0.255 host 10.1.1.100
Switch(config-ext-nacl))# permit ip host 10.1.1.4 host 10.1.1.100
Switch(config-ext-nacl))# permit ip host 10.1.1.8 host 10.1.1.100
Switch(config-ext-nacl))# exit
```

Step 2 Define a VLAN map using the ACL to drop IP packets that match SERVER1_ACL and forward IP packets that do not match the ACL.

```
Switch(config)# vlan access-map SERVER1_MAP
Switch(config-access-map)# match ip address SERVER1_ACL
Switch(config-access-map)# action drop
Switch(config)# vlan access-map SERVER1_MAP 20
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
```

Step 3 Apply the VLAN map to VLAN 10.

```
Switch(config)# vlan filter SERVER1_MAP vlan-list 10.
```

Displaying VLAN Access Map Information

To display information about VLAN access maps or VLAN filters, perform one of these tasks.

Command	Purpose
`Switch#` **`show vlan access-map`** `[`*`mapname`*`]`	Show information about all VLAN access-maps or the specified access map.
`Switch#` **`show vlan filter`** `[`**`access-map`** *`name`* `|` **`vlan`** *`vlan-id`*`]`	Show information about all VLAN filters or about a specified VLAN or VLAN access map.

This is a sample output of the **show vlan access-map** command:

```
Switch# show vlan access-map
Vlan access-map "map_1" 10
  Match clauses:
     ip address: ip1
  Action:
     drop
Vlan access-map "map_1" 20
  Match clauses:
     mac address: mac1
  Action:
     forward
Vlan access-map "map_1" 30
  Match clauses:
  Action:
     drop
```

Note Sequence 30 does not have a match clause. All packets (IP as well as non-IP) are matched against it and dropped.

This is a sample output of the **show vlan filter** command:

```
Switch# show vlan filter
VLAN Map map_1 is filtering VLANs:
  20-22
```

Using VLAN Maps with Router ACLs

If the VLAN map has a match clause for a packet type (IP or MAC) and the packet does not match the type, the default is to drop the packet. If there is no match clause in the VLAN map, and no action is specified, the packet is forwarded if it does not match any VLAN map entry.

Note You cannot combine VLAN maps or input router ACLs with port ACLs on a switch.

Guidelines for Using Router ACLs and VLAN Maps

Use these guidelines when you need to use a router ACL and a VLAN map on the same VLAN.

Because the switch hardware performs one lookup for each direction (input and output), you must merge a router ACL and a VLAN map when they are configured on the same VLAN. Merging the router ACL with the VLAN map can significantly increase the number of ACEs.

When possible, try to write the ACL so that all entries have a single action except for the final, default action. You should write the ACL using one of these two forms:

```
permit...
permit...
permit...
deny ip any any
```

or

```
deny...
deny...
deny...
permit ip any any
```

To define multiple permit or deny actions in an ACL, group each action type together to reduce the number of entries.

If you need to specify the full-flow mode and the ACL contains both IP ACEs and TCP/UDP/ICMP ACEs with Layer 4 information, put the Layer 4 ACEs at the end of the list. Doing this gives priority to the filtering of traffic based on IP addresses.

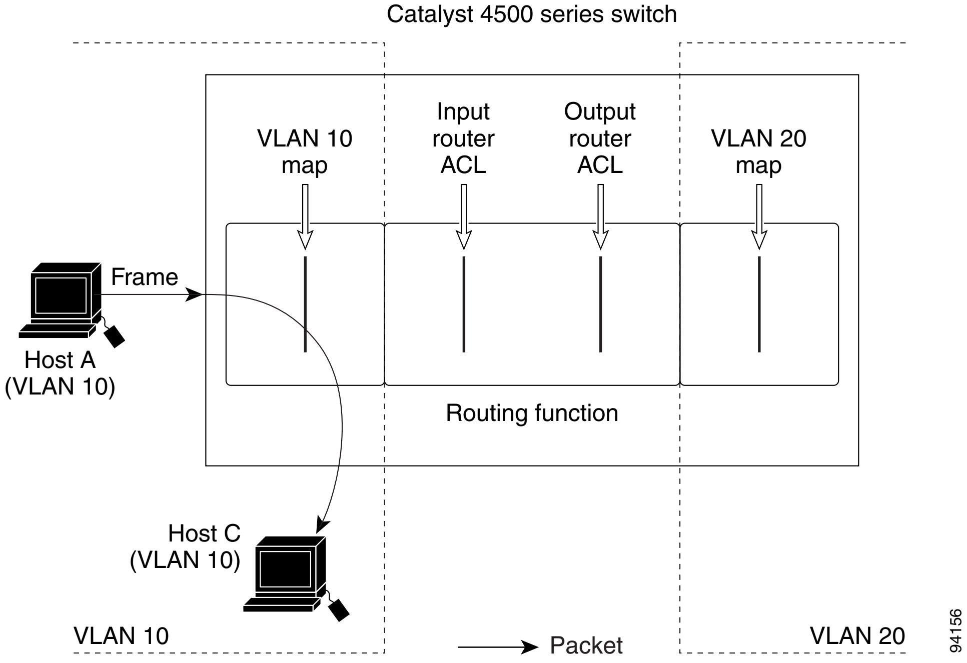
Examples of Router ACLs and VLAN Maps Applied to VLANs

These examples show how router ACLs and VLAN maps are applied on a VLAN to control the access of switched, bridged, routed, and multicast packets. Although the following illustrations show packets being forwarded to their destination, each time a packet crosses a line indicating a VLAN map or an ACL, the packet could be dropped rather than forwarded.

ACLs and Switched Packets

Figure 39-5 shows how an ACL processes packets that are switched within a VLAN. Packets switched within the VLAN are not processed by router ACLs.

Figure 39-5 Applying ACLs on Switched Packets

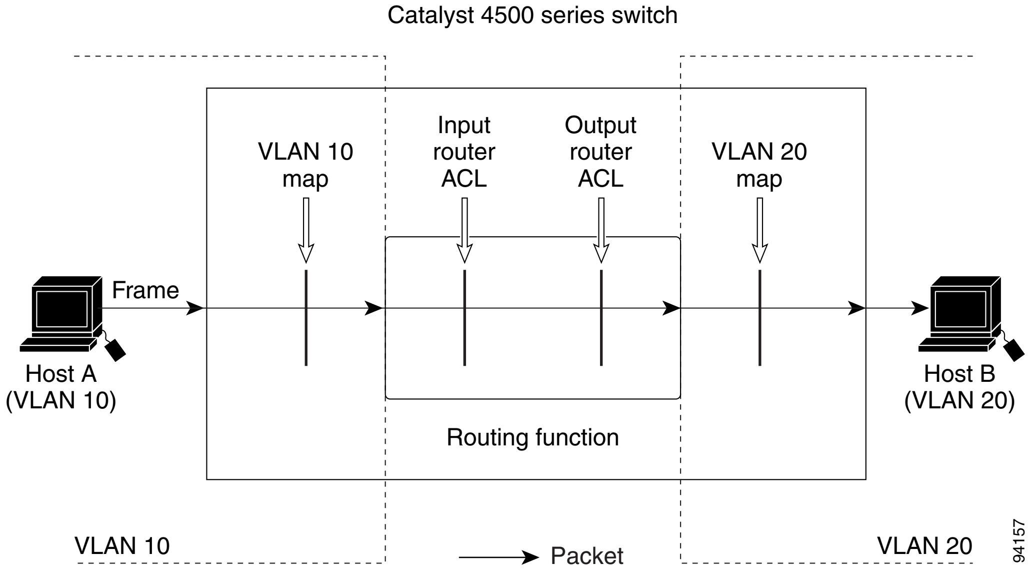


ACLs and Routed Packets

Figure 39-6 shows how ACLs are applied on routed packets. For routed packets, the ACLs are applied in this order:

1. VLAN map for input VLAN
2. Input router ACL
3. Output router ACL
4. VLAN map for output VLAN

Figure 39-6 Applying ACLs on Routed Packets



Configuring PACLs

This section describes how to configure PACLs, which are used to control filtering on Layer 2 interfaces. PACLs can filter traffic to or from Layer 2 interfaces based on Layer 3 information, Layer 4 head information or non-IP Layer 2 information.

This section contains the following topics:

- Creating a PACL, page 39-32
- PACL Configuration Guidelines, page 39-33
- Configuring IP and MAC ACLs on a Layer 2 Interface, page 39-33
- Using PACL with Access-Group Mode, page 39-34
- Configuring Access-group Mode on Layer 2 Interface, page 39-34
- Applying ACLs to a Layer 2 Interface, page 39-35
- Displaying an ACL Configuration on a Layer 2 Interface, page 39-35

Creating a PACL

To create a PACL and apply it to one or more interfaces, perform this task:

Step 1 Create the standard or extended IP ACLs or named MAC extended ACLs that you want to apply to the interface.

Step 2 Use the **ip access-group** or **mac access-group interface** command to apply a IP ACL or MAC ACL to one or more Layer 2 interfaces.

PACL Configuration Guidelines

Consider the following guidelines when configuring PACLs:

- There can be at most one IP access list and MAC access list applied to the same Layer 2 interface per direction.
- The IP access list filters only IP packets, whereas the MAC access list filters only non-IP packets.
- The number of ACLs and ACEs that can be configured as part of a PACL are bounded by the hardware resources on the switch. Those hardware resources are shared by various ACL features (for example, RACL, VACL) that are configured on the system. If there are insufficient hardware resources to program PACL in hardware, the actions for input and output PACLs differ:
  - For input PACLs, some packets are sent to CPU for software forwarding.
  - For output PACLs, the PACL is disabled on the port.
- These restrictions pertain to output PACLs only:
  - If there are insufficient hardware resources to program the PACL, the output PACL is not applied to the port, and you receive a warning message.
  - If an output PACL is configured on a Layer 2 port, then neither a VACL nor a Router ACL can be configured on the VLANs to which the Layer 2 port belongs.

    If any VACL or Router ACL is configured on the VLANs to which the Layer 2 port belongs, the output PACL cannot be configured on the Layer 2 port. That is, PACLs and VLAN-based ACLs (VACL and Router ACL) are mutually exclusive on Layer 2 ports.
- The input IP ACL logging option is supported, although logging is not supported for output IP ACLs, and MAC ACLs.
- The access group mode can change the way PACLs interact with other ACLs. To maintain consistent behavior across Cisco platforms, use the default access group mode.

Configuring IP and MAC ACLs on a Layer 2 Interface

Only IP or MAC ACLs can be applied to Layer 2 physical interfaces. Standard (numbered, named) and Extended (numbered, named) IP ACLs, and Extended Named MAC ACLs are also supported.

To apply IP or MAC ACLs on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	`Switch# `**`configure t`**	Enters global configuration mode.
Step 2	`Switch(config)# `**`interface`** *`interface`*	Enters interface config mode.
Step 3	`Switch(config-if)# `**`[no]`**` {`**`ip`**` | `**`mac`**` } `**`access-group`**` {`**`name`**` | `**`number`**`| `**`in`**`| `**`out`**`}`	Applies numbered or named ACL to the Layer 2 interface. The NO prefix deletes the IP or MAC ACL from the Layer 2 interface.
Step 4	`Switch(config)# `**`show running-config`**	Displays the access list configuration.

The following example shows how to configure the Extended Named IP ACL simple-ip-acl to permit all TCP traffic and implicitly deny all other IP traffic:

```
Switch(config)# ip access-list extended simple-ip-acl
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# end
```

The following example shows how to configure the Extended Named MACL simple-mac-acl to permit source host 000.000.011 to any destination host:

```
Switch(config)# mac access-list extended simple-mac-acl
Switch(config-ext-macl)# permit host 000.000.011 any
Switch(config-ext-macl)# end
```

Using PACL with Access-Group Mode

You can use the access group mode to change the way PACLs interact with other ACLs. For example, if a Layer 2 interface belongs to VLAN100, VACL (VLAN filter) V1 is applied on VLAN100, and PACL P1 is applied on the Layer 2 interface. In this situation, you must specify how P1 and V1 impact the traffic with the Layer 2 interface on VLAN100. In a per-interface fashion, the **access-group mode** command can be used to specify one of the desired behaviors that are defined below.

The following modes are defined:

- prefer port mode—If PACL is configured on a Layer 2 interface, then PACL takes effect and overwrites the effect of other ACLs (Router ACL and VACL). If no PACL feature is configured on the Layer 2 interface, other features applicable to the interface are merged and applied on the interface. This is the default access group mode.
- prefer vlan mode—VLAN-based ACL features take effect on the port provided they have been applied on the port and no PACLs are in effect. If no VLAN-based ACL features are applicable to the Layer 2 interface, then the PACL feature already on the interface is applied.
- merge mode—Merges applicable ACL features before they are programmed into the hardware.

Note Because output PACLs are mutually exclusive with VACL and Router ACLs, the access group mode does not change the behavior of output traffic filtering.

Configuring Access-group Mode on Layer 2 Interface

To configure an access mode on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure t`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface`*	Enters interface config mode.
Step 3	`Switch(config-if)#` **`[no]`** **`access-group mode`** {**`prefer`** {**`port`** | **`vlan`**} | **`merge`**}	Applies numbered or named ACL to the Layer 2 interface. The no prefix deletes the IP or MAC ACL from the Layer 2 interface.
Step 4	`Switch(config)#` **`show running-config`**	Displays the access list configuration.

This example shows how to merge and apply features other than PACL on the interface:

```
Switch# configure t
Switch(config)# interface interface
Switch(config-if)# access-group mode prefer port
```

This example shows how to merge applicable ACL features before they are programmed into hardware:

```
Switch# configure t
Switch(config)# interface interface
Switch(config-if)# access-group mode merge
```

Applying ACLs to a Layer 2 Interface

To apply IP and MAC ACLs to a Layer 2 interface, perform one of these tasks:

Command	Purpose
`Switch(config-if)# ip access-group ip-acl {in | out}`	Applies an IP ACL to the Layer 2 interface
`Switch(config-if)# mac access-group mac-acl {in | out}`	Applies a MAC ACL to the Layer 2 interface.

Note Supervisor Engines III and Supervisor Engine IV running on a Catalyst 4500 series switch support both input and output PACLs on an interface.

This example applies the extended named IP ACL simple-ip-acl to interface FastEthernet 6/1 ingress traffic:

```
Switch# configure t
Switch(config)# interface fastEthernet 6/1
Switch(config-if)# ip access-group simple-ip-acl in
```

This example applies the extended named MAC ACL simple-mac-acl to interface FastEthernet 6/1 egress traffic:

```
Switch# configure t
Switch(config)# interface fastEthernet 6/1
Switch(config-if)# mac access-group simple-mac-acl out
```

Displaying an ACL Configuration on a Layer 2 Interface

To display information about an ACL configuration on Layer 2 interfaces, perform one of these tasks:

Command	Purpose
`Switch# show ip interface [interface-name]`	Shows the IP access group configuration on the interface.
`Switch# show mac access-group interface [interface-name]`	Shows the MAC access group configuration on the interface.
`Switch# show access-group mode interface [interface-name]`	Shows the access group mode configuration on the interface.

This example shows that the IP access group simple-ip-acl is configured on the inbound direction of interface fa6/1:

```
Switch# show ip interface fast 6/1
FastEthernet6/1 is up, line protocol is up
  Inbound  access list is simple-ip-acl
  Outgoing access list is not set
```

This example shows that MAC access group simple-mac-acl is configured on the inbound direction of interface fa6/1:

```
Switch# show mac access-group interface fast 6/1
Interface FastEthernet6/1:
   Inbound access-list is simple-mac-acl
   Outbound access-list is not set
```

This example shows that access group merge is configured on interface fa6/1:

```
Switch# show access-group mode interface fast 6/1
Interface FastEthernet6/1:
   Access group mode is: merge
```

Using PACL with VLAN Maps and Router ACLs

For output PACLs, there is no interaction with VACL or output Router ACLs. (See the restrictions listed in the "PACL Configuration Guidelines" section on page 39-33.) For input PACLs, however, the interaction with Router ACLs and VACLs depends on the interface access group mode as shown in Table 39-1.

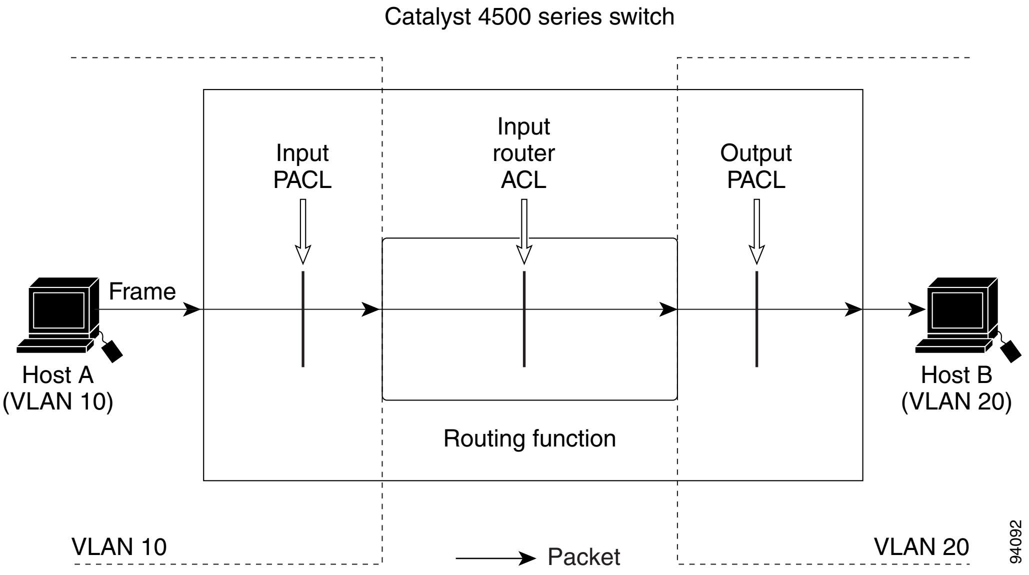
Table 39-1 Interaction Between PACLs, VACLs and Router ACLs

ACL Type(s)	Input PACL		
	prefer port mode	prefer vlan mode	merge mode
1. Input Router ACL	PACL applied	Input Router ACL applied	PACL, Input Router ACL (merged) applied in order (ingress)
2. VACL	PACL applied	VACL applied	PACL, VACL (merged) applied in order (ingress)
3. VACL + Input Router ACL	PACL applied	VACL + Input Router ACL applied	PACL, VACL, Input Router ACL (merged) applied in order (ingress)

Each ACL Type listed in Table 39-1 is synonymous with a different scenario, as explained in the following discussion.

Scenario 1: Host A is connected to an interface in VLAN 20, which has an SVI configured. The interface has input PACL configured, and the SVI has input Router ACL configured as shown in Figure 39-7:

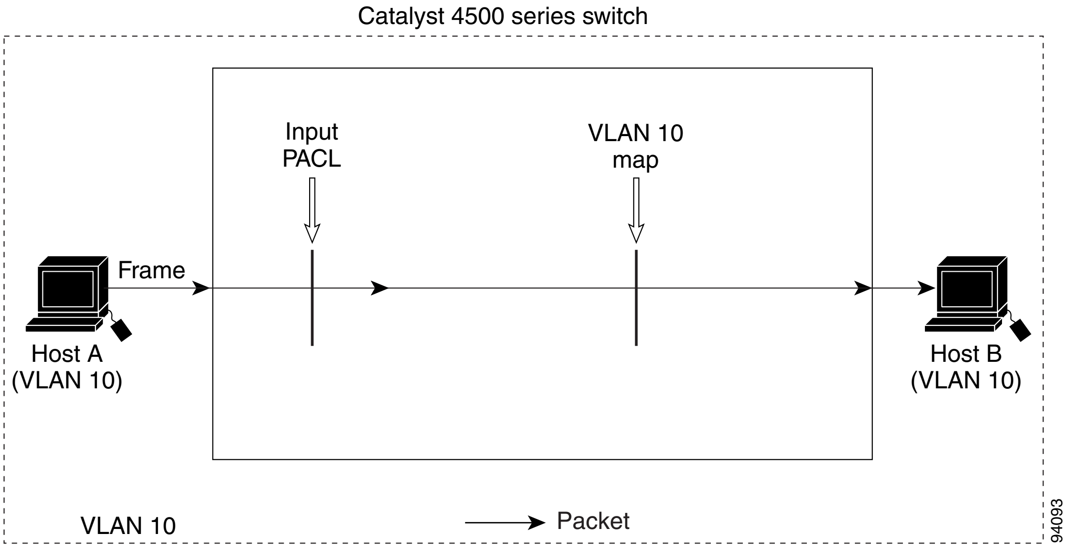
Figure 39-7 Scenario 1: PACL Interaction with an Input Router ACL



If the interface access group mode is prefer port, then only the input PACL is applied on the ingress traffic from Host A. If the mode is prefer vlan, then only the input Router ACL is applied to ingress traffic from Host A that requires routing. If the mode is merge, then the input PACL is first applied to the ingress traffic from Host A, and the input Router ACL is applied on the traffic that requires routing.

Scenario 2: Host A is connected to an interface in VLAN 10, which has a VACL (VLAN Map) configured and an input PACL configured as shown in Figure 39-8:

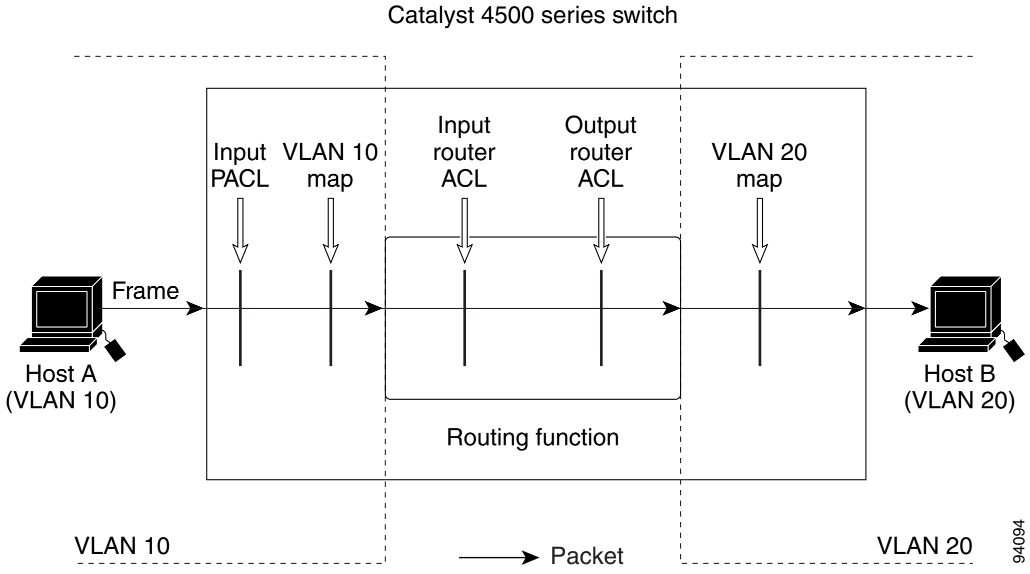
Figure 39-8 Scenario 2: PACL Interaction with a VACL



If the interface access group mode is prefer port, then only the input PACL is applied on the ingress traffic from Host A. If the mode is prefer vlan, then only the VACL is applied to the ingress traffic from Host A. If the mode is merge, the input PACL is first applied to the ingress traffic from Host A, and the VACL is applied on the traffic.

Scenario 3: Host A is connected to an interface in VLAN 10, which has a VACL and an SVI configured. The SVI has an input Router ACL configured and the interface has an input PACL configured, as shown in Figure 39-9:

Figure 39-9 Scenario 3: VACL and Input Router ACL



If the interface access group mode is prefer port, then only the input PACL is applied on the ingress traffic from Host A. If the mode is prefer vlan, then the merged results of the VACL and the input Router ACL are applied to the ingress traffic from Host A. If the mode is merge, the input PACL is first applied to the ingress traffic from Host A, the VACL is applied on the traffic and finally, and the input Router ACL is applied to the traffic that needs routing. (that is, the merged results of the input PACL, VACL, and input Router ACL are applied to the traffic).

CHAPTER 40

Configuring Private VLANs

Note Supervisor Engine 6-E does not support community PVLAN, isolated PVLAN trunk, and promiscuous trunk ports.

This chapter describes private VLANs (PVLANs) on Catalyst 4500 series switches. It also provides restrictions, procedures, and configuration examples.

This chapter includes the following major sections:

- Command List, page 40-1
- Overview of PVLANs, page 40-2
- Configuring PVLANs, page 40-9

For information on how to troubleshoot PVLANs, refer to the "Troubleshooting Private VLANs" section on page 51-39.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Command List

This table lists the commands most commonly used with PVLANs.

Command	Purpose	Location
private-vlan {**community** | **isolated** | **primary**}	Configures a VLAN as a PVLAN.	Configuring a VLAN as a PVLAN, page 40-12
private-vlan association {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Associates the secondary VLAN with the primary VLAN. The list can contain only one VLAN.	Associating a Secondary VLAN with a Primary VLAN, page 40-14

Command	Purpose	Location
show vlan private-vlan [**type**]	Verifies the configuration.	Configuring a VLAN as a PVLAN, page 40-12 Associating a Secondary VLAN with a Primary VLAN, page 40-14
show interface private-vlan mapping	Verifies the configuration.	Permitting Routing of Secondary VLAN Ingress Traffic, page 40-20
switchport mode private-vlan {**host** | **promiscuous** | **trunk promiscuous** | **trunk** [**secondary**]}	Configures a Layer 2 interface as a PVLAN port.	Configuring PVLANs, page 40-9
switchport private-vlan mapping [**trunk**] *primary_vlan_ID* {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Maps the PVLAN promiscuous port to a primary VLAN and to selected secondary VLANs.	Configuring a Layer 2 Interface as a PVLAN Promiscuous Port, page 40-15 Configuring a Layer 2 Interface as a Promiscuous Trunk Port, page 40-19
Switch(config-if)# **switchport private-vlan host-association** *primary_vlan_ID secondary_vlan_ID*	Associates the Layer 2 interface with a PVLAN.	Configuring a Layer 2 Interface as a PVLAN Host Port, page 40-16
switchport private-vlan association trunk *primary_vlan_ID secondary_vlan_ID*	Configures association between primary VLANs and secondary VLANs the PVLAN trunk port with a PVLAN.	Configuring a Layer 2 Interface as a PVLAN Trunk Port, page 40-17
switchport private-vlan trunk allowed vlan *vlan_list* **all** | **none** | [**add** | **remove** | **except**] *vlan_atom*[,*vlan_atom...*]	Configures a list of allowed normal VLANs on a PVLAN trunk port.	Configuring a Layer 2 Interface as a PVLAN Trunk Port, page 40-17
switchport private-vlan trunk native vlan *vlan_id*	Configures a VLAN to which untagged packets (as in IEEE 802.1Q tagging) are assigned on a PVLAN trunk port.	Configuring a Layer 2 Interface as a PVLAN Trunk Port, page 40-17

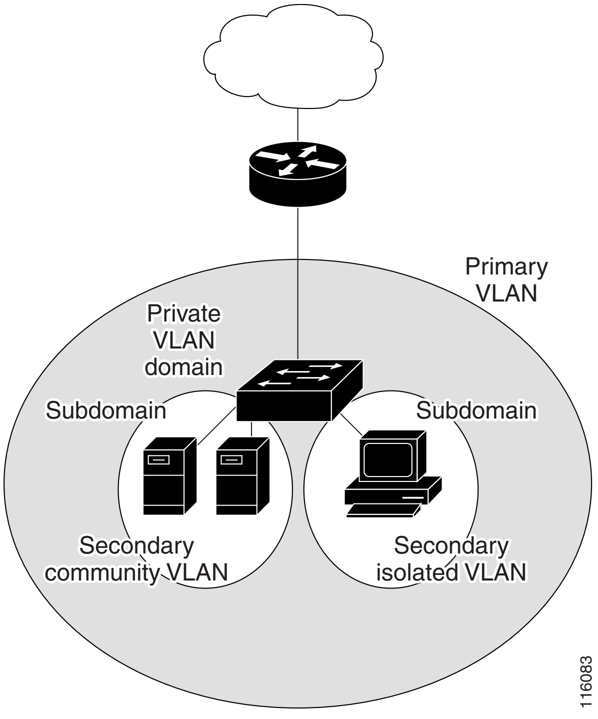
Overview of PVLANs

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

- The switch supports up to 1005 active VLANs. If a service provider assigns one VLAN per customer, this limits the numbers of customers the service provider can support.
- To enable IP routing, each VLAN is assigned a subnet address space or a block of addresses, which can result in wasting the unused IP addresses, and cause IP address management problems.

Using private VLANs provides scalability and IP address management benefits for service providers and Layer 2 security for customers. Private VLANs partition a regular VLAN domain into subdomains. A subdomain is represented by a pair of VLANs: a *primary* VLAN and a *secondary* VLAN. A private VLAN can have multiple VLAN pairs, one pair for each subdomain. All VLAN pairs in a private VLAN share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another. See Figure 40-1.

Figure 40-1 Private-VLAN Domain



There are two types of secondary VLANs:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other communities at the Layer 2 level.

A promiscuous port can serve only one primary VLAN, one isolated VLAN, and multiple community VLANs. Layer 3 gateways are typically connected to the switch through a promiscuous port.

In a switched environment, you can assign an individual private VLAN and associated IP subnet to each individual or common group of end stations. The end stations need to communicate only with a default gateway to communicate outside the private VLAN.

You can use private VLANs to control access to end stations in these ways:

- Configure selected interfaces connected to end stations as isolated ports to prevent any communication at Layer 2. For example, if the end stations are servers, this configuration prevents Layer 2 communication between the servers.
- Configure interfaces connected to default gateways and selected end stations (such as, backup servers) as promiscuous ports to allow all end stations access to a default gateway.
- Reduce VLAN and IP subnet consumption; you can prevent traffic between end stations even though they are in the same VLAN and IP subnet.

  With a promiscuous port, you can connect a wide range of devices as access points to a PVLAN. For example, you can connect a promiscuous port to the server port of a LocalDirector to connect an isolated VLAN or a number of community VLANs to the server. 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 PVLAN servers from an administration workstation.

This section includes the following topics:

- Definition Table, page 40-4
- Standard Trunk Ports, page 40-5
- Private VLANs across Multiple Switches, page 40-5
- Private-VLAN Interaction with Other Features, page 40-8

Definition Table

Term	Definition
Private VLANs	Private VLANs are sets of VLAN pairs that share a common primary identifier and provide a mechanism for achieving layer-2 separation between ports while sharing a single layer-3 router port and IP subnet.
Secondary VLAN	A type of VLAN used to implement private VLANs. Secondary VLANs are associated with a primary VLAN, and are used to carry traffic from hosts to other allowed hosts or to routers.
Community Port	A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with promiscuous ports. These interfaces are isolated at Layer 2 from all other interfaces in other communities and from isolated ports within their private VLAN.
Community VLAN	Community VLAN—A community VLAN is a secondary VLAN that carries upstream traffic from the community ports to the promiscuous port gateways and to other host ports in the same community. You can configure multiple community VLANs in a private VLAN. **Note** Supervisor Engine 6-E does not support community PVLAN.
Isolated Port	An isolated port is a host port that belongs to an isolated secondary VLAN. It has complete Layer 2 separation from other ports within the same private VLAN, except for the promiscuous ports. Private VLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports.
Isolated VLAN	Isolated VLAN —A private VLAN has only one isolated VLAN. An isolated VLAN is a secondary VLAN that carries unidirectional traffic upstream from the hosts toward the promiscuous ports and the gateway.

Term	Definition
Primary VLAN	Primary VLAN—A private VLAN has only one primary VLAN. Every port in a private VLAN is a member of the primary VLAN. The primary VLAN carries unidirectional traffic downstream from the promiscuous ports to the (isolated and community) host ports and to other promiscuous ports.
Private VLAN Trunk Port	A PVLAN trunk port can carry multiple secondary (isolated only) and non-PVLANs. Packets are received and transmitted with secondary or regular VLAN tags on the PVLAN trunk ports. **Note** Only IEEE 802.1q encapsulation is supported.
Promiscuous Port	A promiscuous port belongs to the primary VLAN and can communicate with all interfaces, including the community and isolated host ports and private VLAN trunk ports that belong to the secondary VLANs associated with the primary VLAN.
Promiscuous Trunk Port	A promiscuous trunk port can carry multiple primary and normal VLANs. Packets are received and transmitted with primary or regular VLAN tags. Other than that, the port behaves just like a promiscuous access port. **Note** Only IEEE 802.1q encapsulation is supported. **Note** Supervisor Engine 6-E does not support promisuous trunk port.

Private VLANs across Multiple Switches

This section discusses the following topics:

- Standard Trunk Ports, page 40-5
- Private VLAN Trunks, page 40-6

Standard Trunk Ports

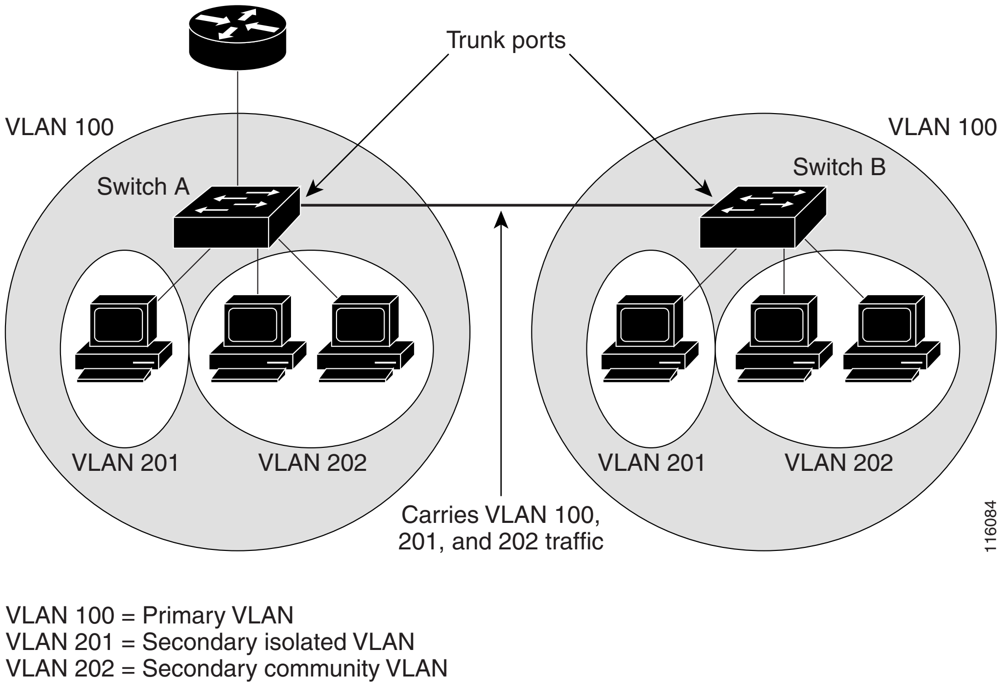
As with regular VLANs, private VLANs can span multiple switches. A trunk port carries the primary VLAN and secondary VLANs to a neighboring switch. The trunk port treats the private VLAN as any other VLAN. A feature of private VLANs across multiple switches is that traffic from an isolated port in switch A does not reach an isolated port on Switch B. See Figure 40-2.

To maintain the security of your private-VLAN configuration and to avoid other use of the VLANs configured as private VLANs, configure private VLANs on all intermediate devices, including devices that have no private-VLAN ports.

Note Trunk ports carry traffic from regular VLANs and also from primary, isolated, and community VLANs.

Note You should use standard trunk ports if both switches undergoing trunking support PVLANs.

Figure 40-2 Private VLANs across Switches



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

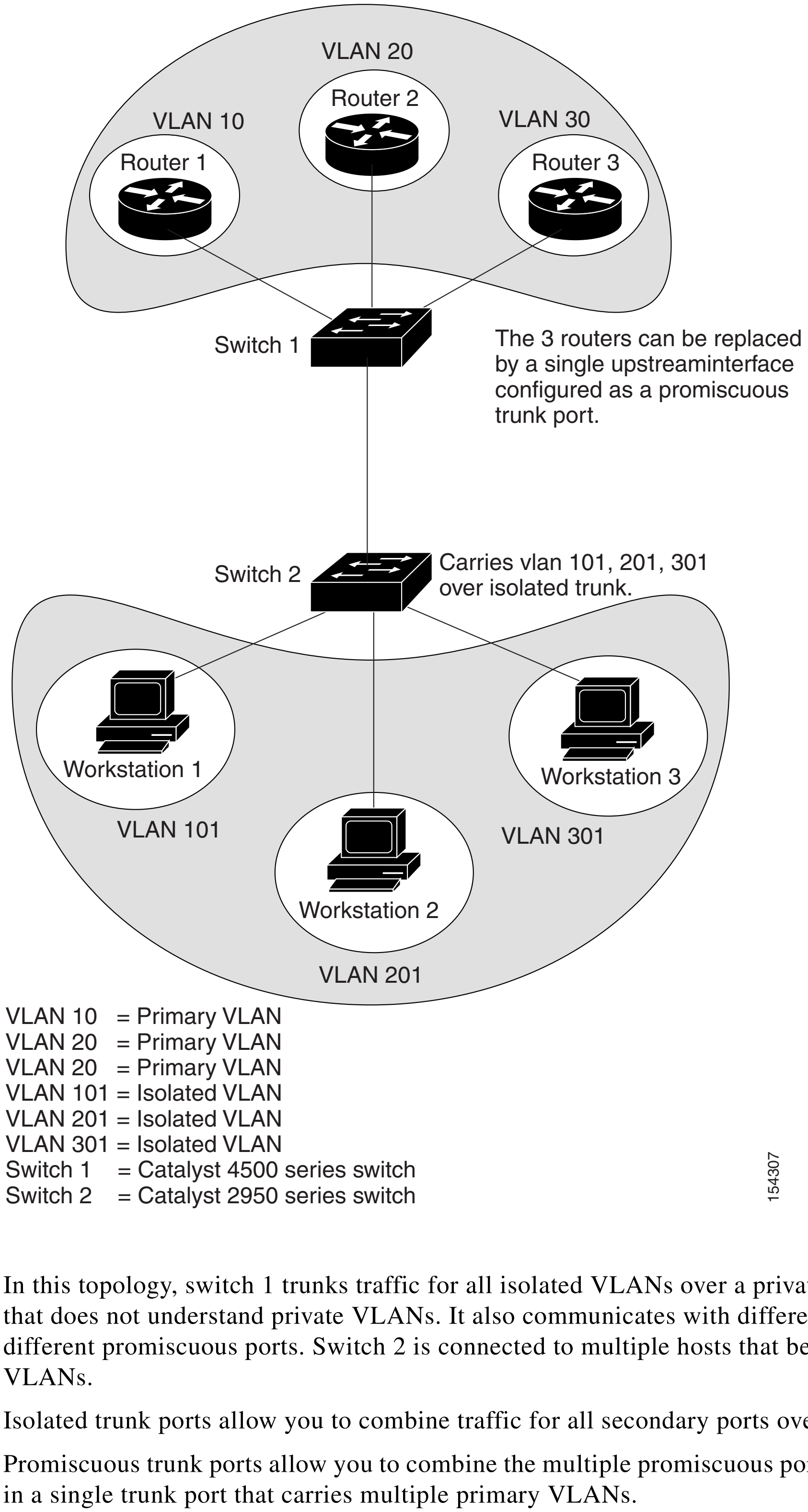
Private VLAN Trunks

A private VLAN isolated trunk is used when you want a private VLAN port to carry multiple secondary VLANs.

Note Supervisor Engine 6-E does *not* support isolated trunk port.

Figure 40-3 provides a typical topology.

Figure 40-3 Private VLAN Trunk Topology



In this topology, switch 1 trunks traffic for all isolated VLANs over a private VLAN trunk to Switch 2 that does not understand private VLANs. It also communicates with different routers connected to different promiscuous ports. Switch 2 is connected to multiple hosts that belong to different secondary VLANs.

Isolated trunk ports allow you to combine traffic for all secondary ports over a trunk.

Promiscuous trunk ports allow you to combine the multiple promiscuous ports required in this topology in a single trunk port that carries multiple primary VLANs.

Private-VLAN Interaction with Other Features

Private VLANs have specific interaction with some other features, described in these sections:

- PVLANs and VLAN ACL/QoS, page 40-8
- Private VLANs and Unicast, Broadcast, and Multicast Traffic, page 40-8
- Private VLANs and SVIs, page 40-9

You should also see the "PVLAN Configuration Guidelines and Restrictions" section on page 40-10 for details.

PVLANs and VLAN ACL/QoS

PVLAN ports use primary and secondary VLANs, as follows:

- A packet received on a PVLAN host port belongs to the secondary VLAN.
- A packet received on a PVLAN trunk port belongs to the secondary VLAN if the packet is tagged with a secondary VLAN or if the packet is untagged and the native VLAN on the port is a secondary VLAN.

A packet received on a PVLAN host or trunk port and assigned to a secondary VLAN is bridged on the secondary VLAN. Because of this bridging, the secondary VLAN ACL as well as the secondary VLAN QoS (on input direction) apply.

When a packet is transmitted out of a PVLAN host or trunk port, the packet logically belongs to the primary VLAN. This relationship applies even though the packet may be transmitted with the secondary VLAN tagging for PVLAN trunk ports. In this situation, the primary VLAN ACL and the primary VLAN QoS on output apply to the packet.

- Similarly, a packet received on a PVLAN promiscuous access port belongs to primary VLAN.
- A packet received on a PVLAN promiscuous trunk port could belong to the primary VLAN or normal VLAN depending on incoming VLAN.

For traffic flowing in normal VLAN on promiscuous trunk ports, normal VLAN ACL and QoS policies apply. For traffic flowing in a private VLAN domain, a packet received on a promiscuous port is bridged in primary VLAN. Therefore, the primary VLAN ACL and QoS policies apply on input.

When a packet is transmitted out of a promiscuous trunk port, the packet could logically belong to secondary VLAN if received from a secondary port, or in primary VLAN if bridged from another promiscuous port. Because we cannot differentiate between both packets, all VLAN QoS policies are ignored on packets egressing promiscuous trunk ports.

Private VLANs and Unicast, Broadcast, and Multicast Traffic

In regular VLANs, devices in the same VLAN can communicate with each other at the Layer 2 level, but devices connected to interfaces in different VLANs must communicate at the Layer 3 level. In private VLANs, the promiscuous ports are members of the primary VLAN, while the host ports belong to secondary VLANs. Because the secondary VLAN is associated to the primary VLAN, members of the these VLANs can communicate with each other at the Layer 2 level.

In a regular VLAN, broadcasts are forwarded to all ports in that VLAN. Private VLAN broadcast forwarding depends on the port sending the broadcast:

- An isolated port sends a broadcast only to the promiscuous ports or trunk ports.
- A community port sends a broadcast to all promiscuous ports, trunk ports, and ports in the same community VLAN.
- A promiscuous port sends a broadcast to all ports in the private VLAN (other promiscuous ports, trunk ports, isolated ports, and community ports).

Multicast traffic is routed or bridged across private-VLAN boundaries and within a single community VLAN. Multicast traffic is not forwarded between ports in the same isolated VLAN or between ports in different secondary VLANs.

Private VLANs and SVIs

In a Layer 3 switch, a switch virtual interface (SVI) represents the Layer 3 interface of a VLAN. Layer 3 devices communicate with a private VLAN only through the primary VLAN and not through secondary VLANs. Configure Layer 3 VLAN interfaces (SVIs) only for primary VLANs. You cannot configure Layer 3 VLAN interfaces for secondary VLANs. SVIs for secondary VLANs are inactive while the VLAN is configured as a secondary VLAN.

- If you try to configure a VLAN with an active SVI as a secondary VLAN, the configuration is not allowed until you disable the SVI.
- If you try to create an SVI on a VLAN that is configured as a secondary VLAN and the secondary VLAN is already mapped at Layer 3, the SVI is not created, and an error is returned. If the SVI is not mapped at Layer 3, the SVI is created, but it is automatically shut down.

When the primary VLAN is associated with and mapped to the secondary VLAN, any configuration on the primary VLAN is propagated to the secondary VLAN SVIs. For example, if you assign an IP subnet to the primary VLAN SVI, this subnet is the IP subnet address of the entire private VLAN.

Configuring PVLANs

These sections describe how to configure PVLANs:

Tasks for Configuring Private VLANs

To configure a PVLAN, follow these steps:

Step 1 Set VTP mode to transparent. See the "Disabling VTP (VTP Transparent Mode)" section on page 13-15.

Step 2 Create the secondary VLANs. See the "Configuring a VLAN as a PVLAN" section on page 40-12.

Step 3 Create the primary VLAN. See the "Configuring a VLAN as a PVLAN" section on page 40-12.

Step 4 Associate the secondary VLAN to the primary VLAN. See the "Associating a Secondary VLAN with a Primary VLAN" section on page 40-14.

Note Only one isolated VLAN can be mapped to a primary VLAN, but more than one community VLAN can be mapped to a primary VLAN.

Step 5 Configure an interface as an isolated or community host or trunk port. See the "Configuring a Layer 2 Interface as a PVLAN Host Port" section on page 40-16 and "Configuring a Layer 2 Interface as a PVLAN Trunk Port" section on page 40-17.

Step 6 Associate the isolated port or community port to the primary-secondary VLAN pair. See the "Associating a Secondary VLAN with a Primary VLAN" section on page 40-14.

Step 7 Configure an interface as a promiscuous port. See the "Configuring a Layer 2 Interface as a PVLAN Promiscuous Port" section on page 40-15.

Step 8 Map the promiscuous port to the primary-secondary VLAN pair. See the "Configuring a Layer 2 Interface as a PVLAN Promiscuous Port" section on page 40-15.

Step 9 If you plan to use inter-VLAN routing, configure the primary SVI, and map secondary VLANs to the primary. See the "Permitting Routing of Secondary VLAN Ingress Traffic" section on page 40-20.

Step 10 Verify private-VLAN configuration. See the "Switch#" section on page 40-21.

Default Private-VLAN Configuration

No private VLANs are configured.

PVLAN Configuration Guidelines and Restrictions

Follow these guidelines when configuring PVLANs:

- To configure a PVLAN correctly, enable VTP in transparent mode.

  You cannot change the VTP mode to client or server for PVLANs.
- Do not include VLAN 1 or VLANs 1002 through 1005 in PVLANs.
- Use only PVLAN commands to assign ports to primary, isolated, or community VLANs.

  Layer 2 interfaces on primary, isolated, or community VLANs are inactive in PVLANs. Layer 2 trunk interfaces remain in the STP forwarding state.
- You cannot configure Layer 3 VLAN interfaces for secondary VLANs.

Layer 3 VLAN interfaces for isolated and community (secondary) 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, its associated EtherChannel configuration is inactive.
- 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 is part of the PVLAN configuration.
- To prevent spanning tree loops due to misconfigurations, enable PortFast on the PVLAN trunk ports with the **spanning-tree portfast trunk** command.
- Any VLAN ACL configured on a secondary VLAN is effective in the input direction, and any VLAN ACL configured on the primary VLAN associated with the secondary VLAN is effective in the output direction.
- You can stop Layer 3 switching on an isolated or community VLAN by deleting the mapping of that VLAN with its primary VLAN.
- PVLAN ports can be on different network devices as long as the devices are trunk-connected and the primary and secondary VLANs remain associated with the trunk
- Isolated ports on two different devices cannot communicate with each other, but community VLAN ports can.
- 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 monitor egress or ingress traffic separately.

  For more information about SPAN, see Chapter 43, “Configuring SPAN and RSPAN.”
- A primary VLAN can be associated with multiple community VLANs, but only one isolated VLAN.
- An isolated or community VLAN can be associated with only one primary VLAN.
- If you delete a VLAN used in a private VLAN configuration, the private VLAN ports associated with the VLAN become inactive.
- VTP does not support private VLANs. You must configure private VLANs on each device in which you plan to use private VLAN ports.
- To maintain the security of your PVLAN configuration and avoid other use of VLANs configured as PVLANs, configure PVLANs on all intermediate devices, even if the devices have no PVLAN ports.
- Prune the PVLANs from trunks on devices that carry no traffic in the PVLANs.
- With port ACLS functionality available, you can apply Cisco IOS ACLS to secondary VLAN ports and Cisco IOS ACLS to PVLANS (VACLs). For more information on VACLs, see Chapter 39, “Configuring Network Security with ACLs.”
- You can apply different quality of service (QoS) configurations to primary, isolated, and community VLANs. (See Chapter 32, “Configuring Quality of Service.”) Cisco IOS ACLs applied to the Layer 3 VLAN interface of a primary VLAN automatically apply to the associated isolated and community VLANs.
- On a PVLAN trunk port a secondary VLAN ACL is applied on ingress traffic and a primary VLAN ACL is applied on egress traffic.
- On a promiscuous port the primary VLAN ACL is applied on ingress traffic.

- Both PVLAN secondary and promiscuous trunk ports support only IEEE 802.1q encapsulation.
- Community VLANs cannot be propagated or carried over private VLAN trunks.
- ARP entries learned on Layer 3 PVLAN interfaces are termed “sticky” ARP entries (we recommend that you display and verify PVLAN interface ARP entries).
- For security reasons, PVLAN port sticky ARP entries do not age out. Connecting a device with a different MAC address but with the same IP address generates an error message and the ARP entry is not created.
- Because PVLAN port sticky ARP entries do not age out, you must manually remove the entries if you change the MAC address. To overwrite a sticky ARP entry, first delete the entry with the **no arp** command, then overwrite the entry with the **arp** command.
- In a DHCP environment, if you shut down your PC, it is not possible to give your IP address to someone else. To solve this problem, the Catalyst 4500 series switch supports the **no ip sticky-arp** command. This command promotes IP address overwriting and reuse in a DHCP environment.
- Normal VLANs can be carried on a promiscuous trunk port.
- The default native VLAN for promiscuous trunk port is VLAN 1, the management VLAN. All untagged packets are forwarded in the native VLAN. Either the primary VLANs or a regular VLAN can be configured as native VLAN.
- Promiscuous trunks cannot be configured to carry secondary VLANs. If a secondary VLAN is specified in the allowed VLAN list, the configuration is accepted but the port is not operational/forwarding in the secondary VLAN. This includes even those VLANs that are of scondary but not associated with any primary VLAN on given port.
- On a promiscuous trunk port, the primary VLAN ACL and QoS are applied on ingress traffic coming in primary VLANs.
- On a promiscuous trunk port, no VLAN ACL or QoS is applied to the egress traffic. This is because for upstream direction, traffic in private VLAN logically flows in the secondary VLAN. Due to VLAN translation in hardware, information about received secondary VLANs has been lost. Hence, no policies are applied. This restriction also applies to traffic bridged from other ports in the same primary VLANs.
- Do not configure port security on PVLAN promiscuous trunk port and vice versa.

  If port security is enabled on a promiscuous trunk port, that port may behave in an unpredictable manner because this functionality is not supported.
- Do not configure IEEE 802.1X on a PVLAN promiscuous trunk port.

Configuring a VLAN as a PVLAN

Note Supervisor Engine 6-E does not support community PVLAN.

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

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters configuration mode.
Step 2	`Switch(config)# vlan vlan_ID`	Enters VLAN configuration mode.

	Command	Purpose
Step 3	`Switch(config-vlan)#` **`private-vlan`** `{`**`community`** `|` **`isolated`** `|` **`primary`**`}`	Configures a VLAN as a PVLAN. • This command does not take effect until you exit VLAN configuration submode. You can use the **no** keyword to clear PVLAN status. **Note** Supervisor Engine 6-E does *not* support community and isolated PVLAN trunk ports.
Step 4	`Switch(config-vlan)#` **`end`**	Exits VLAN configuration mode.
Step 5	`Switch#` **`show vlan private-vlan`** `[`**`type`**`]`	Verifies the configuration.

This example shows how to configure VLAN 202 as a primary VLAN and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 202
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# end
Switch# 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:

```
Switch# configure terminal
Switch(config)# vlan 303
Switch(config-vlan)# private-vlan community
Switch(config-vlan)# end
Switch# 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:

```
Switch# configure terminal
Switch(config)# vlan 440
Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# end
Switch# show vlan private-vlan

Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
        303       community
        440       isolated
```

Associating a Secondary VLAN with a Primary VLAN

To associate secondary VLANs with a primary VLAN, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters configuration mode.
Step 2	Switch(config)# **vlan** *primary_vlan_ID*	Enters VLAN configuration mode for the primary VLAN.
Step 3	Switch(config-vlan)# **private-vlan association** {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Associates the secondary VLAN with the primary VLAN. The list can contain only one VLAN. You can use the **no** keyword to clear all secondary associations.
Step 4	Switch(config-vlan)# **end**	Exits VLAN configuration mode.
Step 5	Switch# **show vlan private-vlan** [**type**]	Verifies the configuration.

When you associate secondary VLANs with a primary VLAN, note the following:

- 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:

```
Switch# configure terminal
Switch(config)# vlan 202
Switch(config-vlan)# private-vlan association 303-307,309,440
Switch(config-vlan)# end
Switch# 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
```

Note The secondary VLAN 308 has no associated primary VLAN.

Configuring a Layer 2 Interface as a PVLAN Promiscuous Port

To configure a Layer 2 interface as a PVLAN promiscuous port, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*	Specifies the LAN interface to configure.
Step 3	Switch(config-if)# **switchport mode private-vlan** {**host** | **promiscuous** | **trunk promiscuous** | **trunk** [**secondary**]}	Configures a Layer 2 interface as a PVLAN promiscuous port.
Step 4	Switch(config-if)# [**no**] **switchport private-vlan mapping** [**trunk**] *primary_vlan_ID* {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Maps the PVLAN promiscuous port to a primary VLAN and to selected secondary VLANs.
Step 5	Switch(config-if)# **end**	Exits configuration mode.
Step 6	Switch# **show interfaces** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* **switchport**	Verifies the configuration.

Note The maximum number of unique private VLAN pairs supported by the **switchport private-vlan mapping trunk** command above is 500. For example, one thousand secondary VLANs could map to one primary VLAN, or one thousand secondary VLANs could map one to one to one thousand primary VLANs.

When you configure a Layer 2 interface as a PVLAN promiscuous port, note the following:

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single PVLAN ID or a hyphenated range of PVLAN IDs.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to map the secondary VLANs to the PVLAN promiscuous port.
- Use the **remove** keyword with a *secondary_vlan_list* to clear the mapping between secondary VLANs and the PVLAN promiscuous port.

This example shows how to configure interface FastEthernet 5/2 as a PVLAN promiscuous port, map it to a PVLAN, and verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/2
Switch(config-if)# switchport mode private-vlan promiscuous
Switch(config-if)# switchport private-vlan mapping 200 2
Switch(config-if)# end
Switch# show interfaces fastethernet 5/2 switchport
Name:Fa5/2
Switchport:Enabled
Administrative Mode:private-vlan promiscuous
Operational Mode:private-vlan promiscuous
Administrative Trunking Encapsulation:negotiate
Operational Trunking Encapsulation:native
Negotiation of Trunking:Off
Access Mode VLAN:1 (default)
Trunking Native Mode VLAN:1 (default)
Voice VLAN:none
```

```
Administrative Private VLAN Host Association:none
Administrative Private VLAN Promiscuous Mapping:200 (VLAN0200) 2 (VLAN0002)
Private VLAN Trunk Native VLAN:none
Administrative Private VLAN Trunk Encapsulation:dot1q
Administrative Private VLAN Trunk Normal VLANs:none
Administrative Private VLAN Trunk Private VLANs:none
Operational Private VLANs:
  200 (VLAN0200) 2 (VLAN0002)
Trunking VLANs Enabled:ALL
Pruning VLANs Enabled:2-1001
Capture Mode Disabled
Capture VLANs Allowed:ALL
```

Configuring a Layer 2 Interface as a PVLAN Host Port

To configure a Layer 2 interface as a PVLAN host port, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters configuration mode.
Step 2	Switch(config)# **interface** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port*	Specifies the LAN port to configure.
Step 3	Switch(config-if)# **switchport mode private-vlan** {**host** | **promiscuous** | **trunk promiscuous** | **trunk** [**secondary**]}	Configures a Layer 2 interface as a PVLAN host port.
Step 4	Switch(config-if)# [**no**] **switchport private-vlan host-association** *primary_vlan_ID secondary_vlan_ID*	Associates the Layer 2 interface with a PVLAN. You can use the **no** keyword to delete all associations from the primary VLAN.
Step 5	Switch(config-if)# **end**	Exits configuration mode.
Step 6	Switch# **show interfaces** {**fastethernet** | **gigabitethernet** | **tengigabitethernet**} *slot/port* **switchport**	Verifies the configuration.

This example shows how to configure interface FastEthernet 5/1 as a PVLAN host port and verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 202 440
Switch(config-if)# end
```

```
Switch# show interfaces fastethernet 5/1 switchport
Name: Fa5/1
Switchport: Enabled
Administrative Mode: private-vlan host
Operational Mode: private-vlan host
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Appliance trust: none
```

```
Administrative Private Vlan
  Host Association: 202 (VLAN0202) 440 (VLAN0440)
  Promiscuous Mapping: none
  Trunk encapsulation : dot1q
  Trunk vlans:
Operational private-vlan(s):
  202 (VLAN0202) 440 (VLAN0440)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
```

Configuring a Layer 2 Interface as a PVLAN Trunk Port

To configure a Layer 2 interface as a PVLAN trunk port, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** {**`fastethernet`** | **`gigabitethernet`** | **`tengigabitethernet`**} *`slot/port`*	Specifies the LAN port to configure.
Step 3	`Switch(config-if)#` **`switchport mode private-vlan`** {**`host`** | **`promiscuous`** | **`trunk promiscuous`** | **`trunk`** [**`secondary`**]}	Configures a Layer 2 interface as a PVLAN trunk port.
Step 4	`Switch(config-if)#` [**`no`**] **`switchport private-vlan association trunk`** *`primary_vlan_ID secondary_vlan_ID`*	Configures association between primary VLANs and secondary VLANs the PVLAN trunk port with a PVLAN. **Note** Multiple PVLAN pairs can be specified using this command so that a PVLAN trunk port can carry multiple secondary VLANs. If an association is specified for the existing primary VLAN, the existing association is replaced. If there is no trunk association, any packets received on secondary VLANs are dropped. You can use the **no** keyword to delete all associations from the primary VLAN.
Step 5	`Switch(config-if)#` [**`no`**] **`switchport private-vlan trunk allowed vlan`** *`vlan_list`* **`all`** | **`none`** | [**`add`** | **`remove`** | **`except`**] *`vlan_atom`*[,*`vlan_atom...`*]	Configures a list of allowed normal VLANs on a PVLAN trunk port. You can use the **no** keyword to remove all allowed normal VLANs on a PVLAN trunk port.

	Command	Purpose
Step 6	`Switch(config-if)#` **`switchport private-vlan trunk native vlan`** *`vlan_id`*	Configures a VLAN to which untagged packets (as in IEEE 802.1Q tagging) are assigned on a PVLAN trunk port. If there is no native VLAN configured, all untagged packets are dropped. If the native VLAN is a secondary VLAN and the port does not have the association for the secondary VLAN, the untagged packets are dropped. You can use the **no** keyword to remove all native VLANs on a PVLAN trunk port.
Step 7	`Switch(config-if)#` **`end`**	Exits configuration mode.
Step 8	`Switch#` **`show interfaces`** {**`fastethernet`** | **`gigabitethernet`** | **`tengigabitethernet`**} *`slot/port`* **`switchport`**	Verifies the configuration.

This example shows how to configure interface FastEthernet 5/2 as a secondary trunk port, and verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/2
Switch(config-if)# switchport mode private-vlan trunk secondary
Switch(config-if)# switchport private-vlan trunk native vlan 10
Switch(config-if)# switchport private-vlan trunk allowed vlan 10. 3-4
Switch(config-if)# switchport private-vlan association trunk 3 301
Switch(config-if)# end
Switch# show interfaces fastethernet 5/2 switchport
Name: Fa5/2
    Switchport: Enabled
    Administrative Mode: private-vlan trunk secondary
    Operational Mode: private-vlan trunk secondary
    Administrative Trunking Encapsulation: negotiate
    Operational Trunking Encapsulation: dot1q
    Negotiation of Trunking: On
    Access Mode VLAN: 1 (default)
    Trunking Native Mode VLAN: 1 (default)
    Administrative Native VLAN tagging: enabled
    Voice VLAN: none
    Administrative private-vlan host-association: none A
    dministrative private-vlan mapping: none
    Administrative private-vlan trunk native VLAN: 10
    Administrative private-vlan trunk Native VLAN tagging: enabled
    Administrative private-vlan trunk encapsulation: dot1q
    Administrative private-vlan trunk normal VLANs: none
    Administrative private-vlan trunk associations:
        3 (VLAN0003) 301 (VLAN0301)
    Administrative private-vlan trunk mappings: none
    Operational private-vlan: none
    Operational Normal VLANs: none
    Trunking VLANs Enabled: ALL
    Pruning VLANs Enabled: 2-1001
    Capture Mode Disabled Capture VLANs Allowed: ALL

    Unknown unicast blocked: disabled
    Unknown multicast blocked: disabled
    Appliance trust: none
```

Configuring a Layer 2 Interface as a Promiscuous Trunk Port

Note Supervisor Engine 6-E does *not* support promiscuous trunk port.

To configure a Layer 2 interface as a PVLAN promiscuous trunk port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface {fastethernet | gigabitethernet | tengigabitethernet} slot/port`	Specifies the LAN interface to configure.
Step 3	`Switch(config-if)# switchport mode private-vlan {host | promiscuous | trunk promiscuous | trunk [secondary]}`	Configures a Layer 2 interface as a PVLAN promiscuous trunk port.
Step 4	`Switch(config-if)# [no] switchport private-vlan mapping [trunk] primary_vlan_ID {secondary_vlan_list | add secondary_vlan_list | remove secondary_vlan_list}`	Maps the PVLAN promiscuous port to a primary VLAN and to selected secondary VLANs. This command offers 3 levels of removal. See the examples that follow this table.
Step 5	`Switch(config-if)# end`	Exits configuration mode.
Step 6	`Switch# show interfaces {fastethernet | gigabitethernet | tengigabitethernet} slot/port switchport`	Verifies the configuration.

Note The maximum number of unique private VLAN pairs supported by the **switchport private-vlan mapping trunk** command above is 500. For example, one thousand secondary VLANs could map to one primary VLAN, or one thousand secondary VLANs could map one to one to one thousand primary VLANs.

Note By default, when you configure the mode to private VLAN trunk promiscuous, the native VLAN is set to 1.

The [**no**] **switchport private-vlan mapping** command provides the following three levels of removal:

- Remove one or more secondary VLANs from the list. For example:

```
Switch(config-if)# switchport private-vlan mapping trunk 2 remove 222
```

- Remove the entire mapping of PVLAN promiscuous trunk port to the specified primary VLAN (and all of its selected secondary VLANs). For example:

```
Switch(config-if)# no switchport private-vlan mapping trunk 2
```

- Remove the mapping of a PVLAN promiscuous trunk port to all previously configured primary VLANs (and all of their selected secondary VLANs). For example:

```
Switch(config-if)# no switchport private-vlan mapping trunk
```

When you configure a Layer 2 interface as a PVLAN promiscuous port, note the following:

- Multiple private VLAN pairs can be specified using the **switchport private-vlan mapping trunk** command so that a promiscuous trunk port can carry multiple primary VLANs.

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single PVLAN ID or a hyphenated range of PVLAN IDs.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to map the secondary VLANs to the PVLAN promiscuous port.
- Use the **remove** keyword with a *secondary_vlan_list* to clear the mapping between secondary VLANs and the PVLAN promiscuous port.

This example shows how to configure interface FastEthernet 5/2 as a promiscuous trunk port and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/2
Switch(config-if)# switchport mode private-vlan trunk promiscuous
Switch(config-if)# switchport private-vlan trunk native vlan 10
Switch(config-if)# switchport private-vlan trunk allowed vlan 10, 3-4
Switch(config-if)# switchport private-vlan mapping trunk 3 301, 302
Switch(config-if)# end
Switch# show interfaces fastethernet 5/2 switchport
Name: Fa5/2
Switchport: Enabled
Administrative Mode: private-vlan trunk promiscuous
Operational Mode: private-vlan trunk promiscuous
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: 10
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: 3-4,10
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings:
    3 (VLAN0003) 301 (VLAN0301)  302 (VLAN0302)
Operational private-vlan:
  3 (VLAN0003) 301 (VLAN0301) 302 (VLAN0302)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
```

Permitting Routing of Secondary VLAN Ingress Traffic

Note Isolated and community VLANs are both called secondary VLANs.

To permit routing of secondary VLAN ingress traffic, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **interface vlan** *primary_vlan_ID*	Enters interface configuration mode for the primary VLAN.
Step 3	Switch(config-if)# [**no**] **private-vlan mapping** *primary_vlan_ID* {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	To permit routing on the secondary VLAN ingress traffic, map the secondary VLAN to the primary VLAN. You can use the **no** keyword to delete all associations from the primary VLAN.
Step 4	Switch(config-if)# **end**	Exits configuration mode.
Step 5	Switch# **show interface private-vlan mapping**	Verifies the configuration.

When you permit routing on the secondary VLAN ingress traffic, note the following:

- 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:

```
Switch# configure terminal
Switch(config)# interface vlan 202
Switch(config-if)# private-vlan mapping add 303-307,309,440
Switch(config-if)# end
Switch# 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

Switch#
```

CHAPTER 41

Port Unicast and Multicast Flood Blocking

This chapter describes how to configure multicast and unicast flood blocking on the Catalyst 4000 family switch. This chapter contains these topics:

- Overview of Flood Blocking, page 41-1
- Configuring Port Blocking, page 41-1

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Overview of Flood Blocking

Occasionally, unknown unicast or multicast traffic is flooded to a switch port because a MAC address has timed out or has not been learned by the switch. (This condition is especially undesirable for a private VLAN isolated port.) To guarantee that no unicast and multicast traffic is flooded to the port, use the **switchport block unicast** and **switchport block multicast** commands to enable flood blocking on the switch.

Note The flood blocking feature is supported on all switched ports (including PVLAN ports) and is applied to all VLANs on which the port is forwarding.

Configuring Port Blocking

By default, a switch floods packets with unknown destination MAC addresses to all ports. If unknown unicast and multicast traffic is forwarded to a switch port, there might be security issues. To prevent forwarding such traffic, you can configure a port to block unknown unicast or multicast packets.

Note Blocking of unicast or multicast traffic is not automatically enabled on a switch port; you must explicitly configure it.

Blocking Flooded Traffic on an Interface

Note The interface can be a physical interface (for example, GigabitEthernet 1/1) or an EtherChannel group (such as port-channel 5). When you block multicast or unicast traffic for a port channel, it is blocked on all ports in the port channel group.

To disable the flooding of multicast and unicast packets to an interface, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and enter the type and number of the switchport interface (for example, GigabitEthernet 1/1).
Step 3	`Switch(config-if)#` **`switchport block multicast`**	Blocks unknown multicast forwarding to the port.
Step 4	`Switch(config-if)#` **`switchport block unicast`**	Blocks unknown unicast forwarding to the port.
Step 5	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.

Step 6	`Switch#`
	`show interface` *`interface-id`* **`switchport`**
		Verifies your entry.

Step 7	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

This example shows how to block unicast and multicast flooding on a GigabitEthernet interface1/1 and how to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# switchport block multicast
Switch(config-if)# switchport block unicast
Switch(config-if)# end
Switch# show interface gigabitethernet1/1 switchport
Name: Gi1/3
Switchport: Enabled

<output truncated>

Port Protected: On
Unknown Unicast Traffic: Not Allowed
Unknown Multicast Traffic: Not Allowed

Broadcast Suppression Level: 100
Multicast Suppression Level: 100
Unicast Suppression Level: 100
```

Resuming Normal Forwarding on a Port

To resume normal forwarding on a port, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and enter the type and number of the switchport interface (GigabitEthernet1/1)**.**
Step 3	`Switch(config-if)#` **`no switchport block multicast`**	Enables unknown multicast flooding to the port.
Step 4	`Switch(config-if)#` **`no switchport block unicast`**	Enables unknown unicast flooding to the port.
Step 5	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 6	`Switch#` **`show interface`** *`interface-id`* **`switchport`**	Verifies your entry.
Step 7	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

CHAPTER 42

Configuring Storm Control

This chapter describes how to configure port-based traffic control on the Catalyst 4500 series switch.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

This chapter consists of these sections:

- Overview of Storm Control, page 42-1
- Enabling Broadcast Storm Control, page 42-3
- Enabling Multicast Storm Control, page 42-4
- Disabling Broadcast Storm Control, page 42-6
- Disabling Multicast Storm Control, page 42-7
- Displaying Storm Control, page 42-7

Overview of Storm Control

This section contains the following subsections:

- Hardware-based Storm Control Implementation, page 42-2
- Software-based Storm Control Implementation, page 42-2

Storm control prevents LAN interfaces from being disrupted by a broadcast storm. A broadcast storm occurs when broadcast packets flood the subnet, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation or in the network configuration can cause a broadcast storm.

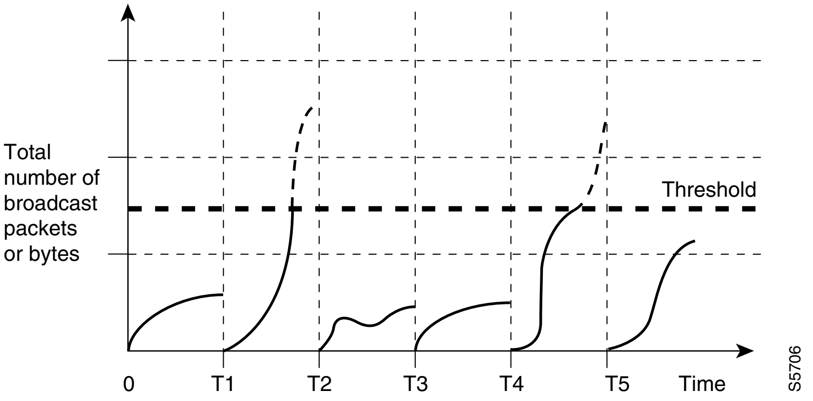
Note Storm control is supported in hardware on all ports on the WS-X4516 supervisor engine and Supervisor Engine 6-E. In contrast, the supervisor engines WS-X4515, WS-X4014, and WS-X4013+ support storm control in hardware on non-blocking gigabit ports and in software on all other ports, implying that the counters for these interfaces are approximate and computed. Multicast storm control is only supported on the WS-X4516 supervisor engine and Supervisor Engine 6-E.

Hardware-based Storm Control Implementation

Broadcast suppression uses filtering that measures broadcast activity in a subnet over a one-second interval and compares the measurement with a predefined threshold. If the threshold is reached, further broadcast activity is suppressed for the duration of the interval. Broadcast suppression is disabled by default.

Figure 42-1 shows the broadcast traffic patterns on a LAN interface over a given interval. In this example, broadcast suppression occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 42-1 Storm Control Example - Hardware-based Implementation



The broadcast suppression threshold numbers and the time interval combination make the broadcast suppression algorithm work with different levels of granularity. A higher threshold allows more broadcast packets to pass through.

Broadcast suppression on the Catalyst 4500 series switches (including Supervisor Engine 6-E) is implemented in hardware. The suppression circuitry monitors packets passing from a LAN interface to the switching bus. If the packet destination address is broadcast, then the broadcast suppression circuitry tracks the current count of broadcasts within the one-second interval, and when a threshold is reached, it filters out subsequent broadcast packets.

Because hardware broadcast suppression uses a bandwidth-based method to measure broadcast activity, the most significant implementation factor is setting the percentage of total available bandwidth that can be used by broadcast traffic. Because packets do not arrive at uniform intervals, the one-second interval during which broadcast activity is measured can affect the behavior of broadcast suppression.

Software-based Storm Control Implementation

When storm control is enabled on an interface, the switch monitors packets received on the interface and determines whether or not the packets are broadcast. The switch monitors the number of broadcast packets received within a one-second time interval. When the interface threshold is met, all incoming data traffic on the interface is dropped. This threshold is specified as a percentage of total available bandwidth that can be used by broadcast traffic. If the lower threshold is specified, all data traffic is forwarded as soon as the incoming traffic falls below that threshold.

Enabling Broadcast Storm Control

To enable storm control, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and enter the port to configure.
Step 3	`Switch(config-if)#` **`storm-control broadcast level`** `[`*`high level`*`] [`*`lower level`*`]`	Configures broadcast storm control. Specifies the upper threshold levels for broadcast traffic. The storm control action occurs when traffic utilization reaches this level. (Optional) Specifies the falling threshold level. The normal transmission restarts (if the action is filtering) when traffic drops below this level for interfaces that support software-based suppression. **Note** The **lower level** keyword does not apply to Supervisor Engine 6E implementations. **Note** For ports that perform hardware-based suppression, the lower threshold is ignored.
Step 4	`Switch(config-if)#` **`storm-control action`** `{`**`shutdown`** `|` **`trap`**`}`	Specifies the action to be taken when a storm is detected. The default is to filter out the broadcast traffic and not to send out traps. The **shutdown** keyword sets the port to error-disable state during a storm. If the recover interval is not set, the port remains in shutdown state. **Note** The **trap** keyword generates an SNMP trap when a storm is detected. This keyword is available but not supported in Cisco IOS Release 12.1(19)EW.
Step 5	`Switch(config-if)#` **`exit`**	Returns to configuration mode.
Step 6	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 7	`Switch#` **`show storm-control`** `[`**`interface`**`]` **`broadcast`**	Displays the number of packets suppressed.
Step 8	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

The following example shows how to enable storm control on interface.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fa3/1
Switch(config-if)# storm-control broadcast level 50
Switch(config-if)# end

Switch# show storm-control //Supervisor Engine 6-E
Interface  Filter State   Broadcast Multicast Level
---------  -------------  --------- --------- -----
Fi3/1      Forwarding     Enabled   Disabled  50.00%
```

```
Switch# show int fa2/1 capabilities //Supervisor Engine 6-E
FastEthernet2/1
  Model:                  WS-X4148-RJ45V-RJ-45
  Type:                   10/100BaseTX
  Speed:                  10,100,auto
  Duplex:                 half,full,auto
  Auto-MDIX:              no
  Trunk encap. type:      802.1Q
  Trunk mode:             on,off,desirable,nonegotiate
  Channel:                yes
  Broadcast suppression: percentage(0-100), hw
  Multicast suppression: percentage(0-100), hw <===== unique to Sup Engine 6-E systems
  Flowcontrol:            rx-(none),tx-(none)
  VLAN Membership:        static, dynamic
  Fast Start:             yes
  CoS rewrite:            yes
  ToS rewrite:            yes
  Inline power:           yes (Cisco Voice Protocol)
  SPAN:                   source/destination
  UDLD:                   yes
  Link Debounce:          no
  Link Debounce Time:     no
  Port Security:          yes
  Dot1x:                  yes
  Maximum MTU:            1552 bytes (Baby Giants)
  Multiple Media Types:  no
  Diagnostic Monitoring: N/A
```

Enabling Multicast Storm Control

Topics include:

- Multicast Suppression on the Supervisor Engine 6-E, page 42-4
- Multicast Suppression on the WS-X4516 Supervisor Engine, page 42-5
- Multicast Suppression on the WS-X4515, WS-X4014, and WS-X4013+ Supervisor Engines, page 42-6

Note Beginning with Cisco IOS Release 12.2(18)EW, the counters displayed with the **show interface counters storm-control** command includes any multicast packets that were dropped.

Multicast Suppression on the Supervisor Engine 6-E

Supervisor Engine 6-E supports per-interface multicast suppression. This allows the user to subject incoming multicast and broadcast traffic on an interface to suppression.

Note Multicast and broadcast suppression share a common threshold per interface.
Multicast suppression takes effect *only* if broadcast suppression is enabled.
Disabling broadcast suppression on an interface also disables multicast suppression.

To enable multicast suppression on a Supervisor Engine 6-E, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and enter the port to configure.
Step 3	`Switch(config-if)#` **`storm-control broadcast include multicast`**	Enables multicast suppression.
Step 4	`Switch(config-if)#` **`exit`**	Returns to configuration mode.
Step 5	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 6	`Switch#` **`show storm-control`**	Verifies the configuration.

The following example shows how to enable multicast suppression on ports that have broadcast suppression already enabled:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# int fa3/1
Switch(config-if)# storm-control broadcast include multicast
Switch(config-if)# end
Switch#
Switch# show storm-control
Interface  Filter State   Broadcast Multicast Level
---------  -------------  --------- --------- -----
Fi3/1      Forwarding     Enabled   Enabled   50.00%
```

Multicast Suppression on the WS-X4516 Supervisor Engine

Multicast suppression can be enabled on a WS-X4516 supervisor engine for all ports that have storm control enabled. Multicast suppression applies to all ports that have broadcast suppression configured on them. It also applies to ports that are configured for broadcast storm-control in the future; you cannot suppress multicast traffic only.

Separate thresholds cannot be provided for broadcast and/or multicast traffic. The threshold you configure for broadcast suppression applies to both the incoming multicast traffic and broadcast traffic.

To enable multicast suppression on a WS-X4516 supervisor engine, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode and enter the port to configure.
Step 3	`Switch(config-if)#` **`storm-control broadcast include multicast`**	Enable multicast suppression.
Step 4	`Switch(config-if)#` **`exit`**	Returns to configuration mode.
Step 5	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.

The following example shows how to enable multicast suppression on ports that have broadcast suppression already enabled:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# storm-control broadcast include multicast
Switch(config)# end
Switch#
```

Multicast Suppression on the WS-X4515, WS-X4014, and WS-X4013+ Supervisor Engines

Hardware does not provide support for multicast suppression on the WS-X4515, WS-X4014, and WS-X4013+ supervisor engines. One consequence of using software-based broadcast suppression on these modules is that all incoming data packets are dropped. Irrespective of your selecting to configure broadcast suppression only, multicast packets are filtered as well on stub and blocking gigabit ports. The non blocking gigabit ports that do provide broadcast suppression in hardware also do not filter multicast packets.

Disabling Broadcast Storm Control

To disable storm control, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode and enter the port to configure.
Step 3	`Switch(config-if)# no storm-control broadcast level`	Disables port storm control.
Step 4	`Switch(config-if)# no storm-control action {shutdown | trap}`	Disables the specified storm control action and returns to default filter action.
Step 5	`Switch(config-if)# exit`	Returns to configuration mode.
Step 6	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 7	`Switch# show storm-control broadcast`	Verifies your entries.
Step 8	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

The following example shows how to disable storm control on interface.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# int fa3/1
Switch(config-if)# no storm-control broadcast level
Switch(config-if)# end
Switch# show storm-control //Supervisor Engine 2+ to V-10GE
Interface  Filter State   Upper    Lower    Current
---------  -------------  -------  -------  -------
Switch#
```

```
Switch# show storm-control //Supervisor Engine 6-E
Interface Filter State Broadcast Multicast Level
--------- ------------- --------- --------- -----
Switch#
```

Disabling Multicast Storm Control

To disable multicast suppression on WS-X4516, WS-X4515, WS-X4014, and WS-X4013+ supervisor engines, perform the following task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode and enter the port to configure.
Step 3	`Switch(config-if)# [no] storm-control broadcast include multicast`	Enables multicast suppression.
Step 4	`Switch(config-if)# end`	Returns to configuration mode.
Step 5	`Switch(config)# end`	Returns to privileged EXEC mode.

To disable multicast suppression on the Supervisor Engine 6-E, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# [no] storm-control broadcast include multicast`	Enables/disables multicast suppression.
Step 3	`Switch(config-if)# no storm-control broadcast level`	Disables port storm control (broadcast and multicast).
Step 4	`Switch(config-if)# end`	Returns to configuration mode.
Step 5	`Switch(config)# end`	Returns to privileged EXEC mode.

Displaying Storm Control

Note Use the **show interface capabilities** command to determine the mode in which storm control is supported on an interface.

The following example shows an interface that supports broadcast suppression in software (sw).

```
Switch# show int fa2/1 capabilities
FastEthernet2/1
  Model:                WS-X4148-RJ45V-RJ-45
  Type:                 10/100BaseTX
  Speed:                10,100,auto
  Duplex:               half,full,auto
  Auto-MDIX:            no
  Trunk encap. type:    802.1Q
  Trunk mode:           on,off,desirable,nonegotiate
  Channel:                yes
  Broadcast suppression: percentage(0-100), hw
  Multicast suppression: percentage(0-100), hw <=====unique to Sup Engine 6-E
  Flowcontrol:            rx-(none),tx-(none)
  VLAN Membership:        static, dynamic
  Fast Start:             yes
  CoS rewrite:            yes
  ToS rewrite:            yes
  Inline power:           yes (Cisco Voice Protocol)
  SPAN:                   source/destination
  UDLD:                   yes
  Link Debounce:          no
  Link Debounce Time:     no
  Port Security:          yes
  Dot1x:                  yes
  Maximum MTU:            1552 bytes (Baby Giants)
  Multiple Media Types:  no
  Diagnostic Monitoring: N/A
```

Note Use the **show interfaces counters storm-control** command to display a count of discarded packets.

```
Switch# show interfaces counters storm-control
Port         Broadcast   Multicast           Level    TotalSuppressedPackets
Fa2/1          Enabled    Disabled          10.00%              46516510
Gi3/1          Enabled     Enabled          50.00%                     0
```

The following example shows the output of the **show storm-control** command:

```
Switch# show storm-control //Supervisor Engine 2+ to V-10GE
Interface  Filter State   Upper    Lower    Current
---------  -------------  -------  -------  -------
Gi4/4      Forwarding       2.00%    2.00%    N/A
Switch
```

Note In the previous example, “current” represents the percentage of traffic suppressed at a given instant, and the value is N/A for ports that perform suppression in hardware.

```
Switch# show storm-control //Supervisor Engine 6-E
Interface  Filter State   Broadcast Multicast Level
---------  -------------  --------- --------- -----
Fa2/1      Blocking       Enabled   Disabled   10.00%
Gi3/1      Link Down      Enabled   Enabled    50.00%
```

CHAPTER 43

Configuring SPAN and RSPAN

This chapter describes how to configure the Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) on the Catalyst 4500 series switches. SPAN selects network traffic for analysis by a network analyzer, such as a SwitchProbe device or other Remote Monitoring (RMON) probe.

This chapter consists of the following sections:

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cr/index.htm.

Overview of SPAN and RSPAN

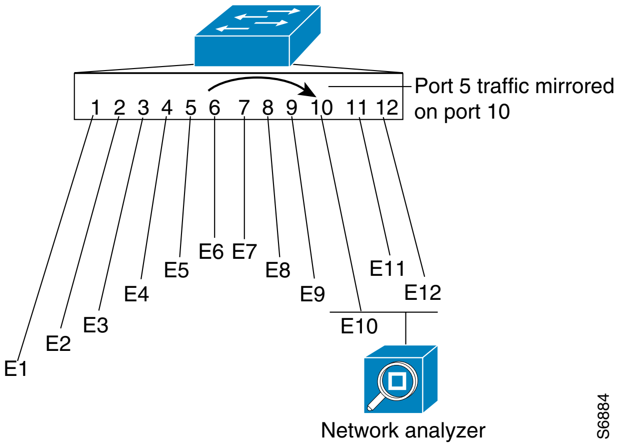
This sections includes the following subsections:

SPAN mirrors traffic from one or more source interfaces on any VLAN or from one or more VLANs to a destination interface for analysis. In Figure 43-1, all traffic on Ethernet interface 5 (the source interface) is mirrored to Ethernet interface 10. A network analyzer on Ethernet interface 10 receives all network traffic from Ethernet interface 5 without being physically attached to it.

For SPAN configuration, the source interfaces and the destination interface must be on the same switch.

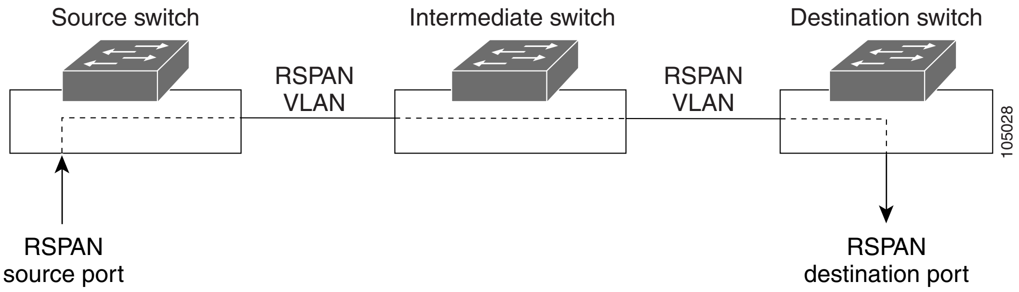
SPAN does not affect the switching of network traffic on source interfaces; copies of the packets received or transmitted by the source interfaces are sent to the destination interface.

Figure 43-1 Example SPAN Configuration



RSPAN extends SPAN by enabling remote monitoring of multiple switches across your network. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches. The SPAN traffic from the sources is copied onto the RSPAN VLAN and then forwarded over trunk ports that are carrying the RSPAN VLAN to any RSPAN destination sessions monitoring the RSPAN VLAN, as shown in Figure 43-2.

Figure 43-2 Example of RSPAN Configuration



SPAN and RSPAN do not affect the switching of network traffic on source ports or source VLANs; a copy of the packets received or sent by the sources is sent to the destination. Except for traffic that is required for the SPAN or RSPAN session, by default, destination ports do not receive or forward traffic.

You can use the SPAN or RSPAN destination port to forward transmitted traffic from a network security device. For example, if you connect a Cisco Intrusion Detection System (IDS) sensor appliance to a destination port, the IDS device can send TCP reset packets to close down the TCP session of a suspected attacker.

SPAN and RSPAN Concepts and Terminology

This section describes concepts and terminology associated with SPAN and RSPAN configuration and includes the following subsections:

- SPAN Session, page 43-3
- Traffic Types, page 43-3
- Source Port, page 43-4
- Destination Port, page 43-5
- VLAN-Based SPAN, page 43-5
- SPAN Traffic, page 43-6

SPAN Session

A local SPAN session associates a destination port with source ports. You can monitor incoming or outgoing traffic on a series or range of ports and source VLANs. An RSPAN session associates source ports and source VLANs across your network with an RSPAN VLAN. The destination source is the RSPAN VLAN.

You configure SPAN sessions by using parameters that specify the source of network traffic to monitor.

You can configure multiple SPAN or RSPAN sessions with separate or overlapping sets of SPAN sources. Both switched and routed ports can be configured as SPAN sources or destination ports.

An RSPAN source session associates SPAN source ports or VLANs with a destination RSPAN VLAN. An RSPAN destination session associates an RSPAN VLAN with a destination port.

SPAN sessions do not interfere with the normal operation of the switch; however, an oversubscribed SPAN destination (for example, a 10-Mbps port monitoring a 100-Mbps port) results in dropped or lost packets.

You can configure SPAN sessions on disabled ports; however, a SPAN session does not become active unless you enable the destination port and at least one source port or VLAN for that session.

A SPAN session remains inactive after system startup until the destination port is operational.

Traffic Types

SPAN sessions include these traffic types:

- Receive (Rx) SPAN—The goal of receive (or ingress) SPAN is to monitor as much as possible all packets received by the source interface or VLAN before any modification or processing is performed by the switch. A copy of each packet received by the source is sent to the destination port for that SPAN session. You can monitor a series or range of ingress ports or VLANs in a SPAN session.

  On tagged packets (Inter-Switch Link [ISL] or IEEE 802.1Q), the tagging is removed at the ingress port. At the destination port, if tagging is enabled, the packets appear with the ISL or 802.1Q headers. If no tagging is specified, packets appear in the native format.

  Packets that are modified because of routing are copied without modification for Rx SPAN; that is, the original packet is copied. Packets that are modified because of quality of service (QoS)—for example, modified Differentiated Services Code Point (DSCP)—are copied with modification for Rx SPAN.

Some features that can cause a packet to be dropped during receive processing have no effect on SPAN; the destination port receives a copy of the packet even if the actual incoming packet is dropped. These features include IP standard and extended input access control lists (ACLs), IP standard and extended output ACLs for unicast and ingress QoS policing, VLAN maps, ingress QoS policing, and policy-based routing. Switch congestion that causes packets to be dropped also has no effect on SPAN.

- Transmit (Tx) SPAN—The goal of transmit (or egress) SPAN is to monitor as much as possible all packets sent by the source interface after the switch performs all modification and processing. After the packet is modified, the source sends a copy of each packet to the destination port for that SPAN session. You can monitor a range of egress ports in a SPAN session.

  Packets that are modified because of routing—for example, with a time-to-live (TTL) or MAC-address modification—are duplicated at the destination port. On packets that are modified because of QoS, the modified packet might not have the same DSCP (IP packet) or CoS (non-IP packet) as the SPAN source.

  Some features that can cause a packet to be dropped during transmit processing might also affect the duplicated copy for SPAN. These features include VLAN maps, IP standard and extended output ACLs on multicast packets, and egress QoS policing. In the case of output ACLs, if the SPAN source drops the packet, the SPAN destination would also drop the packet. In the case of egress QoS policing, if the SPAN source drops the packet, the SPAN destination might not drop it. If the source port is oversubscribed, the destination ports have different dropping behavior.

- Both—In a SPAN session, you can monitor a single port series or a range of ports for both received and sent packets.

Source Port

A source port (also called a monitored port) is a switched or routed port that you monitor for network traffic analysis. In a single local SPAN session or RSPAN source session, you can monitor source port traffic, such as received (Rx), transmitted (Tx), or bidirectional (both). The switch supports any number of source ports (up to the maximum number of available ports on the switch) and any number of source VLANs.

A source port has these characteristics:

- It can be any port type (for example, EtherChannel, Fast Ethernet, Gigabit Ethernet, and so forth).
- It can be monitored in multiple SPAN sessions.
- It cannot be a destination port.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor. For EtherChannel sources, the monitored direction would apply to all physical ports in the group.
- Source ports can be in the same or different VLANs.
- For VLAN SPAN sources, all active ports in the source VLAN are included as source ports.

You can configure a trunk port as a source port. By default, all VLANs active on the trunk are monitored. You can limit SPAN traffic monitoring on trunk source ports to specific VLANs by using VLAN filtering. Only switched traffic in the selected VLANs is sent to the destination port. This feature affects only traffic forwarded to the destination SPAN port and does not affect the switching of normal traffic. This feature is not allowed in sessions with VLAN sources.

Destination Port

Each local SPAN session or RSPAN destination session must have a destination port (also called a monitoring port) that receives a copy of traffic from the source ports and VLANs.

A destination port has these characteristics:

- A destination port must reside on the same switch as the source port (for a local SPAN session).
- A destination port can be any Ethernet physical port.
- A destination port can participate in only one SPAN session at a time. (A destination port in one SPAN session cannot be a destination port for a second SPAN session.)
- A destination port cannot be a source port.
- A destination port cannot be an EtherChannel group.
- A destination port can be a physical port that is assigned to an EtherChannel group, even if the EtherChannel group has been specified as a SPAN source. The port is removed from the group while it is configured as a SPAN destination port.
- The port does not transmit any traffic except that traffic required for the SPAN session unless learning is enabled. If learning is enabled, the port also transmits traffic directed to hosts that have been learned on the destination port.
- If ingress traffic forwarding is enabled for a network security device, the destination port forwards traffic at Layer 2.
- A destination port does not participate in spanning tree while the SPAN session is active.
- When it is a destination port, it does not participate in any of the Layer 2 protocols (STP, VTP, CDP, DTP, PagP).
- A destination port that belongs to a source VLAN of any SPAN session is excluded from the source list and is not monitored.
- A destination port receives copies of sent and received traffic for all monitored source ports. If a destination port is oversubscribed, it could become congested. This congestion could affect traffic forwarding on one or more of the source ports.

VLAN-Based SPAN

VLAN-based SPAN (VSPAN) is the monitoring of the network traffic in one or more VLANs.

Use these guidelines for VSPAN sessions:

- Traffic on RSPAN VLANs is not monitored by VLAN-based SPAN sessions.
- Only traffic on the monitored VLAN is sent to the destination port.
- If a destination port belongs to a source VLAN, it is excluded from the source list and is not monitored.
- If ports are added to or removed from the source VLANs, the traffic on the source VLAN received by those ports is added to or removed from the sources being monitored.
- VLAN pruning and the VLAN allowed list have no effect on SPAN monitoring.
- VSPAN monitors only traffic that enters the switch, not traffic that is routed between VLANs. For example, if a VLAN is being Rx-monitored, and the multilayer switch routes traffic from another VLAN to the monitored VLAN, that traffic is not monitored and is not received on the SPAN destination port.

- You cannot use filter VLANs in the same session with VLAN sources.
- You can monitor only Ethernet VLANs.

SPAN Traffic

You can use local SPAN to monitor all network traffic, including multicast and bridge protocol data unit (BPDU) packets, Cisco Discovery Protocol (CDP), VLAN Trunk Protocol (VTP), Dynamic Trunking Protocol (DTP), Spanning Tree Protocol (STP), and Port Aggregation Protocol (PAgP) packets. You cannot use RSPAN to monitor Layer 2 protocols. (See the "RSPAN Configuration Guidelines" section on page 43-16 for more information.)

In some SPAN configurations, multiple copies of the same source packet are sent to the SPAN destination port. For example, a bidirectional (both Rx and Tx) SPAN session is configured for the sources a1 Rx monitor and the a2 Rx and Tx monitor to destination port d1. If a packet enters the switch through a1 and is switched to a2, both incoming and outgoing packets are sent to destination port d1. Both packets are the same (unless a Layer-3 rewrite occurs, in which case the packets are different because of the added Layer 3 information).

SPAN and RSPAN Session Limits

You can configure up to two simultaneous SPAN sessions containing ingress sources and up to four simultaneous SPAN sessions containing egress sources. Bidirectional sources count as both ingress and egress. RSPAN destination sessions count as a session containing an ingress source.

Default SPAN and RSPAN Configuration

Table 43-1 shows the default SPAN and RSPAN configuration.

Table 43-1 Default SPAN and RSPAN Configuration

Feature	Default Setting
SPAN state	Disabled.
Source port traffic to monitor	Both received and sent traffic (**both**).
Filters	All VLANs, all packet types, all address types.
Encapsulation type (destination port)	Native form (no encapsulation type header).
Ingress forwarding (destination port)	Disabled.
Host learning (destination port)	Disabled.

Configuring SPAN

The following sections describe how to configure SPAN:

- SPAN Configuration Guidelines and Restrictions, page 43-7
- Configuring SPAN Sources, page 43-8
- Configuring SPAN Destinations, page 43-9
- Monitoring Source VLANs on a Trunk Interface, page 43-9

- Configuration Scenario, page 43-10
- Verifying a SPAN Configuration, page 43-10

Note Entering SPAN configuration commands does not clear previously configured SPAN parameters. You must enter the **no monitor session** command to clear configured SPAN parameters.

SPAN Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring SPAN:

- You must use a network analyzer to monitor interfaces.
- You cannot mix source VLANs and filter VLANs within a SPAN session. You can have source VLANs or filter VLANs, but not both at the same time.
- EtherChannel interfaces can be SPAN source interfaces; they cannot be SPAN destination interfaces.
- When you specify source interfaces and do not specify a traffic type (Tx, Rx, or both), "both" is used by default.
- If you specify multiple SPAN source interfaces, the interfaces can belong to different VLANs.
- You must enter the **no monitor session** *number* command with no other parameters to clear the SPAN session *number*.
- The **no monitor** command clears all SPAN sessions.
- SPAN destinations never participate in any spanning tree instance. SPAN includes BPDUs in the monitored traffic, so any BPDUs seen on the SPAN destination are from the SPAN source.
- SPAN is limited to one destination port per session.

Configuring SPAN Sources

To configure the source for a SPAN session, perform this task:

Command	Purpose
Switch(config)# [**no**] **monitor session** {*session_number*} {**source** {**interface** <*interface_list*> | {**vlan** *vlan_IDs* | **cpu** [**queue** *queue_ids*] } [**rx** | **tx** | **both**]	Specifies the SPAN session number (1 through 6), the source interfaces (FastEthernet or GigabitEthernet), VLANs (1 through 4094), whether or not traffic received or sent from the CPU is copied to the session destination, and the traffic direction to be monitored. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *interface-list*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). For *vlan_IDs*, specifies the source VLAN. For *queue_ids*, specifies the queue(s) involved. (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports. • **Rx**—Monitor received traffic. • **Tx**—Monitor transmitted traffic. • **both**—Monitor both received and transmitted traffic (bidirectional). Queues may be identified either by number or by name. Queue names may subsume multiple numbered queues for convenience. Use the **no** keyword to restore the defaults.

This example shows how to configure SPAN session 1 to monitor bidirectional traffic from source interface Fast Ethernet 5/1:

```
Switch(config)# monitor session 1 source interface fastethernet 5/1
```

This example shows how to configure sources with differing directions within a SPAN session:

```
Switch(config)# monitor session 1 source interface fa2/3 rx
Switch(config)# monitor session 1 source interface fa2/2 tx
Switch(config)#
```

Configuring SPAN Destinations

To configure the destination for a SPAN session, perform this task:

Command	Purpose
`Switch(config)# [no] monitor session <session_number> destination interface <interface> [encapsulation {isl | dot1q}] [ingress [vlan vlan_IDs] [learning}]`	Specifies the SPAN session number (1 through 6) and the destination interfaces or VLANs. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *interface*, specifies the destination interface. For *vlan_IDs*, specifies the destination VLAN. Use the **no** keyword to restore the defaults.

Note SPAN is limited to one destination port per session.

This example shows how to configure interface Fast Ethernet 5/48 as the destination for SPAN session 1:

```
Switch(config)# monitor session 1 destination interface fastethernet 5/48
```

Monitoring Source VLANs on a Trunk Interface

To monitor specific VLANs when the SPAN source is a trunk interface, perform this task:

Command	Purpose
`Switch(config)# [no] monitor session {session_number} filter {vlan vlan_IDs [, | - ]} | {packet-type {good | bad}} | {address-type {unicast | multicast | broadcast} [rx | tx | both]}`	Monitors specific VLANs when the SPAN source is a trunk interface. The filter keyword restricts monitoring to traffic that is on the specified VLANs; it is typically used when monitoring a trunk interface. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *vlan_IDs*, specifies the VLAN. Monitoring is established through all the ports in the specified VLANs Use the **no** keyword to restore the defaults.

This example shows how to monitor VLANs 1 through 5 and VLAN 9 when the SPAN source is a trunk interface:

```
Switch(config)# monitor session 2 filter vlan 1 - 5 , 9
```

Configuration Scenario

This example shows how to use the commands described in this chapter to completely configure and unconfigure a span session. Assume that you want to monitor bidirectional traffic from source interface Fast Ethernet 4/10, which is configured as a trunk interface carrying VLANs 1 through 4094. Moreover, you want to monitor only traffic in VLAN 57 on that trunk. Using Fast Ethernet 4/15 as your destination interface, you would enter the following commands:

```
Switch(config)# monitor session 1 source interface fastethernet 4/10
Switch(config)# monitor session 1 filter vlan 57
Switch(config)# monitor session 1 destination interface fastethernet 4/15
```

You are now monitoring traffic from interface Fast Ethernet 4/10 that is on VLAN 57 out of interface FastEthernet 4/15. To disable the span session enter the following command:

```
Switch(config)# no monitor session 1
```

Verifying a SPAN Configuration

This example shows how to verify the configuration of SPAN session 2:

```
Switch# show monitor session 2
Session 2
---------
Source Ports:
    RX Only:       Fa5/12
    TX Only:       None
    Both:          None
Source VLANs:
    RX Only:       None
    TX Only:       None
    Both:          None
Destination Ports: Fa5/45
Filter VLANs:      1-5,9
Switch#
```

CPU Port Sniffing

When configuring a SPAN session, you can specify the CPU (or a subset of CPU queues) as a SPAN source. Queues may be specified either by number or by name. When such a source is specified, traffic going to the CPU through one of the specified queues is mirrored and sent out of the SPAN destination port in the session. This traffic includes both control packets and regular data packets that are sent to or from the CPU (due to software forwarding).

You can mix the CPU source with either regular port sources or VLAN sources.

To configure CPU source sniffing, perform this task:

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td>Switch(config)# [<b>no</b>] <b>monitor session</b> {<i>session_number</i>} {<b>source</b> {<b>interface</b> <i>interface_list</i> | {<b>vlan</b> <i>vlan_IDs</i> | <b>cpu</b> [<b>queue</b> <i>queue_ids</i>] } [<b>rx</b> | <b>tx</b> | <b>both</b>]</td><td>Specifies that the CPU causes traffic received by or sent from the CPU to be copied to the destination of the session. The queue identifier optionally allows sniffing-only traffic (received) on the specified CPU queue(s).
For <i>session_number</i>, specifies the session number identified with this SPAN session (1 through 6).
For <i>interface-list</i>, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (<b>port-channel</b> <i>port-channel-number</i>).
For <i>vlan_IDs</i>, specifies the source VLAN.
For <i>queue_ids</i>, specifies the queue(s) involved.
(Optional) [<b>,</b> | <b>-</b>] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen.
(Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports.
• <b>Rx</b>—Monitor received traffic.
• <b>Tx</b>—Monitor transmitted traffic.
• <b>both</b>—Monitor both received and transmitted traffic (bidirectional).
Queues may be identified either by number or by name. Queue names may subsume multiple numbered queues for convenience.
Use the <b>no</b> keyword to restore the defaults.</td></tr>
</table>

This example shows how to configure a CPU source to sniff all packets received by the CPU:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# monitor session 1 source cpu rx
```

This example shows how to use queue names and queue number ranges for the CPU as a SPAN source on Supervisor Engines 2+ to V 10-GE:

```
Switch(config)# monitor session 2 source cpu queue control-packet rx
Switch(config)# monitor session 3 source cpu queue 21 -23 rx
```

This example shows how to use queue names and queue number ranges for the CPU as a SPAN source on Supervisor Engine 6-E:

```
Switch(config)# monitor session 2 source cpu queue control-packet rx
Switch(config)# monitor session 3 source cpu queue 10 rx
```

Note For Supervisor Engine 6-E, *control-packet* is mapped to queue 10.

Encapsulation Configuration

When configuring a SPAN destination port, you can explicitly specify the encapsulation type used by the port. Packets sent out the port are tagged in accordance with the specified mode. (The encapsulation mode also controls how tagged packets are handled when the ingress packet option is enabled.) The Catalyst 4500 series switch supervisor engines support ISL encapsulation and 802.1q encapsulation, as well as untagged packets.

Note Supervisor Engine 6-E supports *only* 802.1q encapsulation.

The “replicate” encapsulation type (in which packets are transmitted from the destination port using whatever encapsulation applied to the original packet) is not supported. If no encapsulation mode is specified, the port default is untagged. To view the task of configuring encapsulation, see the command table below.

Ingress Packets

When ingress is enabled, the SPAN destination port accepts incoming packets (potentially tagged depending on the specified encapsulation mode) and switches them normally. When configuring a SPAN destination port, you can specify whether or not the ingress feature is enabled and what VLAN to use to switch untagged ingress packets. (Specifying an ingress VLAN is not required when ISL encapsulation is configured, as all ISL encapsulated packets have VLAN tags.) Although the port is STP forwarding, it does not participate in the STP, so use caution when configuring this feature lest a spanning-tree loop be introduced in the network. When both ingress and a trunk encapsulation are specified on a SPAN destination port, the port goes forwarding in all active VLANs. Configuring a non-existent VLAN as an ingress VLAN is not allowed.

By default, host learning is disabled on SPAN destination ports with ingress enabled. The port is also removed from VLAN floodsets, so regular traffic is not switched out of the destination port. If learning is enabled, however, then traffic for hosts learned on the destination port is switched out the destination port. Host connected to SPAN destination port will not receive broadcast ARP request thus will not respond. It is also possible to configure static host entries (including a static ARP entry and a static entry in the MAC-address table) on SPAN destination ports.

Note This configuration does not work if the SPAN session does not have a source configured; the session is half configured with only the SPAN destination port.

To configure ingress packets and encapsulation, perform this task:

Command	Purpose
`Switch(config)# [no] monitor session <session_number> destination interface <interface> [encapsulation {isl | dot1q}] [ingress [vlan vlan_IDs] [learning]]`	Specifies the configuration of the ingress packet and the encapsulation type of the destination port. **Note** The **isl** keyword is *not* supported on the Supervisor Engine 6-E. For *session_number*, specifies the session number identified with this SPAN session (1 through 6). For *interface*, specifies the destination interface. For *vlan_IDs*, specifies the destination VLAN. Use the **no** keyword to restore the defaults.

This example shows how to configure a destination port with 802.1q encapsulation and ingress packets using native VLAN 7:

```
Switch(config)# monitor session 1 destination interface fastethernet 5/48
encapsulation dot1q ingress vlan 7
```

With this configuration, traffic from SPAN sources associated with session 1 would be copied out of interface Fast Ethernet 5/48, with 802.1q encapsulation. Incoming traffic would be accepted and switched, with untagged packets being classified into VLAN 7.

Access List Filtering

When configuring a SPAN session, you can apply access list filtering. Access list filtering applies to all packets passing through a SPAN destination port that might be sniffed in the egress or ingress direction. Access list filters are allowed on local SPAN sessions only. If the SPAN destination is an RSPAN VLAN, the access list filter is rejected.

Note Access list filtering is available in Cisco IOS Release 12.2(20)EW and later releases.

ACL Configuration Guidelines

You can configure ACLs on a SPAN session. Use these guidelines for ACL/SPAN sessions:

- If an ACL is associated with a SPAN session, the rules associated with that ACL are applied against all packets exiting the SPAN destination interface. Rules pertaining to other VACLs or RACLs previously associated with the SPAN destination interface are not applied.
- Only one ACL can be associated with a SPAN session.

- When no ACLs are applied to packets exiting a SPAN destination interface, all traffic is permitted regardless of the PACLs, VACLs, or RACLs that have been previously applied to the destination interface or VLAN to which the SPAN destination interface belongs.
- If an ACL is removed from a SPAN session, all traffic is permitted once again.
- If SPAN configuration is removed from the SPAN session, all rules associated with the SPAN destination interface are applied once again.
- If a SPAN destination port is configured as a trunk port and the VLANs to which it belongs have ACLs associated with them, the traffic is not subjected to the VACLs.
- ACL configuration applies normally to the RSPAN VLAN and to trunk ports carrying the RSPAN VLAN. This configuration enables the user to apply VACLs on RSPAN VLANs. If a user attempts to configure an ACL on a SPAN session with the destination port as an RSPAN VLAN, the configuration is rejected.
- If CAM resources are exhausted and packets are passed to the CPU for lookup, any output port ACLs associated with a SPAN session are not applied.
- If a named IP ACL is configured on a SPAN session before an ACL is created, the configuration is accepted, and the software creates an empty ACL with no ACEs. (An empty ACL permits all packets.) Subsequently, the rules can be added to the ACL.
- The ACLs associated with a SPAN session are applied on the destination interface on output.
- No policing is allowed on traffic exiting SPAN ports.
- Only IP ACLs are supported on SPAN sessions.

Configuring Access List Filtering

To configure access list filtering, perform this task:

Command	Purpose
`Switch(config)# [`**`no`**`]` **`monitor session`** `{`*`session_number`*`}` **`filter`** `{`**`ip access-group`** `[`*`name`* `|` *`id`*`] }{`**`vlan`** *`vlan_IDs`* `[`**`,`** `|` **`-`** `] } | {`**`packet-type`** `{`**`good`** `|` **`bad`**`}} | {`**`address-type {unicast`** `|` **`multicast`** `|` **`broadcast`**`}` **`[rx`** `|` **`tx`** `|` **`both`**`]}`	Specifies filter sniffing based on the access list. For *session_number*, specify the session number identified with this SPAN session (1 through 6). You can specify either a name or a numeric ID for the access list. For *name*, specify the IP access list name. For *id*, specify a standard <1 to 199> or extended <1300-2699> IP access list.

Note IP access lists must be created in configuration mode as described in the chapter "Configuring Network Security with ACLs."

This example shows how to configure IP access group 10 on a SPAN session and verify that an access list has been configured:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# monitor session 1 source interface fa6/1 both
Switch(config)# monitor session 1 destination interface fa6/2
Switch(config)# monitor session 1 filter vlan 1
Switch(config)# monitor session 1 filter ip access-group 10
Switch(config)# exit
Switch# show monitor

Session 1
---------
Type              : Local Session
Source Ports      :
    Both          : Fa6/1
Destination Ports : Fa6/2
    Encapsulation : Native
          Ingress : Disabled
         Learning : Disabled
Filter VLANs      : 1
IP Access-group   : 10
```

Packet Type Filtering

When configuring a SPAN session, you can specify packet filter parameters similar to VLAN filters. When specified, the packet filters indicate types of packets that may be sniffed. If no packet filters are specified, packets of all types may be sniffed. Different types of packet filters may be specified for ingress and egress traffic.

There are two categories of packet filtering: packet-based (good, error) or address-based (unicast/multicast/broadcast). Packet-based filters can only be applied in the ingress direction. Packets are classified as broadcast, multicast, or unicast by the hardware based on the destination address.

Note When filters of both types are configured, only packets that pass both filters are spanned. For example, if you set both “error” and “multicast,” only multicast packets with errors are spanned.

To configure packet type filtering, perform this task:

Command	Purpose
`Switch(config)# [`**`no`**`] `**`monitor session`**` {`*`session_number`*`} `**`filter`**` {`**`vlan`**` `*`vlan_IDs`*` [`**`,`**` | `**`-`**` ] } | {`**`packet-type`**` {`**`good`**` | `**`bad`**`}} | {`**`address-type {unicast`**` | `**`multicast`**` | `**`broadcast`**`} `**`[rx`**` | `**`tx`**` | `**`both`**`]}`	Specifies filter sniffing of the specified packet types in the specified directions. For *session_number*, specifies the session number identified with this SPAN session (1 through 6). For *vlan_IDs*, specifies the VLAN. You can specify both Rx and Tx type filters, as well as specify multiple type filters at the same time (such as **good** and **unicast** to only sniff non-error unicast frames). As with VLAN filters, if no type or filter is specified, then the session sniffs all packet types. Use the **no** keyword to restore the defaults.

This example shows how to configure a session to accept only unicast packets in the ingress direction:

```
Switch(config)# monitor session 1 filter address-type unicast rx
```

Configuration Example

The following is an example of SPAN configuration using some of the SPAN enhancements.

In the example below, you configure a session to sniff unicast traffic arriving on interface Gi1/1. The traffic is mirrored out of interface Gi1/2 with ISL encapsulation. Ingress traffic is permitted.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# monitor session 1 source interface gi1/1 rx
Switch(config)# monitor session 1 destination interface gi1/2 encapsulation isl ingress
Switch(config)# monitor session 1 filter address-type unicast rx
Switch(config)# exit
Switch# show monitor

Session 1
---------
Type              : Local Session
Source Ports      :
    RX Only       : Gi1/1
Destination Ports : Gi1/2
    Encapsulation : ISL
          Ingress : Enabled
         Learning : Disabled
Filter Addr Type  :
    RX Only       : Unicast
```

Configuring RSPAN

Note This feature is not supported on Supervisor Engine 6-E.

This section describes how to configure RSPAN on your switch and it contains this configuration information:

- RSPAN Configuration Guidelines, page 43-16
- Creating an RSPAN Session, page 43-17
- Creating an RSPAN Destination Session, page 43-19
- Creating an RSPAN Destination Session and Enabling Ingress Traffic, page 43-20
- Removing Ports from an RSPAN Session, page 43-21
- Specifying VLANs to Monitor, page 43-22
- Specifying VLANs to Filter, page 43-23

RSPAN Configuration Guidelines

Follow these guidelines when configuring RSPAN:

Note Since RSPAN VLANs have special properties, you should reserve a few VLANs across your network for use as RSPAN VLANs; do not assign access ports to these VLANs.

Note You can apply an output access control list (ACL) to RSPAN traffic to selectively filter or monitor specific packets. Specify these ACLs on the RSPAN VLAN in the RSPAN source switches.

- RSPAN sessions can coexist with SPAN sessions within the limits described in the "SPAN and RSPAN Session Limits" section on page 43-6.
- For RSPAN configuration, you can distribute the source ports and the destination ports across multiple switches in your network.
- RSPAN does not support BPDU packet monitoring or other Layer 2 switch protocols.
- The RSPAN VLAN is configured only on trunk ports and not on access ports. To avoid unwanted traffic in RSPAN VLANs, make sure that all participating switches support the VLAN remote-span feature. Access ports on the RSPAN VLAN are silently disabled.
- You should create an RSPAN VLAN before configuring an RSPAN source or destination session.
- 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 for VLAN-IDs that are lower than 1005.
- Because RSPAN traffic is carried across a network on an RSPAN VLAN, the original VLAN association of the mirrored packets is lost. Therefore, RSPAN can only support forwarding of traffic from an IDS device onto a single user-specified VLAN.

Creating an RSPAN Session

First create an RSPAN VLAN that does not exist for the RSPAN session in any of the switches that participate in RSPAN. With VTP enabled in the network, you can create the RSPAN VLAN in one switch, and then VTP propagates it to the other switches in the VTP domain for VLAN-IDs that are lower than 1005.

Use VTP pruning to get efficient flow of RSPAN traffic, or manually delete the RSPAN VLAN from all trunks that do not need to carry the RSPAN traffic.

To start an RSPAN source session and to specify the source (monitored) ports and the destination RSPAN VLAN, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`no monitor session`** `{`*`session_number`* `|` **`all`** `|` **`local`** `|` **`remote`**`}`	Clears any existing RSPAN configuration for the session. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). Specifies **all** to remove all RSPAN sessions, **local** to remove all local sessions, or **remote** to remove all remote SPAN sessions.
Step 3	`Switch(config)#` **`vlan`** `{remote_vlan_ID}`	Specifies a remote VLAN ID. Ensure that the VLAN ID is not being used for any user traffic.
Step 4	`Switch(config-vlan)#` **`remote-span`**	Converts the VLAN ID to a remote VLAN ID.
Step 5	`Switch(config-vlan)#` **`exit`**	Returns to global configuration mode.

	Command	Purpose
Step 6	Switch(config)# [**no**] **monitor session** {*session_number*} {**source** {**interface** <*interface_list*> | {**vlan** *vlan_IDs* | **cpu** [**queue** *queue_ids*]} [**rx** | **tx** | **both**]	Specifies the RSPAN session and the source port (monitored port). For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *interface-list*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). For *vlan-IDs*, specifies the source VLAN or VLANs to monitor. Valid VLANs are in the range from 1 to 4094. For *queue_ids*, specifies either a set of CPU queue numerical identifiers from 1 to 32, or a named queue. (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports. • **Rx**—Monitor received traffic. • **Tx**—Monitor transmitted traffic. • **both**—Monitor both received and transmitted traffic (bidirectional).
Step 7	Switch(config)# **monitor session** *session_number* **destination remote vlan** *vlan-ID*	Specifies the RSPAN session and the destination remote VLAN. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *vlan-ID*, specifies the RSPAN VLAN to carry the monitored traffic to the destination port.
Step 8	Switch(config)# **end**	Returns to privileged EXEC mode.
Step 9	Switch# **show monitor** [**session** *session_number*]	Verifies your entries.
Step 10	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

This example shows how to clear any existing RSPAN configuration for session 1, configure RSPAN session 1 to monitor multiple source interfaces, and configure the destination RSPAN VLAN.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# no monitor session 1
Switch(config)# monitor session 1 source interface fastEthernet3/10 tx
Switch(config)# monitor session 1 source interface fastEthernet3/2 rx
Switch(config)# monitor session 1 source interface fastEthernet3/3 rx
Switch(config)# monitor session 1 source interface port-channel 102 rx
Switch(config)# monitor session 1 destination remote vlan 901
Switch(config)# end
```

Creating an RSPAN Destination Session

To create an RSPAN destination session and to specify the source RSPAN VLAN and the destination port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# monitor session session_number source remote vlan vlan-ID`	Specifies the RSPAN session and the source RSPAN VLAN. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *vlan-ID*, specifies the source RSPAN VLAN to monitor.
Step 3	`Switch(config)# [no] monitor session <session_number> destination interface <interface> [encapsulation {isl	dot1q}] [ingress [vlan vlan_IDs] [learning]]`	Specifies the RSPAN session and the destination interface. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *interface*, specifies the destination interface. For *vlan_IDs*, specifies the ingress VLAN, if necessary. (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic. Only received (**rx**) traffic can be monitored on additional source ports. • **isl**—Use ISL encapsulation. • **dot1q**—Use 802.1Q encapsulation.
Step 4	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show monitor [session session_number]`	Verifies your entries.
Step 6	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to configure VLAN 901 as the source remote VLAN and port 5 as the destination interface:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# monitor session 1 source remote vlan 901
Switch(config)# monitor session 1 destination interface gigabitEthernet1/2
Switch(config)# end
```

Creating an RSPAN Destination Session and Enabling Ingress Traffic

To create an RSPAN destination session, to specify the source RSPAN VLAN, and to enable ingress traffic on the destination port for a network security device (such as a Cisco IDS [Intrusion Detection System] sensor appliance), perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`monitor session`** {*`session_number`*} **`source vlan`** *`vlan_IDs`*	Specifies the RSPAN session and the source RSPAN VLAN. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *vlan_IDs*, specifies the source VLAN or VLANs to monitor. Valid VLANs are in the range from 1 to 4094.
Step 3	`Switch(config)#` [**`monitor session`** *`session_number`* **`destination interface`** *`interface-id`* [**`encapsulation`** {**`dot1q`** [**`ingress vlan`** *`vlan id`*] | **`ISL`** [**`ingress`**]} | **`ingress vlan`** *`vlan id`*] [**`learning`**]]	Specifies the RSPAN session, the destination port, the packet encapsulation, and the ingress VLAN. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *interface-id*, specifies the destination port. Valid interfaces include physical interfaces. (Optional) Specifies the encapsulation of the packets transmitted on the RSPAN destination port. If no encapsulation is specified, all transmitted packets are sent in native format (untagged). • Enter **encapsulation dot1q** to send native VLAN packets untagged, and all other VLAN **tx** packets tagged **dot1q**. • Enter **encapsulation isl** to send all **tx** packets encapsulated using ISL. (Optional) Specifies whether forwarding is enabled for ingress traffic on the RSPAN destination port. • For native (untagged) and **dot1q** encapsulation, specify **ingress vlan** *vlan id* to enable ingress forwarding with *vlan id* as the native VLAN; *vlan id* is also used as the native VLAN for transmitted packets. • Specify **ingress** to enable ingress forwarding when using ISL encapsulation. • Specify **learning** to enable learning when ingress is enabled.
Step 4	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Switch#` **`show monitor`** [**`session`** *`session_number`*]	Verifies your entries.
Step 6	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

This example shows how to configure VLAN 901 as the source remote VLAN and how to configure the destination port for ingress traffic on VLAN 5 by using a security device that supports 802.1Q encapsulation:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# monitor session 1 source remote vlan 901
Switch(config)# monitor session 1 destination interface gigabitEthernet1/2 ingress vlan 5
Switch(config)# end
```

Removing Ports from an RSPAN Session

To remove a port as an RSPAN source for a session, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# [no] monitor session {session_number} {source {interface interface_list | {vlan vlan_IDs | cpu [queue queue_ids]} [rx | tx | both]`	Specifies the characteristics of the RSPAN source port (monitored port) to remove. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *interface-list*, specifies the source port to no longer monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). For *vlan_IDs*, specifies the source vlan or vlans to monitor. Valid VLANs are in the range from 1 to 4094. For *queue_ids*, specifies either a set of CPU queue numerical identifiers from 1 to 32, or a named queue. (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports. • **Rx**—Monitor received traffic. • **Tx**—Monitor transmitted traffic. • **both**—Monitor both received and transmitted traffic (bidirectional).
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show monitor [session session_number]`	Verifies your entries.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to remove port 1 as an RSPAN source for RSPAN session 1:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# no monitor session 1 source interface gigabitEthernet1/1
Switch(config)# end
```

This example shows how to disable received traffic monitoring on port 1, which was configured for bidirectional monitoring:

```
Switch(config)# no monitor session 1 source interface gigabitEthernet1/1 rx
```

The monitoring of traffic received on port 1 is disabled, but traffic transmitted from this port continues to be monitored.

Specifying VLANs to Monitor

VLAN monitoring is similar to port monitoring. To specify VLANs to monitor, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# **no monitor session** {*session_number* | **all** | **local** | **remote**}	Clears any existing SPAN configuration for the session. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). Specify **all** to remove all SPAN sessions, **local** to remove all local sessions, or **remote** to remove all remote SPAN sessions.
Step 3	Switch(config)# [**no**] **monitor session** {*session_number*} {**source** {**interface** *interface_list* | {**vlan** *vlan_IDs* | **cpu** [**queue** *queue_ids*]} [**rx** | **tx** | **both**]	Specifies the RSPAN session and the source VLANs (monitored VLANs). You can monitor only received (**rx**) traffic on VLANs. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *interface-list*, specifies the source port to no longer monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). For *vlan-IDs*, the range is 1 to 4094; do not enter leading zeros. For *queue_ids*, specifies the source queue. (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports. • **Rx**—Monitor received traffic. • **Tx**—Monitor transmitted traffic. • **both**—Monitor both received and transmitted traffic (bidirectional).
Step 4	Switch(config)# **monitor session** *session_number* **destination remote vlan** *vlan-id*	Specifies the RSPAN session, the destination remote VLAN. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *vlan-id*, specifies the RSPAN VLAN to carry the monitored traffic to the destination port.

	Command	Purpose
Step 5	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 6	`Switch#` **`show monitor`** `[`**`session`** *`session_number`*`]`	Verifies your entries.
Step 7	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To remove one or more source VLANs from the RSPAN session, use the **no monitor session** *session_number* **source vlan** *vlan-id* **rx** global configuration command.

This example shows how to clear any existing configuration on RSPAN session 2, configure RSPAN session 2 to monitor received traffic on all ports belonging to VLANs 1 through 3, and send it to destination remote VLAN 902. The configuration is then modified to also monitor received traffic on all ports belonging to VLAN 10.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source vlan 1 - 3 rx
Switch(config)# monitor session 2 destination remote vlan 902
Switch(config)# monitor session 2 source vlan 10 rx
Switch(config)# end
```

Specifying VLANs to Filter

To limit RSPAN source traffic to specific VLANs, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`no monitor session`** `{`*`session_number`* `|` **`all`** `|` **`local`** `|` **`remote`**`}`	Clears any existing SPAN configuration for the session. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). Specify **all** to remove all SPAN sessions, **local** to remove all local sessions, or **remote** to remove all remote SPAN sessions.

	Command	Purpose
Step 3	Switch(config)# [**no**] **monitor session** {*session_number*} {**source** {**interface** *interface_list* | {**vlan** *vlan_IDs* | **cpu** [**queue** *queue_ids*]} [**rx** | **tx** | **both**]	Specifies the characteristics of the source port (monitored port) and RSPAN session. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *interface-list*, specifies the source port to monitor. The interface specified must already be configured as a trunk port. For *vlan-IDs*, the range is 1 to 4094; do not enter leading zeros. For *queue_ids*, specifies the source queue. (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports. • **Rx**—Monitor received traffic. • **Tx**—Monitor transmitted traffic. • **both**—Monitor both received and transmitted traffic (bidirectional).
Step 4	Switch(config)# **monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**]	Limits the RSPAN source traffic to specific VLANs. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *vlan-id*, the range is 1 to 4094; do not enter leading zeros. (Optional) Use a comma (**,**) to specify a series of VLANs or use a hyphen (**-**) to specify a range of VLANs. Enter a space after the comma; enter a space before and after the hyphen.
Step 5	Switch(config)# **monitor session** *session_number* **destination remote vlan** *vlan-id*	Specifies the RSPAN session, the destination remote VLAN. For *session_number*, specifies the session number identified with this RSPAN session (1 through 6). For *vlan-id*, specifies the RSPAN VLAN to carry the monitored traffic to the destination port.
Step 6	Switch(config)# **end**	Returns to privileged EXEC mode.
Step 7	Switch# **show monitor** [**session** *session_number*]	Verifies your entries.
Step 8	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To monitor all VLANs on the trunk port, use the **no monitor session** *session_number* **filter vlan** global configuration command.

This example shows how to clear any existing configuration on RSPAN session 2, configure RSPAN session 2 to monitor traffic received on trunk port 4, and send traffic for only VLANs 1 through 5 and 9 to destination remote VLAN 902.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source interface gigabitethernet1/1 rx
Switch(config)# monitor session 2 filter vlan 1 - 5 , 9
Switch(config)# monitor session 2 destination remote vlan 902
Switch(config)# end
```

Displaying SPAN and RSPAN Status

To display the status of the current SPAN or RSPAN configuration, use the **show monitor** privileged EXEC command.

This example displays the output for the **show monitor** command for SPAN source session 1:

```
Switch# show monitor session 1
Session 1
---------
Type: Local Source Session
Source Ports:
    RX Only:  Fa3/13
    TX Only:       None
    Both:          None

Source VLANs:
    RX Only:       None
    TX Only:       None
    Both:          None
Source RSPAN VLAN: None
Destination Ports: None
    Encapsulation: DOT1Q
    Ingress:Enabled, default VLAN=5
Filter VLANs:      None
Dest RSPAN VLAN: None
Ingress : Enabled, default VLAN=2
Learning : Disabled
```

CHAPTER 44

Configuring System Message Logging

This chapter describes how to configure system message logging on the Catalyst 4500 series switch.

Note For complete syntax and usage information for the commands used in this chapter, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.4*.

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124tcr/tcf_r/index.htm

This chapter consists of these sections:

Understanding System Message Logging

By default, a switch sends the output from system messages and **debug** privileged EXEC commands to a logging process. The logging process controls the distribution of logging messages to various destinations, such as the logging buffer, terminal lines, or a UNIX syslog server, depending on your configuration. The process also sends messages to the console.

Note The syslog format is compatible with 4.3 BSD UNIX.

When the logging process is disabled, messages are sent only to the console. The messages are sent as they are generated, so message and debug output are interspersed with prompts or output from other commands. Messages are displayed on the console after the process that generated them has finished.

You can set the severity level of the messages to control the type of messages displayed on the consoles and each of the destinations. You can time-stamp log messages or set the syslog source address to enhance real-time debugging and management. For information on possible messages, see the system message guide for this release.

You can access logged system messages by using the switch command-line interface (CLI) or by saving them to a properly configured syslog server. The switch software saves syslog messages in an internal buffer on the switchIf the switchfails, the log is lost unless you had saved it to flash memory.

You can remotely monitor system messages by viewing the logs on a syslog server or by accessing the switch through Telnet or through the console port.

Configuring System Message Logging

These sections describe how to configure system message logging:

- System Log Message Format, page 44-2
- Default System Message Logging Configuration, page 44-3
- Disabling Message Logging, page 44-4 (optional)
- Setting the Message Display Destination Device, page 44-4 (optional)
- Synchronizing Log Messages, page 44-5 (optional)
- Enabling and Disabling Timestamps on Log Messages, page 44-7 (optional)
- Enabling and Disabling Sequence Numbers in Log Messages, page 44-7 (optional)
- Defining the Message Severity Level, page 44-8 (optional)
- Limiting Syslog Messages Sent to the History Table and to SNMP, page 44-9 (optional)
- Configuring UNIX Syslog Servers, page 44-10 (optional)

System Log Message Format

System log messages can contain up to 80 characters and a percent sign (%), which follows the optional sequence number or time-stamp information, if configured. Messages are displayed in this format:

seq no:timestamp: %facility-severity-MNEMONIC:description

The part of the message preceding the percent sign depends on the setting of the **service sequence-numbers**, **service timestamps log datetime**, **service timestamps log datetime** [**localtime**] [**msec**] [**show-timezone**], or **service timestamps log uptime** global configuration command.

Table 44-1 describes the elements of syslog messages.

Table 44-1 System Log Message Elements

Element	Description
seq no:	Stamps log messages with a sequence number only if the **service sequence-numbers** global configuration command is configured. For more information, see the "Enabling and Disabling Sequence Numbers in Log Messages" section on page 44-7.
timestamp formats: *mm/dd hh:mm:ss* or *hh:mm:ss* (short uptime) or *d h* (long uptime)	Date and time of the message or event. This information appears only if the **service timestamps log** [**datetime** | **log**] global configuration command is configured. For more information, see the "Enabling and Disabling Timestamps on Log Messages" section on page 44-7.
facility	The facility to which the message refers (for example, SNMP, SYS, and so forth). For a list of supported facilities, see Table 44-4 on page 44-12.
severity	Single-digit code from 0 to 7 that is the severity of the message. For a description of the severity levels, see Table 44-3 on page 44-9.

Table 44-1 System Log Message Elements (continued)

Element	Description
MNEMONIC	Text string that uniquely describes the message.
description	Text string containing detailed information about the event being reported.

This example shows a partial switch system message:

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1, changed
state to down 2
*Mar  1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar  1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

Default System Message Logging Configuration

Table 44-2 shows the default system message logging configuration.

Table 44-2 Default System Message Logging Configuration

Feature	Default Setting
System message logging to the console	Enabled.
Console severity	Debugging (and numerically lower levels; see Table 44-3 on page 44-9).
Logging file configuration	No filename specified.
Logging buffer size	4096 bytes.
Logging history size	1 message.
Timestamps	Disabled.
Synchronous logging	Disabled.
Logging server	Disabled.
Syslog server IP address	None configured.
Server facility	Local7 (see Table 44-4 on page 44-12).
Server severity	Informational (and numerically lower levels; see Table 44-3 on page 44-9).

Disabling Message Logging

Message logging is enabled by default. It must be enabled to send messages to any destination other than the console. When enabled, log messages are sent to a logging process, which logs messages to designated locations asynchronously to the processes that generated the messages.

To disable message logging, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`no logging on`**	Disables message logging.
Step 3	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.

Step 4	`Switch#` **`show running-config`**
	`or`
	`show logging`
		Verifies your entries.

Step 5	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Disabling the logging process can slow down the switch because a process must wait until the messages are written to the console before continuing. When the logging process is disabled, messages are displayed on the console as soon as they are produced, often appearing in the middle of command output.

The **logging synchronous** global configuration command also affects the display of messages to the console. When this command is enabled, messages appear only after you press Return. For more information, see the "Synchronizing Log Messages" section on page 44-5.

To re-enable message logging after it has been disabled, use the **logging on** global configuration command.

Setting the Message Display Destination Device

If message logging is enabled, you can send messages to specific locations in addition to the console.

To specify the locations that receive messages, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`logging buffered`** [*`size`*]	Logs messages to an internal buffer on the switch. The default buffer size is 4096. The range is 4096 to 2147483647 bytes. If the switch, the log file is lost unless you previously saved it to flash memory. See Step 4. **Note** Do not make the buffer size too large because the switch could run out of memory for other tasks. Use the **show memory** privileged EXEC command to view the free processor memory on the switch. However, this value is the maximum available, and the buffer size should *not* be set to this amount.

	Command	Purpose
Step 3	`Switch(config)#` **`logging`** *host*	Logs messages to a UNIX syslog server host. For *host*, specify the name or IP address of the host to be used as the syslog server. To build a list of syslog servers that receive logging messages, enter this command more than once. For complete syslog server configuration steps, see the "Configuring UNIX Syslog Servers" section on page 44-10.
Step 4	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Switch#` **`terminal monitor`**	Logs messages to a nonconsole terminal during the current session. Terminal parameter-setting commands are set locally and do not remain in effect after the session has ended. You must perform this step for each session to see the debugging messages.
Step 6	`Switch#` **`show running-config`**	Verifies your entries.
Step 7	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

The **logging buffered** global configuration command copies logging messages to an internal buffer. The buffer is circular, so newer messages overwrite older messages after the buffer is full. To display the messages that are logged in the buffer, use the **show logging** privileged EXEC command. The first message displayed is the oldest message in the buffer. To clear the contents of the buffer, use the **clear logging** privileged EXEC command.

To disable logging to the console, use the **no logging console** global configuration command. To disable logging to a file, use the **no logging file** [*severity-level-number* | *type*] global configuration command.

Synchronizing Log Messages

You can synchronize unsolicited messages and **debug** privileged EXEC command output with solicited device output and prompts for a specific console port line or virtual terminal line. You can identify the types of messages to be output asynchronously based on the level of severity. You can also configure the maximum number of buffers for storing asynchronous messages for the terminal after which messages are dropped.

When synchronous logging of unsolicited messages and **debug** command output is enabled, unsolicited device output appears on the console or printed after solicited device output appears or printed. Unsolicited messages and **debug** command output appears on the console after the prompt for user input is returned. Therefore, unsolicited messages and **debug** command output are not interspersed with solicited device output and prompts. After the unsolicited messages are displayed, the console again displays the user prompt.

To configure synchronous logging, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**line** [**console** | **vty**] *line-number* [*ending-line-number*]	Specifies the line to be configured for synchronous logging of messages. • Use the **console** keyword for configurations that occur through the switch console port. • Use the **line vty** *line-number* command to specify which vty lines are to have synchronous logging enabled. You use a vty connection for configurations that occur through a Telnet session. The range of line numbers is from 0 to 15. You can change the setting of all 16 vty lines at once by entering: **line vty 0 15** Or you can change the setting of the single vty line being used for your current connection. For example, to change the setting for vty line 2, enter: **line vty 2** When you enter this command, the mode changes to line configuration.
Step 3	**logging synchronous** [**level** [*severity-level* | **all**] | **limit** *number-of-buffers*]	Enables synchronous logging of messages. • (Optional) For **level** *severity-level*, specify the message severity level. Messages with a severity level equal to or higher than this value are printed asynchronously. Low numbers mean greater severity and high numbers mean lesser severity. The default is 2. • (Optional) Specifying **level all** means that all messages are printed asynchronously regardless of the severity level. • (Optional) For **limit** *number-of-buffers*, specify the number of buffers to be queued for the terminal after which new messages are dropped. The range is 0 to 2147483647. The default is 20.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable synchronization of unsolicited messages and debug output, use the **no logging synchronous** [**level** *severity-level* | **all**] [**limit** *number-of-buffers*] line configuration command.

Enabling and Disabling Timestamps on Log Messages

Note By default, log messages are not time-stamped.

To enable time-stamping of log messages, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**service timestamps log uptime** or **service timestamps log datetime** [**msec**] [**localtime**] [**show-timezone**]	Enables log time-stamps. The first command enables time-stamps on log messages, showing the time since the system was rebooted. The second command enables time-stamps on log messages. Depending on the options selected, the timestamp can include the date, time in milliseconds relative to the local time zone, and the time zone name.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable time-stamps for both debug and log messages, use the **no service timestamps** global configuration command.

This example shows part of a logging display with the **service timestamps log datetime** global configuration command enabled:

```
*Mar  1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

This example shows part of a logging display with the **service timestamps log uptime** global configuration command enabled:

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
```

Enabling and Disabling Sequence Numbers in Log Messages

Because more than one log message can have the same timestamp, you can display messages with sequence numbers so that you can unambiguously refer to a single message. By default, sequence numbers in log messages are not displayed.

To enable sequence numbers in log messages, perform this task. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**service sequence-numbers**	Enables sequence numbers.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable sequence numbers, use the **no service sequence-numbers** global configuration command.

This example shows part of a logging display with sequence numbers enabled:

```
000019: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

Defining the Message Severity Level

You can limit messages displayed to the selected device by specifying the severity level of the message, which are described in Table 44-3.

To define the message severity level, perform this task. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**logging console** *level*	Limits messages logged to the console. By default, the console receives debugging messages and numerically lower levels (see Table 44-3 on page 44-9).
Step 3	**logging monitor** *level*	Limits messages logged to the terminal lines. By default, the terminal receives debugging messages and numerically lower levels (see Table 44-3 on page 44-9).
Step 4	**logging trap** *level*	Limits messages logged to the syslog servers. By default, syslog servers receive informational messages and numerically lower levels (see Table 44-3 on page 44-9). For complete syslog server configuration steps, see the "Configuring UNIX Syslog Servers" section on page 44-10.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config** or **show logging**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Note Specifying a *level* causes messages at that level and numerically lower levels to be displayed at the destination.

To disable logging to the console, use the **no logging console** global configuration command. To disable logging to a terminal other than the console, use the **no logging monitor** global configuration command. To disable logging to syslog servers, use the **no logging trap** global configuration command.

Table 44-3 describes the *level* keywords. It also lists the corresponding UNIX syslog definitions from the most severe level to the least severe level.

Table 44-3 Message Logging Level Keywords

Level Keyword	Level	Description	Syslog Definition
emergencies	0	System unstable	LOG_EMERG
alerts	1	Immediate action needed	LOG_ALERT
critical	2	Critical conditions	LOG_CRIT
errors	3	Error conditions	LOG_ERR
warnings	4	Warning conditions	LOG_WARNING
notifications	5	Normal but significant condition	LOG_NOTICE
informational	6	Informational messages only	LOG_INFO
debugging	7	Debugging messages	LOG_DEBUG

The software generates four other categories of messages:

- Error messages about software or hardware malfunctions, displayed at levels **warnings** through **emergencies**. These types of messages mean that the functionality of the switch is affected. For information on how to recover from these malfunctions, see the system message guide for this release.
- Output from the **debug** commands, displayed at the **debugging** level. Debug commands are typically used only by the Technical Assistance Center.
- Interface up or down transitions and system restart messages, displayed at the **notifications** level. This message is only for information; switch functionality is not affected.
- Reload requests and low-process stack messages, displayed at the **informational** level. This message is only for information; switch functionality is not affected.

Limiting Syslog Messages Sent to the History Table and to SNMP

If you enabled syslog message traps to be sent to an SNMP network management station by using the **snmp-server enable trap** global configuration command, you can change the level of messages sent and stored in the switch history table. You also can change the number of messages that are stored in the history table.

Messages are stored in the history table because SNMP traps are not guaranteed to reach their destination. By default, one message of the level **warning** and numerically lower levels (see Table 44-3 on page 44-9) are stored in the history table even if syslog traps are not enabled.

To change the level and history table size defaults, perform this task. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**logging history** *level*[1]	Changes the default level of syslog messages stored in the history file and sent to the SNMP server. See Table 44-3 on page 44-9 for a list of *level* keywords. By default, **warnings**, **errors**, **critical**, **alerts**, and **emergencies** messages are sent.
Step 3	**logging history size** *number*	Specifies the number of syslog messages that can be stored in the history table. The default is to store one message. The range is 0 to 500 messages.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

[1] Table 44-3 lists the level keywords and severity level. For SNMP usage, the severity level values increase by 1. For example, emergencies equal 1, not 0, and critical equals 3, not 2.

When the history table is full (it contains the maximum number of message entries specified with the **logging history size** global configuration command), the oldest message entry is deleted from the table to allow the new message entry to be stored.

To return the logging of syslog messages to the default level, use the **no logging history** global configuration command. To return the number of messages in the history table to the default value, use the **no logging history size** global configuration command.

Configuring UNIX Syslog Servers

The next sections describe how to configure the UNIX server syslog daemon and how to define the UNIX system logging facility.

Logging Messages to a UNIX Syslog Daemon

Before you can send system log messages to a UNIX syslog server, you must configure the syslog daemon on a UNIX server. This procedure is optional.

Log in as root, and perform these steps:

Note Some recent versions of UNIX syslog daemons no longer accept by default syslog packets from the network. If this is the case with your system, use the UNIX **man syslogd** command to decide what options must be added to or removed from the syslog command line to enable logging of remote syslog messages.

Step 1 Add a line such as the following to the file /etc/syslog.conf:

```
local7.debug /usr/adm/logs/cisco.log
```

The **local7** keyword specifies the logging facility to be used; see Table 44-4 on page 44-12 for information on the facilities. The **debug** keyword specifies the syslog level; see Table 44-3 on page 44-9 for information on the severity levels. The syslog daemon sends messages at this level or at a more severe level to the file specified in the next field. The file must already exist, and the syslog daemon must have permission to write to it.

Step 2 Create the log file by entering these commands at the UNIX shell prompt:

```
$ touch /var/log/cisco.log
$ chmod 666 /var/log/cisco.log
```

Step 3 Make sure the syslog daemon reads the new changes:

```
$ kill -HUP `cat /etc/syslog.pid`
```

For more information, see the **man syslog.conf** and **man syslogd** commands on your UNIX system.

Configuring the UNIX System Logging Facility

When sending system log messages to an external device, you can cause the switch to identify its messages as originating from any of the UNIX syslog facilities.

Beginning in privileged EXEC mode, follow these steps to configure UNIX system facility message logging. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**logging** *host*	Logs messages to a UNIX syslog server host by entering its IP address. To build a list of syslog servers that receive logging messages, enter this command more than once.
Step 3	**logging trap** *level*	Limits messages logged to the syslog servers. Be default, syslog servers receive informational messages and lower. See Table 44-3 on page 44-9 for *level* keywords.
Step 4	**logging facility** *facility-type*	Configures the syslog facility. See Table 44-4 on page 44-12 for *facility-type* keywords. The default is **local7**.
Step 5	**end**	Returns to privileged EXEC mode.

	Command	Purpose
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To remove a syslog server, use the **no logging** *host* global configuration command, and specify the syslog server IP address. To disable logging to syslog servers, enter the **no logging trap** global configuration command.

Table 44-4 lists the UNIX system facilities supported by the software. For more information about these facilities, consult the operator's manual for your UNIX operating system.

Table 44-4 Logging Facility-Type Keywords

Facility Type Keyword	Description
auth	Authorization system
cron	Cron facility
daemon	System daemon
kern	Kernel
local0-7	Locally defined messages
lpr	Line printer system
mail	Mail system
news	USENET news
sys9-14	System use
syslog	System log
user	User process
uucp	UNIX-to-UNIX copy system

Displaying the Logging Configuration

To display the logging configuration and the contents of the log buffer, use the **show logging** privileged EXEC command. For information about the fields in this display, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.3*.

CHAPTER 45

Configuring SNMP

This chapter describes how to configure the Simple Network Management Protocol (SNMP) on the Catalyst 4500 series switch.

Note For complete syntax and usage information for the commands used in this chapter, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.4*.

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124tcr/tcf_r/index.htm

This chapter consists of these sections:

- Understanding SNMP, page 45-1
- Configuring SNMP, page 45-5
- Displaying SNMP Status, page 45-17

Understanding SNMP

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a MIB. The SNMP manager can be part of a network management system (NMS) such as CiscoWorks. The agent and MIB reside on the switch. To configure SNMP on the switch, you define the relationship between the manager and the agent.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into the agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager’s requests to get or set data.

An agent can send unsolicited traps to the manager. Traps are messages alerting the SNMP manager to a condition on the network. Traps can mean improper user authentication, restarts, link status (up or down), MAC address tracking, closing of a Transmission Control Protocol (TCP) connection, loss of connection to a neighbor, or other significant events.

This section includes information about these topics:

- SNMP Versions, page 45-2
- SNMP Manager Functions, page 45-3
- SNMP Agent Functions, page 45-4

- SNMP Community Strings, page 45-4
- Using SNMP to Access MIB Variables, page 45-4
- SNMP Notifications, page 45-5

SNMP Versions

The Catalyst 4500 series switch supports these SNMP versions:

- SNMPv1—The Simple Network Management Protocol, a Full Internet Standard, defined in RFC 1157.
- SNMPv2C replaces the Party-based Administrative and Security Framework of SNMPv2Classic with the community-string-based Administrative Framework of SNMPv2C while retaining the bulk retrieval and improved error handling of SNMPv2Classic. It has these features:
  - SNMPv2—Version 2 of the Simple Network Management Protocol, a Draft Internet Standard, defined in RFCs 1902 through 1907.
  - SNMPv2C—The community-string-based Administrative Framework for SNMPv2, an Experimental Internet Protocol defined in RFC 1901.
- SNMPv3—Version 3 of the SNMP is an interoperable standards-based protocol defined in RFCs 2273 to 2275. SNMPv3 provides secure access to devices by authenticating and encrypting packets over the network and includes these security features:
  - Message integrity—ensuring that a packet was not tampered with in transit
  - Authentication—determining that the message is from a valid source
  - Encryption—mixing the contents of a package to prevent it from being read by an unauthorized source.

Note To select encryption, enter the **priv** keyword. This keyword is available only when the crypto (encrypted) software image is installed.

Both SNMPv1 and SNMPv2C use a community-based form of security. The community of managers able to access the agent’s MIB is defined by an IP address access control list and password.

SNMPv2C includes a bulk retrieval mechanism and more detailed error message reporting to management stations. The bulk retrieval mechanism retrieves tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2C improved error-handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes in SNMPv2C report the error type.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy set up for a user and the group within which the user resides. A security level is the permitted level of security within a security model. A combination of the security level and the security model determine which security mechanism is used when handling an SNMP packet. Available security models are SNMPv1, SNMPv2C, and SNMPv3.

Table 45-1 identifies the characteristics of the different combinations of security models and levels.

Table 45-1 SNMP Security Models and Levels

Model	Level	Authentication	Encryption	Result
SNMPv1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv2C	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv3	noAuthNoPriv	Username	No	Uses a username match for authentication.
SNMPv3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
SNMPv3	authPriv (requires the cryptographic software image)	MD5 or SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard.

You must configure the SNMP agent to use the SNMP version supported by the management station. Because an agent can communicate with multiple managers, you can configure the software to support communications using SNMPv1, and SNMPv2C, and SNMPv3 protocols.

SNMP Manager Functions

The SNMP manager uses information in the MIB to perform the operations described in Table 45-2.

Table 45-2 SNMP Operations

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves a value from a variable within a table.[1]
get-bulk-request[2]	Retrieves large blocks of data, such as multiple rows in a table, that would otherwise require the transmission of many small blocks of data.
get-response	Replies to a get-request, get-next-request, and set-request sent by an NMS.
set-request	Stores a value in a specific variable.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

1. With this operation, an SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table.
2. The **get-bulk** command only works with SNMPv2 or later.

SNMP Agent Functions

The SNMP agent responds to SNMP manager requests as follows:

- Get a MIB variable—The SNMP agent begins this function in response to a request from the NMS. The agent retrieves the value of the requested MIB variable and responds to the NMS with that value.
- Set a MIB variable—The SNMP agent begins this function in response to a message from the NMS. The SNMP agent changes the value of the MIB variable to the value requested by the NMS.

The SNMP agent also sends unsolicited trap messages to notify an NMS that a significant event has occurred on the agent. Examples of trap conditions include, but are not limited to, when a port or module goes up or down, when spanning-tree topology changes occur, and when authentication failures occur.

SNMP Community Strings

SNMP community strings authenticate access to MIB objects and function as embedded passwords. In order for the NMS to access the switch, the community string definitions on the NMS must match at least one of the three community string definitions on the switch.

A community string can have one of these attributes:

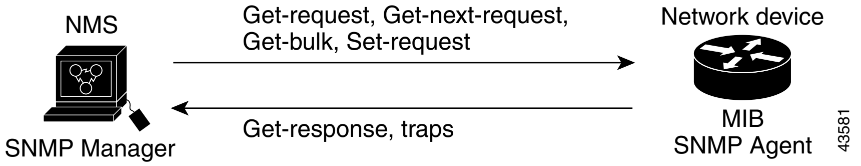
- Read-only (RO)—Gives read access to authorized management stations to all objects in the MIB except the community strings, but does not allow write access
- Read-write (RW)—Gives read and write access to authorized management stations to all objects in the MIB, but does not allow access to the community strings
- Read-write-all—Gives read and write access to authorized management stations to all objects in the MIB, including the community strings

Using SNMP to Access MIB Variables

An example of an NMS is the CiscoWorks network management software. CiscoWorks 2000 software uses the switch MIB variables to set device variables and to poll devices on the network for specific information. The results of a poll can be displayed as a graph and analyzed to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, monitor traffic loads, and more.

As shown in Figure 45-1, the SNMP agent gathers data from the MIB. The agent can send traps, or notification of certain events, to the SNMP manager, which receives and processes the traps. Traps alert the SNMP manager to a condition on the network such as improper user authentication, restarts, link status (up or down), MAC address tracking, and so forth. The SNMP agent also responds to MIB-related queries sent by the SNMP manager in *get-request*, *get-next-request*, and *set-request* format.

Figure 45-1 SNMP Network



SNMP Notifications

SNMP allows the switch to send notifications to SNMP managers when particular events occur. SNMP notifications can be sent as traps or inform requests. In command syntax, unless there is an option in the command to select either traps or informs, the keyword *traps* refers to either traps or informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

Note SNMPv1 does not support informs.

Traps are unreliable because the receiver does not send an acknowledgment when it receives a trap, and the sender cannot determine if the trap was received. When an SNMP manager receives an inform request, it acknowledges the message with an SNMP response protocol data unit (PDU). If the sender does not receive a response, the inform request can be sent again. Because they can be re-sent, informs are more likely than traps to reach their intended destination.

The characteristics that make informs more reliable than traps also consume more resources in the switch and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out. Traps are sent only once, but an inform might be re-sent or retried several times. The retries increase traffic and contribute to a higher overhead on the network. Therefore, traps and informs require a trade-off between reliability and resources. If it is important that the SNMP manager receive every notification, use inform requests. If traffic on the network or memory in the switch is a concern and notification is not required, use traps.

Configuring SNMP

This section describes how to configure SNMP on your switch. It contains this configuration information:

- Default SNMP Configuration, page 45-6
- SNMP Configuration Guidelines, page 45-6
- Disabling the SNMP Agent, page 45-7
- Configuring Community Strings, page 45-7
- Configuring SNMP Groups and Users, page 45-9
- Configuring SNMP Notifications, page 45-11
- Setting the Agent Contact and Location Information, page 45-15
- Limiting TFTP Servers Used Through SNMP, page 45-15
- SNMP Examples, page 45-16

Default SNMP Configuration

Table 45-3 shows the default SNMP configuration.

Table 45-3 Default SNMP Configuration

Feature	Default Setting
SNMP agent	Enabled
SNMP trap receiver	None configured
SNMP traps	None enabled except the trap for TCP connections (**tty**)
SNMP version	If no **version** keyword is present, the default is Version 1.
SNMPv3 authentication	If no keyword is entered, the default is the **noauth** (noAuthNoPriv) security level.
SNMP notification type	If no type is specified, all notifications are sent.

SNMP Configuration Guidelines

An SNMP *group* is a table that maps SNMP users to SNMP views. An SNMP *user* is a member of an SNMP group. An SNMP *host* is the recipient of an SNMP trap operation. An SNMP *engine ID* is a name for the local or remote SNMP engine.

When configuring SNMP, follow these guidelines:

- When configuring an SNMP group, do not specify a notify view. The **snmp-server host** global configuration command autogenerates a notify view for the user and then adds it to the group associated with that user. Modifying the group's notify view affects all users associated with that group. For information about when you should configure notify views, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.
- To configure a remote user, specify the IP address or port number for the remote SNMP agent of the device where the user resides.
- Before you configure remote users for a particular agent, configure the SNMP engine ID, using the **snmp-server engineID** global configuration with the **remote** option. The remote agent's SNMP engine ID and user password are used to compute the authentication and privacy digests. If you do not configure the remote engine ID first, the configuration command fails.
- When configuring SNMP informs, you need to configure the SNMP engine ID for the remote agent in the SNMP database before you can send proxy requests or informs to it.
- If a local user is not associated with a remote host, the switch does not send informs for the **auth** (authNoPriv) and the **priv** (authPriv) authentication levels.
- Changing the value of the SNMP engine ID has important side effects. A user's password (entered on the command line) is converted to an MD5 or SHA security digest based on the password and the local engine ID. The command-line password is then destroyed, as required by RFC 2274. Because of this deletion, if the value of the engine ID changes, the security digests of SNMPv3 users become invalid, and you need to reconfigure SNMP users by using the **snmp-server user** *username* global configuration command. Similar restrictions require the reconfiguration of community strings when the engine ID changes.

Disabling the SNMP Agent

To disable the SNMP agent, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# no snmp-server`	Disables the SNMP agent operation.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show running-config`	Verifies your entries.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

The **no snmp-server** global configuration command disables all running versions (Version 1, Version 2C, and Version 3) on the device. No specific IOS command exists to enable SNMP. The first **snmp-server** global configuration command that you enter enables all versions of SNMP.

Configuring Community Strings

You use the SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to permit access to the agent on the switch. Optionally, you can specify one or more of these characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent
- A MIB view, which defines the subset of all MIB objects accessible to the given community
- Read and write or read-only permission for the MIB objects accessible to the community

To configure a community string on the switch, perform this task:

	Command	Purpose
Step 1	Switch# **configure terminal**	Enters global configuration mode.
Step 2	Switch(config)# [**no**] **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*]	Configures the community string. • For *string*, specify a string that acts like a password and permits access to the SNMP protocol. You can configure one or more community strings up to 117 characters. • (Optional) For **view**, specify the view record accessible to the community. • (Optional) Specify either read-only (**ro**) if you want authorized management stations to retrieve MIB objects, or specify read-write (**rw**) if you want authorized management stations to retrieve and modify MIB objects. By default, the community string permits read-only access to all objects. • (Optional) For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999. To remove a specific community string, use the **no snmp-server community** *string* global configuration command.
Step 3	Switch(config)# **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]	(Optional) If you specified an IP standard access list number in Step 2, then create the list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. • For *source*, enter the IP address of the SNMP managers that are permitted to use the community string to gain access to the agent. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.
Step 4	Switch(config)# **end**	Return to privileged EXEC mode.
Step 5	Switch# **show running-config**	Verifies your entries.
Step 6	Switch# **copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Note To disable access for an SNMP community, set the community string for that community to the null string (do not enter a value for the community string).

Note The **snmp-server enable informs** command is not supported. To enable the sending of SNMP inform notifications, use the **snmp-server enable traps** command combined with the **snmp-server host** *host-addr* **informs** command.

This example shows how to assign the string *comaccess* to SNMP, to allow read-only access, and to specify that IP access list 4 can use the community string to gain access to the switch SNMP agent:

```
Switch(config)# snmp-server community comaccess ro 4
```

Configuring SNMP Groups and Users

You can specify an identification name (engine ID) for the local or remote SNMP server engine on the switch. You can configure an SNMP server group that maps SNMP users to SNMP views, and you can add new users to the SNMP group.

To configure SNMP on the switch, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`snmp-server engineID`** {**`local`** *`engineid-string`* | **`remote`** *`ip-address`* [**`udp-port`** *`port-number`*] *`engineid-string`*}	Configures a name for either the local or remote copy of SNMP. • The *engineid-string* is a 24-character ID string with the name of the copy of SNMP. You need not specify the entire 24-character engine ID if it has trailing zeros. Specify only the portion of the engine ID up to the point where only zeros remain in the value. For example, to configure an engine ID of 123400000000000000000000, you can enter this: **snmp-server engineID local 1234** • If you select **remote**, specify the *ip-address* of the device that contains the remote copy of SNMP and the optional UDP port on the remote device. The default is 162.

	Command	Purpose
Step 3	`Switch(config)#` **`snmp-server group`** *`groupname`* {**`v1`** | **`v2c`** | **`v3`** [**`auth`** | **`noauth`** | **`priv`**]} [**`read`** *`readview`*] [**`write`** *`writeview`*] [**`notify`** *`notifyview`*] [**`access`** *`access-list`*]	Configures a new SNMP group on the remote device. • For *groupname,* specify the name of the group. • Specify a security model: – **v1** is the least secure of the possible security models. – **v2c** is the second least secure model. It allows transmission of informs and integers twice the normal width. – **v3,** the most secure, requires you to select an authentication level: **auth**—Enables the Message Digest 5 (MD5) and the Secure Hash Algorithm (SHA) packet authentication. **noauth**—The noAuthNoPriv security level. This is the default if no keyword is specified. **priv**—Enables Data Encryption Standard (DES) packet encryption (also called *privacy*). **Note** The **priv** keyword is available only when the crypto software image is installed. • (Optional) Enter **read** *readview* with a string (not to exceed 64 characters) that is the name of the view in which you can only view the contents of the agent. • (Optional) Enter **write** *writeview* with a string (not to exceed 64 characters) that is the name of the view in which you enter data and configure the contents of the agent. • (Optional) Enter **notify** *notifyview* with a string (not to exceed 64 characters) that is the name of the view in which you specify a notify, inform, or trap. • (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.

	Command	Purpose
Step 4	`Switch(config)#` **`snmp-server user`** *`username groupname`* `[`**`remote`** *`host`* `[`**`udp-port`** *`port`*`]]` `{`**`v1`** `|` **`v2c`** `|` **`v3`** `[`**`auth`** `{`**`md5`** `|` **`sha`**`}` *`auth-password`*`]}` `[`**`encrypted`**`]` `[`**`access`** *`access-list`*`]`	Configures a new user to an SNMP group. • The *username* is the name of the user on the host that connects to the agent. • The *groupname* is the name of the group to which the user is associated. • (Optional) Enter **remote** to specify a remote SNMP entity to which the user belongs and the hostname or IP address of that entity with the optional UDP port number. The default is 162. • Enter the SNMP version number (**v1**,or **v2c**, or **v3**). If you enter **v3**, you have these additional options: – **auth** is an authentication level setting session, which can be either the HMAC-MD5-96 or the HMAC-SHA-96 authentication level, and requires a password string (not to exceed 64 characters). – **encrypted** specifies that the password appears in encrypted format. • (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.
Step 5	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 6	`Switch#` **`show running-config`**	Verifies your entries.
Step 7	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Configuring SNMP Notifications

A trap manager is a management station that receives and processes traps. Traps are system alerts that the switch generates when certain events occur. By default, no trap manager is defined, and no traps are sent. Switches running IOS Cisco IOS Release 12.2(31)SG can have an unlimited number of trap managers.

Note Many commands use the word *traps* in the command syntax. Unless there is an option in the command to select either traps or informs, the keyword *traps* refers to either traps, informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

Table 45-4 describes the supported switch traps (notification types). You can enable any or all of these traps and configure a trap manager to receive them.

Table 45-4 Switch Notification Types

Notification Type Keyword	Description
bgp	Generates BGP state change traps. **Note** This option is only available when the enhanced multilayer image is installed.
bridge	Generates STP bridge MIB traps.
config	Generates a trap for SNMP configuration changes.
config-copy	Generates a trap for SNMP copy configuration changes.
cpu	Allows cpu-related traps.
eigrp	Enable BGP traps. **Note** This option is only available when the enhanced multilayer image is installed.
entity	Generates a trap for SNMP entity changes.
envmon	Generates environmental monitor traps. You can enable any or all of these environmental traps: fan, shutdown, supply, temperature.
flash	Generates SNMP FLASH notifications.
fru-ctrl	Enable SNMP entity FRU control traps.
hsrp	Generates a trap for Hot Standby Router Protocol (HSRP) changes.
ipmulticast	Generates a trap for IP multicast routing changes.
isis	Enable IS-IS traps. **Note** This option is only available when the enhanced multilayer image is installed.
mac-notification	Generates a trap for MAC address notifications.
msdp	Generates a trap for Multicast Source Discovery Protocol (MSDP) changes. **Note** This option is only available when the enhanced multilayer image is installed.
ospf	Generates a trap for Open Shortest Path First (OSPF) changes. You can enable any or all of these traps: Cisco specific, errors, link-state advertisement, rate limit, retransmit, and state changes. **Note** This option is only available when the enhanced multilayer image is installed.
pim	Generates a trap for Protocol-Independent Multicast (PIM) changes. You can enable any or all of these traps: invalid PIM messages, neighbor changes, and rendezvous point (RP)-mapping changes.
port-security	Generates SNMP port security traps. You can also set a maximum trap rate per second. The range is from 0 to 1000; the default is 0, which means that there is no rate limit.
rf	Enable all SNMP traps defined in Cisco-RF-MIB.
snmp	Generates a trap for SNMP-type notifications for authentication, cold start, warm start, link up or link down.

Table 45-4 Switch Notification Types (continued)

Notification Type Keyword	Description
storm-control	Generates a trap for SNMP storm-control. You can also set a maximum trap rate per second. The range is from 0 to 1000; the default is 0 (no limit is imposed; a trap is sent at every occurrence).
stpx	Generates SNMP STP Extended MIB traps.
syslog	Generates SNMP syslog traps.
tty	Generates a trap for TCP connections. This trap is enabled by default.
vlan-membership	Generates a trap for SNMP VLAN membership changes.
vlancreate	Generates SNMP VLAN created traps.
vlandelete	Generates SNMP VLAN deleted traps.
vtp	Generates a trap for VLAN Trunking Protocol (VTP) changes.

You can use the **snmp-server host** global configuration command to a specific host to receive the notification types listed in Table 45-4.

To configure the switch to send traps or informs to a host, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`snmp-server engineID remote`** *`ip-address engineid-string`*	Specifies the engine ID for the remote host.
Step 3	`Switch(config)#` **`snmp-server user`** *`username groupname`* **`remote`** *`host`* [**`udp-port`** *`port`*] {**`v1`** | **`v2c`** | **`v3`** [**`auth`** {**`md5`** | **`sha`**} *`auth-password`*]} [**`encrypted`**] [**`access`** *`access-list`*]	Configures an SNMP user to be associated with the remote host created in Step 2. **Note** You cannot configure a remote user for an address without first configuring the engine ID for the remote host. If you try to configure the user before configuring the remote engine ID, you receive an error message, and the command is not executed.

	Command	Purpose
Step 4	`Switch(config)# snmp-server host host-addr [traps | informs] [version {1 | 2c | 3 [auth | noauth | priv]}] community-string [udp-port port] [notification-type]`	Specifies the recipient of an SNMP trap operation. • For *host-addr,* specify the name or Internet address of the host (the targeted recipient). • (Optional) Enter **traps** (the default) to send SNMP traps to the host. • (Optional) Enter **informs** to send SNMP informs to the host. • (Optional) Specify the SNMP **version** (**1**, **2c**, or **3**). SNMPv1 does not support informs. • (Optional) For Version 3, select authentication level **auth, noauth**, or **priv.** **Note** The **priv** keyword is available only when the crypto software image is installed. • For *community-string*, enter the password-like community string sent with the notification operation. • (Optional) For **udp-port** *port*, enter the remote device UDP port. • (Optional) For *notification-type*, use the keywords listed in Table 45-4 on page 45-12. If no type is specified, all notifications are sent.
Step 5	`Switch(config)# snmp-server enable traps notification-types`	Enables the switch to send traps or informs and specify the type of notifications to be sent. For a list of notification types, see Table 45-4 on page 45-12, or enter this: **snmp-server enable traps ?** To enable multiple types of traps, you must enter a separate **snmp-server enable traps** command for each trap type.
Step 6	`Switch(config)# snmp-server trap-source interface-id`	(Optional) Specifies the source interface, which provides the IP address for the trap message. This command also sets the source IP address for informs.
Step 7	`Switch(config)# snmp-server queue-length length`	(Optional) Establishes the message queue length for each trap host. The range is 1 to 1000; the default is 10.
Step 8	`Switch(config)# snmp-server trap-timeout seconds`	(Optional) Defines how often to resend trap messages. The range is 1 to 1000; the default is 30 seconds.
Step 9	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 10	`Switch# show running-config`	Verifies your entries.
Step 11	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

The **snmp-server host** command specifies which hosts receive the notifications. The **snmp-server enable trap** command globally enables the mechanism for the specified notification (for traps and informs). To enable a host to receive an inform, you must configure an **snmp-server host informs** command for the host and globally enable informs by using the **snmp-server enable traps** command.

To remove the specified host from receiving traps, use the **no snmp-server host** *host* global configuration command. The **no snmp-server host** command with no keywords disables traps, but not informs, to the host. To disable informs, use the **no snmp-server host informs** global configuration command. To disable a specific trap type, use the **no snmp-server enable traps** *notification-types* global configuration command.

Setting the Agent Contact and Location Information

To set the system contact and location of the SNMP agent so that these descriptions can be accessed through the configuration file, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`snmp-server contact`** *`text`*	Sets the system contact string. For example: **`snmp-server contact Dial System Operator at beeper 21555`**`.`
Step 3	`Switch(config)#` **`snmp-server location`** *`text`*	Sets the system location string. For example: **`snmp-server location Building 3/Room 222`**
Step 4	`Switch(config)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Switch#` **`show running-config`**	Verifies your entries.
Step 6	`Switch#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

Limiting TFTP Servers Used Through SNMP

To limit the TFTP servers used for saving and loading configuration files through SNMP to the servers specified in an access list, perform this task:

	Command	Purpose
Step 1	`Switch#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Switch(config)#` **`snmp-server tftp-server-list`** *`access-list-number`*	Limits TFTP servers used for configuration file copies through SNMP to the servers in the access list. For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.
Step 3	`Switch(config)#` **`access-list`** *`access-list-number`* {**`deny`** | **`permit`**} *`source`* [*`source-wildcard`*]	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. • For *source*, enter the IP address of the TFTP servers that can access the switch. • (Optional) For *source-wildcard*, enter the wildcard bits, in dotted decimal notation, to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.

	Command	Purpose
Step 4	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 5	`Switch# show running-config`	Verifies your entries.
Step 6	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

SNMP Examples

This example shows how to enable all versions of SNMP. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public*. This configuration does not cause the switch to send any traps.

```
Switch(config)# snmp-server community public
```

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string *public*. The switch also sends VTP traps to the hosts 192.180.1.111 and 192.180.1.33 using SNMPv1 and to the host 192.180.1.27 using SNMPv2C. The community string *public* is sent with the traps.

```
Switch(config)# snmp-server community public
Switch(config)# snmp-server enable traps vtp
Switch(config)# snmp-server host 192.180.1.27 version 2c public
Switch(config)# snmp-server host 192.180.1.111 version 1 public
Switch(config)# snmp-server host 192.180.1.33 public
```

This example shows how to allow read-only access for all objects to members of access list 4 that use the *comaccess* community string. No other SNMP managers have access to any objects. SNMP Authentication Failure traps are sent by SNMPv2C to the host *cisco.com* using the community string *public*.

```
Switch(config)# snmp-server community comaccess ro 4
Switch(config)# snmp-server enable traps snmp authentication
Switch(config)# snmp-server host cisco.com version 2c public
```

This example shows how to send Entity MIB traps to the host *cisco.com*. The community string is restricted. The first line enables the switch to send Entity MIB traps in addition to any traps previously enabled. The second line specifies the destination of these traps and overwrites any previous **snmp-server host** commands for the host *cisco.com*.

```
Switch(config)# snmp-server enable traps entity
Switch(config)# snmp-server host cisco.com restricted entity
```

This example shows how to enable the switch to send all traps to the host *myhost.cisco.com* using the community string *public*:

```
Switch(config)# snmp-server enable traps
Switch(config)# snmp-server host myhost.cisco.com public
```

This example shows how to associate a user with a remote host and to send **auth** (authNoPriv) authentication-level informs when the user enters global configuration mode:

```
Switch(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b
Switch(config)# snmp-server group authgroup v3 auth
Switch(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3 auth md5
mypassword
Switch(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
Switch(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser config
Switch(config)# snmp-server enable traps
Switch(config)# snmp-server inform retries 0
```

Displaying SNMP Status

To display SNMP input and output statistics, including the number of illegal community string entries, errors, and requested variables, use the **show snmp** privileged EXEC command.You can also use the other privileged EXEC commands in Table 45-5 to display SNMP information. For information about the fields in the output displays, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.4*.

Table 45-5 Commands for Displaying SNMP Information

Feature	Default Setting
show snmp	Displays SNMP statistics.
show snmp engineID	Displays information on the local SNMP engine and all remote engines that have been configured on the device.
show snmp group	Displays information on each SNMP group on the network.
show snmp pending	Displays information on pending SNMP requests.
show snmp sessions	Displays information on the current SNMP sessions.
show snmp user	Displays information on each SNMP user name in the SNMP users table.

Note The **snmp-server enable informs** command is not supported. To enable the sending of SNMP inform notifications, use the **snmp-server enable traps** command combined with the **snmp-server host** *host-addr* **informs** command.

CHAPTER 46

Configuring NetFlow

Note Netflow is *not* supported on Supervisor Engine 6-E.

This chapter describes how to configure NetFlow Statistics on the Catalyst 4500 series switches. It also provides guidelines, procedures, and configuration examples.

Note To use the NetFlow feature, you must have the Supervisor Engine V-10GE (the functionality is embedded in the supervisor engine), or the NetFlow Services Card (WS-F4531) and either a Supervisor Engine IV or a Supervisor Engine V.

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Note Refer to the *NetFlow Solutions Guide* for more detailed information on NetFlow usage and management.

The following topics are included:

- Overview of NetFlow Statistics Collection, page 46-1
- Configuring NetFlow Statistics Collection, page 46-6
- NetFlow Statistics Collection Configuration Example, page 46-13
- NetFlow Configuration Examples, page 46-14

Overview of NetFlow Statistics Collection

A network flow is defined as a unidirectional stream of packets between a given source and destination —both defined by a network-layer IP address and transport-layer port number. Specifically, a flow is identified as the combination of the following fields: source IP address, destination IP address, source port number, destination port number, protocol type, type of service, and input interface.

NetFlow Statistics is a global traffic monitoring feature that allows flow-level monitoring of all IPv4-routed traffic through the switch using NetFlow Data Export (NDE). Collected statistics can be exported to an external device (NetFlow Collector/Analyzer) for further processing. Network planners can selectively enable NetFlow Statistics (and NDE) on a per-device basis to gain traffic performance, control, or accounting benefits in specific network locations.

NetFlow exports flow information in UDP datagrams in one of two formats. The version 1 format was the initial released version, and version 5 is a later enhancement to add Border Gateway Protocol (BGP) autonomous system (AS) information and flow sequence numbers. In version 1 and version 5 format, the datagram consists of a header and one or more flow records. The first field of the header contains the version number of the export datagram.

This section contains the following subsections:

- Information Derived from Hardware, page 46-3
- Information Derived from Software, page 46-4
- Assigning the Input and Output Interface and AS Numbers, page 46-4
- Feature Interaction of Netflow Statistics with UBRL and Microflow Policing, page 46-5
- VLAN Statistics, page 46-5

NDE Versions

The Catalyst 4500 series switch supports NDE versions 1 and 5 for the captured statistics. NetFlow aggregation requires NDE version 8.

Depending on the current flow mask, some fields in the flow records might not have values. Unsupported fields contain a zero (0).

The following tables describe the supported fields for NDE version 5:

- Table 46-1—Version 5 header format
- Table 46-2—Version 5 flow record format

Table 46-1 NDE Version 5 Header Format

Bytes	Content	Description
0–1	version	NetFlow export format version number
2–3	count	Number of flows exported in this packet (1–30)
4–7	SysUptime	Current time in milliseconds since router booted
8–11	unix_secs	Current seconds since 0000 UTC 1970
12–15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970
16–19	flow_sequence	Sequence counter of total flows seen
20–21	engine_type	Type of flow switching engine
21–23	engine_id	Slot number of the flow switching engine

Table 46-2 NDE Version 5 Flow Record Format

			Flow masks: **• X=Populated** **• A=Additional field**					
Bytes	**Content**	**Description**	**Source**	**Destination**	**Destination Source**	**Destination Source Interface**	**Full**	**Full Interface**
0–3	srcaddr	Source IP address	X		X	X	X	X
4–7	dstaddr	Destination IP address		X	X	X	X	X
8–11	nexthop	Next hop router's IP address		A[1]	A	A	A	A
12–13	input	Ingress interface SNMP ifIndex				X		X
14–15	output	Egress interface SNMP ifIndex		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	X	X	X	X	X	X
28–31	last	SysUptime at the time the last packet of the flow was received	X	X	X	X	X	X
32–33	srcport	Layer 4 source port number or equivalent					X[2]	X[2]
34–35	dstport	Layer 4 destination port number or equivalent					X	X
36	pad1	Unused (zero) byte						
37	tcp_flags	Cumulative OR of TCP flags						
38	prot	Layer 4 protocol (for example, 6=TCP, 17=UDP)					X	X
39	tos	IP type-of-service byte						
40–41	src_as	Autonomous system number of the source, either origin or peer	X		X	X	X	X
42–43	dst_as	Autonomous system number of the destination, either origin or peer		X	X	X	X	X
44–45	src_mask	Source address prefix mask bits	X		X	X	X	X
46–47	dst_mask	Destination address prefix mask bits		X	X	X	X	X
48	pad2	Pad 2 is unused (zero) bytes						

[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, ICMP traffic contains the ICMP code and type values.

Information Derived from Hardware

Information available in a typical NetFlow record from hardware includes the following:

- the packet and byte counts
- start and end timestamps

- source and destination IP addresses
- IP protocol
- source and destination port numbers

Information Derived from Software

Information available in a typical NetFlow record from software includes the following:

- Input and output identifiers
- Routing information, including next-hop address, origin and peer AS, source and destination prefix mask

Assigning the Input and Output Interface and AS Numbers

The following topics are discussed:

- Assigning the Inferred Fields, page 46-4
- Assigning the Output Interface and Output Related Inferred Fields, page 46-4
- Assigning the Input Interface and Input Related Inferred Fields, page 46-5

Assigning the Inferred Fields

The Catalyst 4500 series switch collects netflow flows in hardware. The hardware collects a sub-set of all the netflow flow fields. The rest of the fields are then filled in by the software when the software examines the routing state.

The Netflow Services Card does not provide enough information to accurately and consistently determine the input interface, output interface and other routing information associated with NetFlow Flows. The Catalyst 4500 series switch has a software mechanism to compensate for this. The mechanism is described in the next paragraph.

Assigning the Output Interface and Output Related Inferred Fields

Software determines the output interface information by looking up the Forwarding Information Base (FIB) entry in the default FIB table (based on the destination IP address). From this FIB entry, the software gains access to the destination AS number for this destination IP address, as well as the appropriate adjacency that stores the interface information. Therefore, the output interface is based solely on the destination IP address. If load balancing is enabled on the switch, the load balancing hash, instead of looking at the adjacency in the FIB entry, is applied to access the appropriate FIB path and access the appropriate adjacency. Although this process typically yields correct results, an inaccuracy can occur when using a PBR that shares IP addresses with the default FIB table. Under these circumstances, there would then be multiple FIB table entries and associated adjacencies for the same destination IP address.

Assigning the Input Interface and Input Related Inferred Fields

Similarly, the input interface and the source AS number for the source IP address are determined by looking up the FIB entry in the default FIB table based on the source IP address. Therefore, the input interface is based solely on the source IP address and a reverse lookup is done to determine to which interface a packet with this IP destination address needs to be routed. This process assumes that the forwarding paths are symmetrical. However, if this process yields multiple input interfaces, a deterministic algorithm is applied to pick one of them the one with the lowest IP address. Although this process typically yields correct values, there are scenarios where the values are inaccurate:

- If load balancing is being applied by an upstream adjacent switch, one input interface must be chosen arbitrarily out of the multiple input interfaces available. This action is necessary because the input interface that would be used depends on the type of load balancing algorithm being deployed by the adjacent upstream switch. It is not always feasible to know the algorithm. Therefore, all flow statistics are attributed to one input interface. Software selects the interface with the lowest IP subnet number.
- In an asymmetric routing scheme in which the traffic for an IP subnet might be received on one interface and sent on another, the inferences noted previously for selecting an input interface, based on a reverse lookup, would be incorrect and cannot be verified.
- If PBR or VRF is enabled on the switch and the flow is destined to an address that resides in the PBR or VRF range or is sourced from an address that resides in the PBR or VRF range, the information is incorrect. In this case, the input and output interface most likely points to the default route (if configured) or have no value at all (NULL)
- If VRF is enabled on the switch on some interfaces and the flow comes from a VRF interface, the information is incorrect. In this case, the input and output interface most likely points to the default route (if configured) or have no value (NULL).

Note The Supervisor Engine V-10GE provides the input interface information via hardware, improving the accuracy of NetFlow information.

Feature Interaction of Netflow Statistics with UBRL and Microflow Policing

On systems with Supervisor Engine V-10GE, there is a feature interaction between Netflow Statistics and UBRL (User Based Rate Limiting). As part of correctly configuring UBRL on a given interface, the class-map must specify a flow-mask. In turn, this flow mask is used to create hardware-based netflow statistics for the flow. By default, for traditional full-flow netflow statistics, the full-flow mask is used. With UBRL, however, the masks can differ. If UBRL is configured on a given interface, the statistics are collected based on the mask configured for UBRL. Consequently, the system does not collect full-flow statistics for traffic transiting an interface configured with UBRL. For more details, refer to the "Configuring User Based Rate Limiting" section on page 32-42.

VLAN Statistics

With NetFlow support, you can report Layer 2 output VLAN statistics, as well as VLAN statistics for routed traffic in and out of a VLAN.

The following example shows the CLI output for a specific VLAN:

```
Switch# show vlan counters or show vlan id 22 count
* Multicast counters include broadcast packets
Vlan Id                                            :22
L2 Unicast Packets                                 :38
L2 Unicast Octets                                  :2432
L3 Input Unicast Packets                           :14344621
L3 Input Unicast Octets                            :659852566
L3 Output Unicast Packets                          :8983050
L3 Output Unicast Octets                           :413220300
L3 Output Multicast Packets                        :0
L3 Output Multicast Octets                         :0
L3 Input Multicast Packets                         :0
L3 Input Multicast Octets                          :0
L2 Multicast Packets                               :340
L2 Multicast Octets                                :21760
```

Note NetFlow support has hardware limitations that restrict the platform support to a subset of all NetFlow fields. Specifically, TCP Flags and the ToS byte (DSCP) are not supported.

Configuring NetFlow Statistics Collection

To configure NetFlow switching, complete the tasks in these sections:

- Checking for Required Hardware, page 46-6
- Enabling NetFlow Statistics Collection, page 46-7
- Configuring Switched/Bridged IP Flows, page 46-8
- Exporting NetFlow Statistics, page 46-9
- Managing NetFlow Statistics Collection, page 46-9
- Configuring an Aggregation Cache, page 46-10
- Configuring a NetFlow Minimum Prefix Mask for Router-Based Aggregation, page 46-11
- Configuring NetFlow Aging Parameters, page 46-12

Checking for Required Hardware

To ensure that the necessary hardware is enabled, enter the **show module** command, as follows:

```
Switch# show module all
Chassis Type : WS-C4507R

Power consumed by backplane : 40 Watts

Mod Ports Card Type                              Model
Serial No.
---+-----+--------------------------------------+------------------+-----------
1     2  1000BaseX (GBIC) Supervisor(active)    WS-X4515
JAB062604KB
2     2  1000BaseX (GBIC) Supervisor(standby)   WS-X4515
JAB062408CB
6    48  10/100BaseTX (RJ45)                    WS-X4148
JAB032305UH
```

```
M MAC addresses                    Hw  Fw           Sw               Status
--+--------------------------------+---+------------+----------------+---------
1 0001.6442.2c00 to 0001.6442.2c01 0.4 12.1(14r)EW( 12.1(20030513:00 Ok
2 0001.6442.2c02 to 0001.6442.2c03 0.4 12.1(14r)EW( 12.1(20030513:00 Ok
6 0050.3ed8.6780 to 0050.3ed8.67af 1.6 12.1(14r)EW( 12.1(20030513:00 Ok

Mod  Submodule              Model            Serial No.   Hw   Status
----+-----------------------+----------------+------------+----+---------
1   Netflow Services Card   WS-F4531          JAB062209CG  0.2  Ok
2   Netflow Services Card   WS-F4531          JAB062209AG  0.2  Ok

Switch#
```

Note Enabling this feature does not impact the hardware-forwarding performance of the switch.

The effective size of the hardware flow cache table is 65,000 flows. (The hardware flow cache for the Supervisor Engine V-10GE is 85,000 flows.) If more than 85,000 flows are active simultaneously, statistics may be lost for some of the flows.

The effective size of the software flow table is 256, 000 flows. The NetFlow software manages the consistency between the hardware and software tables, keeping the hardware table open by purging inactive hardware flows to the software table.

User-configured timeout settings dictate when the flows are purged and exported through NDE from the software cache. Hardware flow management ensures consistency between hardware flow purging and the user-configured timeout settings.

Software-forwarded flows are also monitored. Moreover, statistics overflow if any flow receives traffic at a sustained rate exceeding 2 gigabits per second. Generally, this situation should not occur because a port cannot transmit at a rate higher than 1 gigabit per second.

Note By design, even if the timeout settings are high, flows automatically “age out” as they approach their statistics limit.

Enabling NetFlow Statistics Collection

Note NetFlow Flow Statistics are disabled by default.

To enable NetFlow switching, first configure the switch for IP routing as described in the IP configuration chapters in the *Cisco IOS IP and IP Routing Configuration Guide*. After you configure IP routing, perform one of these tasks:

Command	Purpose
`Switch(config)#` **`ip flow ingress`**	Enables NetFlow for IP routing.
`Switch(config)#` **`ip flow ingress infer-fields`**	Enables NetFlow with inferred input/output interfaces and source/destination BGP as information. The **inter-fields** option must be configured for AS information to be determined.

Configuring Switched/Bridged IP Flows

Netflow is defined as a collection of routed IP flows created and tracked for all routed IP traffic. In switching environments, considerable IP traffic is switched within a VLAN and hence is not routed. This traffic is termed *switched/bridged IP traffic*; the associated flow is termed *switched/bridged IP flows*. NetFlow hardware is capable of creating and tracking this type of flow. The NetFlow Switched IP Flows feature enables you to create, track, and export switched IP flows (that is, it creates and tracks flows for IP traffic that is being switched and not routed).

Be aware of the following:

- Switched IP flow collection cannot be enabled in isolation on Catalyst 4500 series switches. You need to enable both routed flow and switched flow collection to start collecting switched IP flows.
- Generally, the input and output interface information are NULL. If the traffic is being switched on a VLAN that is associated with an SVI, the input and output interface information points to the same Layer 3 interface.
- Switched flows are exported according to regular export configurations; a separate export CLI does not exist.
- In the main cache, switched IP flows and routed IP flows are indistinguishable; this is due to a hardware limitation.

Note To enable switched IP flow collection on all interfaces, you need to enter both the **ip flow ingress** and **ip flow ingress layer2-switched** commands.

Note To enable a user-based rate limiting policy on the switched IP flow traffic, you need to enter the **ip flow ingress layer2-switched** command, but not the **ip flow ingress** command. (See “Configuring User Based Rate Limiting” on page 42.

To configure the NetFlow cache and enable switched IP flow collection, perform this task:

	Command	Purpose
Step 1	`Switch# conf terminal`	Enter configuration mode.
Step 2	`Switch(config)# ip flow ingress`	Enable routed flow collection.
Step 3	`Switch(config)# ip flow ingress layer2-switched`	Enable switched flow collection.

This example shows how to display the contents of an IP flow cache that contains switch IP flows:

```
Switch# show ip cache flow
IP Flow Switching Cache, 17826816 bytes
 2 active, 262142 inactive, 2 added
 6 ager polls, 0 flow alloc failures
 Active flows timeout in 30 minutes
 Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 1081480 bytes
 2 active, 65534 inactive, 2 added, 2 added to flow
 0 alloc failures, 0 force free
 1 chunk, 1 chunk added
 last clearing of statistics never
```

```
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Fa1           150.1.1.1       Fa1           13.1.1.1        11 003F 003F   425K
Fa1           13.1.1.1        Fa1           150.1.1.1       11 003F 003F   425K
Switch#
```

Exporting NetFlow Statistics

To configure the switch to export NetFlow Statistics to a workstation when a flow expires, perform one of these tasks:

Command	Purpose
`Switch(config)# ip flow-export destination {hostname | ip-address} udp-port`	(Required) Configures the switch to export NetFlow cache entries to a specific destination (for example, a workstation). **Note** You can specify multiple destinations.
`Switch(config)# ip flow-export version {1 | {5 [origin-as | peer-as]}}`	(Optional) Configures the switch to export NetFlow cache entries to a workstation if you are using receiving software that requires version 1 or 5. Version 1 is the default. **origin-as** causes NetFlow to determine the origin BGP autonomous system of both the source and the destination hosts of the flow. **peer-as** causes NetFlow to determine the peer BGP autonomous system of both the input and output interfaces of the flow.
`Switch(config)# ip flow-export source <interface>`	(Optional) Specifies an interface whose IP address is used as the source IP address in the IP header of the NetFlow Data Export (NDE) packet. Default is the NDE output interface.

Managing NetFlow Statistics Collection

You can display and clear NetFlow Statistics, including IP flow switching cache information and flow information, such as the protocol, total flow, flows per second, and so forth. You can also use the resulting information to obtain information about your switch traffic.

To manage NetFlow switching statistics, perform one or both of following tasks:

Command	Purpose
`Switch# show ip cache flow`	Displays the NetFlow switching statistics.
`Switch# clear ip flow stats`	Clears the NetFlow switching statistics.

Configuring an Aggregation Cache

Aggregation of NetFlow Statistics is typically performed by NetFlow collection tools on management workstations. By extending this support to the Catalyst 4500 series switch, you can do the following:

- Reduce the required bandwidth between the switch and workstations, because fewer NDE packets are exported.
- Reduce the number of collection workstations required.
- Provide visibility to aggregated flow statistics at the CLI.

To configure an aggregation cache, you must enter the aggregation cache configuration mode, and you must decide which type of aggregation scheme you would like to configure: autonomous system, destination prefix, protocol prefix, or source prefix aggregation cache. Once you define the aggregation scheme, define the operational parameters for that scheme. More than one aggregation cache can be configured concurrently.

To configure an aggregation cache, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`ip flow-aggregation cache as`**	Enters aggregation cache configuration mode and enables an aggregation cache scheme (autonomous system, destination-prefix, prefix, protocol-port, or source-prefix).

Step 2	`Router(config-flow-cache)#`
	`cache timeout inactive 199`
		Specifies the number of seconds (in this example, 199) in which an inactive entry is allowed to remain in the aggregation cache before it is deleted.

Step 3	`Router(config-flow-cache)#`
	`cache timeout active 45`
		Specifies the number of minutes (in this example, 45) in which an active entry is active.

Step 4	`Router(config-flow-cache)#`
	`export destination 10.42.41.1 9991`
		Enables the data export.

Step 5	`Router(config-flow-cache)#` **`enabled`**	Enables aggregation cache creation.

Verifying Aggregation Cache Configuration and Data Export

To verify the aggregation cache information, perform this task:

Command	Purpose
`Router#` **`show ip cache flow aggregation destination-prefix`**	Displays the specified aggregation cache information.

To confirm data export, perform the following task:

Command	Purpose
`Router#` **`show ip flow export`**	Displays the statistics for the data export including the main cache and all other enabled caches.

Configuring a NetFlow Minimum Prefix Mask for Router-Based Aggregation

The minimum prefix mask specifies the shortest subnet mask that is used for aggregating flows within one of the IP-address based aggregation caches (e.g. source-prefix, destination-prefix, prefix). In these caches, flows are aggregated based upon the IP address (source, destination, or both, respectively) and masked by the longer of the Minimum Prefix mask and the subnet mask of the route to the source/destination host of the flow (as found in the switch routing table).

Note The default value of the minimum mask is zero. The configurable range for the minimum mask is from 1 to 32. You should chose an appropriate value depending on the traffic. A higher value for the minimum mask provides more detailed network addresses, but it may also result in increased number of flows in the aggregation cache.

To configure a minimum prefix mask for the Router-Based Aggregation feature, perform the tasks described in the following sections. Each task is optional.

- Configuring the Minimum Mask of a Prefix Aggregation Scheme
- Configuring the Minimum Mask of a Destination-Prefix Aggregation Scheme
- Configuring the Minimum Mask of a Source-Prefix Aggregation Scheme
- Monitoring and Maintaining Minimum Masks for Aggregation Schemes

Configuring the Minimum Mask of a Prefix Aggregation Scheme

To configure the minimum mask of a prefix aggregation scheme, perform this task:

	Command	Purpose
Step 1	Router(config)# **ip flow-aggregation cache prefix**	Configures the prefix aggregation cache.
Step 2	Router(config-flow-cache)# **mask source minimum** *value*	Specifies the minimum value for the source mask.
Step 3	Router(config-flow-cache)# **mask destination minimum** *value*	Specifies minimum value for the destination mask.

Configuring the Minimum Mask of a Destination-Prefix Aggregation Scheme

To configure the minimum mask of a destination-prefix aggregation scheme, perform this task:

	Command	Purpose
Step 1	Router(config)# **ip flow-aggregation cache destination-prefix**	Configures the destination aggregation cache.
Step 2	Router(config-flow-cache)# **mask destination minimum** *value*	Specifies the minimum value for the destination mask.

Configuring the Minimum Mask of a Source-Prefix Aggregation Scheme

To configure the minimum mask of a source-prefix aggregation scheme, perform this task:

	Command	Purpose
Step 1	`Router(config)# ip flow-aggregation cache source-prefix`	Configures the source-prefix aggregation cache.
Step 2	`Router(config-flow-cache)# mask source minimum value`	Specifies the minimum value for the source mask.

Monitoring and Maintaining Minimum Masks for Aggregation Schemes

To view the configured value of the minimum mask, use the following commands for each aggregation scheme, as needed:

Command	Purpose
`Router# show ip cache flow aggregation prefix`	Displays the configured value of the minimum mask in the prefix aggregation scheme.
`Router# show ip cache flow aggregation destination-prefix`	Displays the configured value of the minimum mask in the destination-prefix aggregation scheme.
`Router# show ip cache flow aggregation source-prefix`	Displays the configured value of the minimum mask in the source-prefix aggregation scheme.

Configuring NetFlow Aging Parameters

You can control when flows are purged from the software flow cache (and, if configured, reported through NDE) with the configuration aging parameters, **Active** and **Inactive**, of the **ip flow-cache timeout** command.

Active Aging specifies the period of time in which a flow should be removed from the software flow cache after the flow is created. Generally, this parameter is used to periodically notify external collection devices about active flows. This parameter operates independently of existing traffic on the flow. Active timeout settings tend to be on the order of minutes (default is 30min).

Inactive Aging specifies how long to wait before removing a flow after the last packet is seen. The Inactive parameter clears the flow cache of “stale” flows thereby preventing new flows from starving (due to lack of resources). Inactive timeout settings tend to be on the order of seconds (default is 15sec).

NetFlow Statistics Collection Configuration Example

The following example shows how to modify the configuration to enable NetFlow switching. It also shows how to export the flow statistics for further processing to UDP port 9991 on a workstation with the IP address of 40.0.0.2. In this example, existing NetFlow Statistics are cleared, thereby ensuring that the **show ip cache flow** command displays an accurate summary of the NetFlow switching statistics:

```
Switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip route-cache flow
Switch(config)# ip flow-export destination 40.0.0.2 9991
Switch(config)# ip flow-export version 5
Switch(config)# end
Switch# show ip flow export
Flow export is enabled
  Exporting flows to 40.0.0.2 (9991)
  Exporting using source IP address 40.0.0.1
  Version 5 flow records
  2 flows exported in 1 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
  0 export packets were dropped due to fragmentation failures
  0 export packets were dropped due to encapsulation fixup failures
Switch#

Switch# show ip cache flow

IP Flow Switching Cache, 17826816 bytes
  69 active, 262075 inactive, 15087 added
  4293455 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 1081480 bytes
  0 active, 65536 inactive, 0 added, 0 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet          28      0.0       167    40      0.0      20.9      11.9
TCP-other          185      0.0         2    48      0.0       6.2      15.4
UDP-DNS              4      0.0         1    61      0.0       0.0      15.5
UDP-other        13466      0.0   3396586    46  91831.3     139.3      15.9
ICMP                97      0.0         2    95      0.0       2.3      15.4
IGMP                 1      0.0         2    40      0.0       0.9      15.1
IP-other          1120      0.0  38890838    46  87453.0    1354.5      24.0
Total:           14901      0.0   5992629    46 179284.3     227.8      16.5

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Gi6/2         30.20.1.18      Gi6/1         30.10.1.18      11 4001 4001   537K
Gi6/2         30.20.1.19      Gi6/1         30.10.1.19      11 4001 4001   537K
Gi6/2         30.20.1.16      Gi6/1         30.10.1.16      11 4001 4001   537K
Gi6/2         30.20.1.17      Gi6/1         30.10.1.17      11 4001 4001   537K
Gi6/2         30.20.1.20      Gi6/1         30.10.1.20      11 4001 4001   537K
```

```
Gi6/2         30.20.1.10      Gi6/1         30.10.1.10      11 4001 4001   539K
Gi6/2         30.20.1.11      Gi6/1         30.10.1.11      11 4001 4001   539K
Gi6/2         30.20.1.14      Gi6/1         30.10.1.14      11 4001 4001   539K
Gi6/2         30.20.1.15      Gi6/1         30.10.1.15      11 4001 4001   539K
Gi6/2         30.20.1.12      Gi6/1         30.10.1.12      11 4001 4001   539K
Gi6/2         30.20.1.13      Gi6/1         30.10.1.13      11 4001 4001   539K
Gi5/48        171.69.23.149   Local         172.20.64.200   06 8214 0017   759
Gi6/1         30.10.1.12      Gi6/2         30.20.1.12      11 4001 4001   539K
Gi6/1         30.10.1.13      Gi6/2         30.20.1.13      11 4001 4001   539K
Gi6/1         30.10.1.14      Gi6/2         30.20.1.14      11 4001 4001   539K
Gi6/1         30.10.1.15      Gi6/2         30.20.1.15      11 4001 4001   539K
Gi6/1         30.10.1.10      Gi6/2         30.20.1.10      11 4001 4001   539K
Gi6/1         30.10.1.11      Gi6/2         30.20.1.11      11 4001 4001   539K
Gi6/1         30.10.1.20      Gi6/2         30.20.1.20      11 4001 4001   537K
Gi6/1         30.10.1.16      Gi6/2         30.20.1.16      11 4001 4001   537K
Gi6/1         30.10.1.17      Gi6/2         30.20.1.17      11 4001 4001   537K
Gi6/1         30.10.1.18      Gi6/2         30.20.1.18      11 4001 4001   537K
Gi6/1         30.10.1.19      Gi6/2         30.20.1.19      11 4001 4001   537K
Switch#
```

NetFlow Configuration Examples

This section provides the following basic configuration examples:

- Sample NetFlow Enabling Schemes, page 46-14
- Sample NetFlow Aggregation Configurations, page 46-14
- Sample NetFlow Minimum Prefix Mask Router-Based Aggregation Schemes, page 46-16

Sample NetFlow Enabling Schemes

Note Enabling NetFlow on a per interface basis is not supported on a Catalyst 4500 switch.

This example shows how to enable NetFlow globally:

```
Switch# configure terminal
Switch(config)# ip flow ingress
```

This example shows how to enable NetFlow with support for inferred fields:

```
Switch# configure terminal
Switch(config)# ip flow ingress infer-fields
```

Sample NetFlow Aggregation Configurations

This section provides the following aggregation cache configuration examples:

- Autonomous System Configuration, page 46-15
- Destination Prefix Configuration, page 46-15
- Prefix Configuration, page 46-15
- Protocol Port Configuration, page 46-15
- Source Prefix Configuration, page 46-15

Autonomous System Configuration

This example shows how to configure an autonomous system aggregation cache with an inactive timeout of 200 seconds, a cache active timeout of 45 minutes, an export destination IP address of 10.42.42.1, and a destination port of 9992:

```
Switch(config)# ip flow-aggregation cache as
Switch(config-flow-cache)# cache timeout inactive 200
Switch(config-flow-cache)# cache timeout active 45
Switch(config-flow-cache)# export destination 10.42.42.1 9992
Switch(config-flow-cache)# enabled
```

Destination Prefix Configuration

This example shows how to configure a destination prefix aggregation cache with an inactive timeout of 200 seconds, a cache active timeout of 45 minutes, an export destination IP address of 10.42.42.1, and a destination port of 9992:

```
Switch(config)# ip flow-aggregation cache destination-prefix
Switch(config-flow-cache)# cache timeout inactive 200
Switch(config-flow-cache)# cache timeout active 45
Switch(config-flow-cache)# export destination 10.42.42.1 9992
Switch(config-flow-cache)# enabled
```

Prefix Configuration

This example shows how to configure a prefix aggregation cache with an inactive timeout of 200 seconds, a cache active timeout of 45 minutes, an export destination IP address of 10.42.42.1, and a destination port of 9992:

```
Switch(config)# ip flow-aggregation cache prefix
Switch(config-flow-cache)# cache timeout inactive 200
Switch(config-flow-cache)# cache timeout active 45
Switch(config-flow-cache)# export destination 10.42.42.1 9992
Switch(config-flow-cache)# enabled
```

Protocol Port Configuration

This example shows how to configure a protocol port aggregation cache with an inactive timeout of 200 seconds, a cache active timeout of 45 minutes, an export destination IP address of 10.42.42.1, and a destination port of 9992:

```
Switch(config)# ip flow-aggregation cache protocol-port
Switch(config-flow-cache)# cache timeout inactive 200
Switch(config-flow-cache)# cache timeout active 45
Switch(config-flow-cache)# export destination 10.42.42.1 9992
Switch(config-flow-cache)# enabled
```

Source Prefix Configuration

This example shows how to configure a source prefix aggregation cache with an inactive timeout of 200 seconds, a cache active timeout of 45 minutes, an export destination IP address of 10.42.42.1, and a destination port of 9992:

```
Switch(config)# ip flow-aggregation cache source-prefix
Switch(config-flow-cache)# cache timeout inactive 200
Switch(config-flow-cache)# cache timeout active 45
Switch(config-flow-cache)# export destination 10.42.42.1 9992
Switch(config-flow-cache)# enabled
```

Sample NetFlow Minimum Prefix Mask Router-Based Aggregation Schemes

This section provides examples for the NetFlow minimum prefix mask aggregation cache configuration:

- Prefix Aggregation Scheme
- Destination-Prefix Aggregation Scheme
- Source-Prefix Aggregation Scheme

Prefix Aggregation Scheme

This is an example of a prefix aggregation cache configuration:

```
!
ip flow-aggregation cache prefix
mask source minimum 24
mask destination minimum 28
```

In this example, assume the following configuration:

```
ip route 118.42.20.160 255.255.255.224 110.42.13.2
ip route 122.16.93.160 255.255.255.224 111.22.21.2
```

Both routes have a 27-bit subnet mask in the routing table on the switch.

Flows travelling from the 118.42.20.160 subnet to the 122.16.93.160 subnet whose source IP addresses match with a mask of 27 bits and whose destination IP addresses match with a mask of 28 bits are aggregated together in the cache statistics.

Destination-Prefix Aggregation Scheme

This is an example of a destination-prefix aggregation cache configuration:

```
!
ip flow-aggregation cache destination-prefix
mask destination minimum 32
!
```

Source-Prefix Aggregation Scheme

This is an example of a source-prefix aggregation cache configuration:

```
ip flow-aggregation cache source-prefix
mask source minimum 30
```

CHAPTER 47

Configuring RMON

This chapter describes how to configure Remote Network Monitoring (RMON) on your Catalyst 4500 series switch. RMON is a standard monitoring specification that defines a set of statistics and functions that can be exchanged between RMON-compliant console systems and network probes. RMON provides you with comprehensive network-fault diagnosis, planning, and performance-tuning information.

Note For complete syntax and usage information for the commands used in this chapter, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.4*.

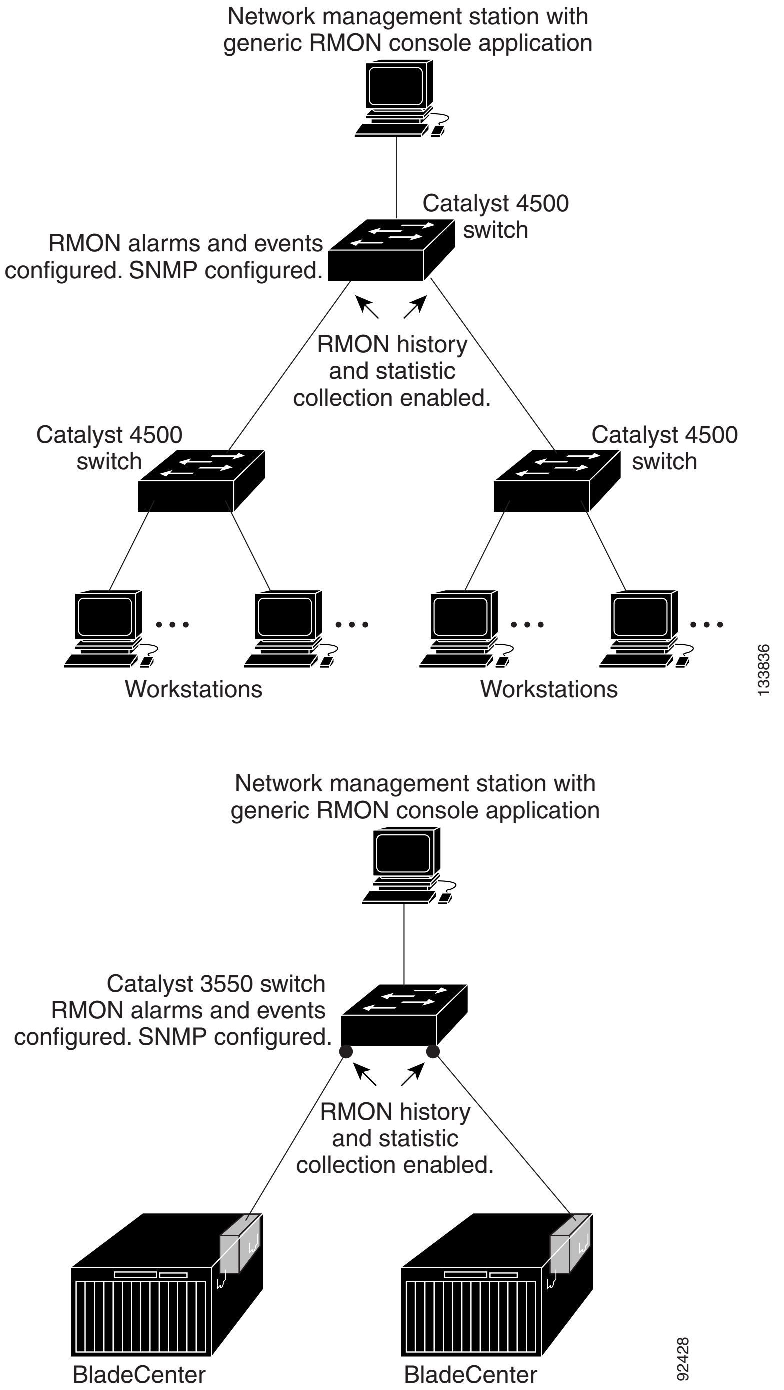
http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124tcr/tcf_r/index.htm

This chapter consists of these sections:

Understanding RMON

RMON is an Internet Engineering Task Force (IETF) standard monitoring specification that allows various network agents and console systems to exchange network monitoring data. You can use the RMON feature with the Simple Network Management Protocol (SNMP) agent in the switch to monitor all the traffic flowing among switches on all connected LAN segments.

Figure 47-1 Remote Monitoring Example


Network management station with generic RMON console application
Catalyst 4500 switch
RMON alarms and events configured. SNMP configured.
RMON history and statistic collection enabled.
Catalyst 4500 switch
Catalyst 4500 switch
Workstations
Workstations
133836
Network management station with generic RMON console application
Catalyst 3550 switch RMON alarms and events configured. SNMP configured.
RMON history and statistic collection enabled.
BladeCenter
BladeCenter
92428

The switch supports these RMON groups (defined in RFC 1757):

- Statistics (RMON group 1)—Collects Ethernet, Fast Ethernet, and Gigabit Ethernet statistics on an interface.
- History (RMON group 2)—Collects a history group of statistics on Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces for a specified polling interval.

- Alarm (RMON group 3)—Monitors a specific MIB object for a specified interval, triggers an alarm at a specified value (rising threshold), and resets the alarm at another value (falling threshold). Alarms can be used with events; the alarm triggers an event, which can generate a log entry or an SNMP trap.
- Event (RMON group 9)—Determines the action to take when an event is triggered by an alarm. The action can be to generate a log entry or an SNMP trap.

Because switches supported by Cisco IOS Release 12.2(31)SG use hardware counters for RMON data processing, the monitoring is more efficient, and little processing power is required.

Configuring RMON

This section describes how to configure RMON on your switch. It contains this configuration information:

- Default RMON Configuration, page 47-3
- Configuring RMON Alarms and Events, page 47-4
- Configuring RMON Collection on an Interface, page 47-5

Default RMON Configuration

RMON is disabled by default; no alarms or events are configured.

Only RMON 1 is supported on the switch.

Configuring RMON Alarms and Events

You can configure your switch for RMON by using the command-line interface (CLI) or an SNMP-compatible network management station. We recommend that you use a generic RMON console application on the network management station (NMS) to take advantage of RMON's network management capabilities. You must also configure SNMP on the switch to access RMON MIB objects. For more information, see Chapter 30, "Configuring SNMP."

To enable RMON alarms and events, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**rmon alarm** *number variable interval* {**absolute** | **delta**} **rising-threshold** *value* [*event-number*] **falling-threshold** *value* [*event-number*] [**owner** *string*]	Sets an alarm on a MIB object. • For *number*, specify the alarm number. The range is 1 to 65535. • For *variable*, specify the MIB object to monitor. • For *interval*, specify the time in seconds the alarm monitors the MIB variable. The range is 1 to 4294967295 seconds. • Specify the **absolute** keyword to test each MIB variable directly; specify the **delta** keyword to test the change between samples of a MIB variable. • For *value*, specify a number at which the alarm is triggered and one for when the alarm is reset. The range for the rising threshold and falling threshold *values* is -2147483648 to 2147483647. • (Optional) For *event-number*, specify the event number to trigger when the rising or falling threshold exceeds its limit. • (Optional) For **owner** *string*, specify the owner of the alarm.
Step 3	**rmon event** *number* [**description** *string*] [**log**] [**owner** *string*] [**trap** *community*]	Adds an event in the RMON event table that is associated with an RMON event number. • For *number*, assign an event number. The range is 1 to 65535. • (Optional) For **description** *string*, specify a description of the event. • (Optional) Use the **log** keyword to generate an RMON log entry when the event is triggered. • (Optional) For **owner** *string*, specify the owner of this event. • (Optional) For *community*, enter the SNMP community string used for this trap.

	Command	Purpose
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable an alarm, use the **no rmon alarm** *number* global configuration command on each alarm you configured. You cannot disable at once all the alarms that you configured. To disable an event, use the **no rmon event** *number* global configuration command. To learn more about alarms and events and how they interact with each other, see RFC 1757.

You can set an alarm on any MIB object. The following example configures RMON alarm number 10 by using the **rmon alarm** command. The alarm monitors the MIB variable *ifEntry.20.1* once every 20 seconds until the alarm is disabled and checks the change in the variable's rise or fall. If the *ifEntry.20.1* value shows a MIB counter increase of 15 or more, such as from 100000 to 100015, the alarm is triggered. The alarm in turn triggers event number 1, which is configured with the **rmon event** command. Possible events can include a log entry or an SNMP trap. If the *ifEntry.20.1* value changes by 0, the alarm is reset and can be triggered again.

```
Switch(config)# rmon alarm 10 ifEntry.20.1 20 delta rising-threshold 15 1 
falling-threshold 0 owner jjohnson
```

The following example creates RMON event number 1 by using the **rmon event** command. The event is defined as *High ifOutErrors* and generates a log entry when the event is triggered by the alarm. The user *jjones* owns the row that is created in the event table by this command. This example also generates an SNMP trap when the event is triggered.

```
Switch(config)# rmon event 1 log trap eventtrap description "High ifOutErrors" owner 
jjones
```

Configuring RMON Collection on an Interface

You must first configure RMON alarms and events to display collection information.

To collect group history statistics on an interface, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Specifies the interface on which to collect history, and enter interface configuration mode.

	Command	Purpose
Step 3	**rmon collection history** *index* [**buckets** *bucket-number*] [**interval** *seconds*] [**owner** *ownername*]	Enables history collection for the specified number of buckets and time period. • For *index*, identify the RMON group of statistics The range is 1 to 65535. • (Optional) For **buckets** *bucket-number,* specify the maximum number of buckets desired for the RMON collection history group of statistics. The range is 1 to 65535. The default is 50 buckets. • (Optional) For **interval** *seconds*, specify the number of seconds in each polling cycle. • (Optional) For **owner** *ownername*, enter the name of the owner of the RMON group of statistics. To disable history collection, use the **no rmon collection history** *index* interface configuration command.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**show rmon history**	Displays the contents of the switch history table.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To collect group Ethernet statistics on an interface, perform this task:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Specifies the interface on which to collect statistics, and enter interface configuration mode.
Step 3	**rmon collection stats** *index* [**owner** *ownername*]	Enables RMON statistic collection on the interface. • For *index*, specify the RMON group of statistics. The range is from 1 to 65535. • (Optional) For **owner** *ownername*, enter the name of the owner of the RMON group of statistics. To disable the collection of group Ethernet statistics, use the **no rmon collection stats** *index* interface configuration command.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entries.
Step 6	**show rmon statistics**	Displays the contents of the switch statistics table.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Displaying RMON Status

To display the RMON status, use one or more of the privileged EXEC commands in Table 47-1:

Table 47-1 Commands for Displaying RMON Status

Command	Purpose
show rmon	Displays general RMON statistics.
show rmon alarms	Displays the RMON alarm table.
show rmon events	Displays the RMON event table.
show rmon history	Displays the RMON history table.
show rmon statistics	Displays the RMON statistics table.

CHAPTER 48

Performing Diagnostics

Diagnostics tests and verifies the functionality of the hardware components of your system (chassis, supervisor engines, modules, and ASICs), while your Catalyst 4500 series switch is connected to a live network. Diagnostics consists of packet switching tests that test hardware components and verify the data path and control signals. Diagnostic tests are non-disruptive (except POST) and run at different times. Some tests run continuously in the background to monitor the status of your system (such as the test for switching modules), while others run only once.

This chapter describes the following types of diagnostics on the Catalyst 4500 series switch:

- Online Diagnostics, page 48-1
- Power-On-Self-Test Diagnostics, page 48-3

Note For complete syntax and usage information for the switch commands used in this chapter, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122sr/cr/index.htm

Online Diagnostics

An online diagnostic test verifies that all ports on a linecard are working correctly. The test can detect whether or not the path to the front panel port on the linecard is broken, but it cannot indicate where along the path the problem occurred.

The test is termed *online* because it runs when your system is running.

Note This test is run only for linecards that have stub chips.

Online diagnostics runs on linecards only once, when they are booting.

This can happen when you insert a linecard or power up a chassis.

Online diagnostics are performed by sending a packet from the CPU to every port on the linecard. Because this packet is marked *loopback*, the CPU expects to see this packet return from the port. The packet first traverses the ASICs on the supervisor engine card, then travels via the chassis backplane and the stub chip on the linecards to the PHYs. The PHY sends it back down the same path.

Note The packet does not reach or exit the front panel port.

Troubleshooting with Online Diagnostics

A faulty linecard occurs if any of the following conditions occurs.

- All ports fail
- All ports on a stub chip fail
- Only one port fails

For all of the above situations, the output of the **show module** command would display the status of the linecard as faulty:

```
Switch# show mod
Chassis Type : WS-C4507R
Power consumed by backplane : 40 Watts

Mod Ports Card Type                              Model              Serial No.
---+-----+--------------------------------------+------------------+-----------
 1     6  Sup II+10GE 10GE (X2), 1000BaseX (SFP) WS-X4013+10GE      JAB091502G0
 2     6  Sup II+10GE 10GE (X2), 1000BaseX (SFP) WS-X4013+10GE      JAB091502FC
 3    48  100BaseX (SFP)                         WS-X4248-FE-SFP    JAB093305RP
 4    48  10/100BaseTX (RJ45)V                   WS-X4148-RJ45V     JAE070717E5
 5    48  10/100BaseTX (RJ45)V                   WS-X4148-RJ45V     JAE061303U3
 6    48  10/100BaseTX (RJ45)V                   WS-X4148-RJ45V     JAE061303WJ
 7    24  10/100/1000BaseT (RJ45)V, Cisco/IEEE   WS-X4524-GB-RJ45V  JAB0815059Q

 M MAC addresses                    Hw  Fw           Sw               Status
--+--------------------------------+---+------------+----------------+---------
 1 000b.5f27.8b80 to 000b.5f27.8b85 0.2 12.2(27r)SG( 12.2(37)SG Ok
 2 000b.5f27.8b86 to 000b.5f27.8b8b 0.2 12.2(27r)SG( 12.2(37)SG Ok
 3 0005.9a80.6810 to 0005.9a80.683f 0.4                               Ok
 4 000c.3016.aae0 to 000c.3016.ab0f 2.6                               Ok
 5 0008.a3a3.4e70 to 0008.a3a3.4e9f 1.6                               Ok
 6 0008.a3a3.3fa0 to 0008.a3a3.3fcf 1.6                                 Faulty
 7 0030.850e.3e78 to 0030.850e.3e8f 1.0                               Ok

Mod  Redundancy role     Operating mode      Redundancy status
----+-------------------+-------------------+----------------------------------
 1   Active Supervisor   SSO                 Active
 2   Standby Supervisor  SSO                 Standby hot
```

To troubleshoot a faulty linecard, do the following:

Step 1 Enter the command **show diagnostic result module 3**.

If a faulty linecard was inserted in the chassis, it would have failed diagnostics and the output would be similar to the following:

```
Diagnostic[module 3]: Diagnostic handle is not found for the card.

module 3:

  Overall diagnostic result: PASS

  Test results: (. = Pass, F = Fail, U = Untested)

    1) linecard-online-diag --------------------> F
```

RMA the linecard, contact TAC, and skip steps 2 & 3.

However, if the output shows the following:

```
module 3:
 Overall diagnostic result: PASS

Test results: (. = Pass, F = Fail, U = Untested)

1) linecard-online-diag --------------------> .
```

The linecard passed online diagnostics either 1) when it was inserted into the chassis the last time or 2) when the switch was powered up (as reported by the "."). Further investigation is required.

Step 2 Insert a different supervisor engine card and re-insert the linecard.

If the linecard passes the test, it suggests that the supervisor engine card is defective.

RMA the supervisor engine, contact TAC, and skip step 3.

Because online diagnostics is not run on the supervisor engine card(s), so you cannot use the **#show diagnostic module 1** command to test whether the supervisor engine card is faulty.

Step 3 Re-insert the linecard in a different chassis.

If the linecard passes the test, the problem is associated with the chassis.

RMA the chassis and contact TAC.

Power-On-Self-Test Diagnostics

The following topics are discussed:

- Overview, page 48-3
- Sample POST Results, page 48-4
- Power-On-Self-Test Results for Supervisor Engine V-10GE, page 48-8
- Causes of Failure and Troubleshooting, page 48-14

Overview

All Catalyst 4500 series switches have power-on-self-test (POST) diagnostics that run whenever a supervisor engine boots. POST tests the basic hardware functionality for the supervisor switching engine, its associated packet memory and other on board hardware components. The results of POST impacts how the switch boots, as the health of the supervisor engine is critical to the operation of the switch. The switch might boot in a marginal or faulty state.

POST is currently supported on the following supervisor engines:

- WS-X4014
- WS-X4515
- WS-X4516
- WS-X4516-10GE
- WS-X4013+
- WS-X4013+TS
- WS-X4013+10GE
- WS-C4948G

- WS-C4948G-10GE
- ME-4924-10GE
- WS-X45-SUP6-E

The POST results are indicated with a '.' or a ‘Pass’ for Pass, an 'F' for a ‘Fail’ and a 'U' for Untested.

Sample POST Results

For all the supervisor engines, POST performs CPU, traffic, system, system memory, and feature tests.

For CPU tests, POST verifies appropriate activity of the supervisor SEEPROM, temperature sensor, and Ethernet-end-of-band channel (eobc), when used.

The following example illustrates the output of a CPU subsystem test on all supervisor engines except the WS-X4013+TS:

```
[..]
Cpu Subsystem Tests ...
seeprom: . temperature_sensor: . eobc: .
[..]
```

The following example illustrates the output of a CPU subsystem test on a WS-X4013+TS supervisor engine.

```
[..]
Cpu Subsystem Tests ...
seeprom: . temperature_sensor: .
[..]
```

For traffic tests, POST sends packets from the CPU to the switch. These packets loop several times within the switch core and validate the switching, the Layer 2 and the Layer 3 functionality. To isolate the hardware failures accurately, the loop back is done both inside and outside the switch ports.

The following example illustrates the output of a Layer 2 traffic test at the switch ports on the supervisor engines WS-X4516, WS-X4516-10GE, WS-X4013+10GE, WS-C4948G-10GE:

```
Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
```

The following example illustrates the output of a Layer 2 traffic test at the switch ports on the supervisor engines WS-X4013+TS, WS-X4515, WS-X4013+, WS-X4014, WS-C4948G:

```
Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31:
```

POST also performs tests on the packet and system memory of the switch. These are numbered dynamically in ascending order starting with 1 and represent different memories.

The following example illustrates the output from a system memory test:

```
Switch Subsystem Memory ...
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: . 52: . 53: . 54: . 55: .
```

POST also tests the Netflow services card (Supervisor Engine IV and Supervisor Engine V) and the Netflow services feature (Supervisor Engine V -10GE). Failures from these tests are treated as marginal, as they do not impact functionality of the switch (except for the unavailability of the Netflow features):

```
Netflow Services Feature ...
se: . cf: . 52: . 53: . 54: . 55: . 56: . 57: . 58: . 59: . 60: . 61: .
62: . 63: . 64: . 65: .
```

Note Supervisor Engine VI-E retains most of the previous supervisors' POST features including the CPU subsystem tests, Layer 3 and Layer 2 traffic tests, and memory tests. Redundant ports on redundant systems are not tested. All POST diagnostics are local to the supervisor running the tests.

The following example shows the output for a WS-X4516 supervisor engine:

```
Switch# show diagnostic result module 2 detail

module 2:

  Overall diagnostic result: PASS

  Test results: (. = Pass, F = Fail, U = Untested)

  ___________________________________________________________________________

    1) supervisor-bootup -----------------------> .

          Error code --------------------------> 0 (DIAG_SUCCESS)
          Total run count ---------------------> 1
          Last test execution time ------------> Jul 20 2005 14:15:52
          First test failure time -------------> n/a
          Last test failure time --------------> n/a
          Last test pass time -----------------> Jul 20 2005 14:15:52
          Total failure count -----------------> 0
          Consecutive failure count -----------> 0

Power-On-Self-Test Results for ACTIVE Supervisor


Power-on-self-test for Module 2:  WS-X4516
 Port/Test Status: (. = Pass, F = Fail, U = Untested)
 Reset Reason: PowerUp RemoteDebug



Cpu Subsystem Tests ...
seeprom: . temperature_sensor: . eobc: .

Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .


Port Traffic: L2 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
```

```
Port Traffic: L3 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .


Switch Subsystem Memory ...
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: . 52: . 53: . 54: . 55: .


Module 2 Passed


  ___________________________________________________________________________

    2) packet-memory-bootup --------------------> 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
packet buffers on free list: 64557 bad: 0 used for ongoing tests: 979


Exhaustive packet memory tests did not run at bootup.
Bootup test results:5
No  errors.

  ___________________________________________________________________________

    3) packet-memory-ongoing -------------------> 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
packet buffers on free list: 64557 bad: 0 used for ongoing tests: 979


Packet memory errors: 0 0
Current alert level: green
Per 5 seconds in the last minute:
    0 0 0 0 0 0 0 0 0 0
    0 0
Per minute in the last hour:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Per hour in the last day:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0
Per day in the last 30 days:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Direct memory test failures per minute in the last hour:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Potential false positives: 0 0
  Ignored because of rx errors: 0 0
  Ignored because of cdm fifo overrun: 0 0
  Ignored because of oir: 0 0
  Ignored because isl frames received: 0 0
  Ignored during boot: 0 0
  Ignored after writing hw stats: 0 0
  Ignored on high gigaport: 0
Ongoing diag action mode: Normal
Last 1000 Memory Test Failures:
Last 1000 Packet Memory errors:
First 1000 Packet Memory errors:

  ___________________________________________________________________________

Switch#
```

The following example shows the output for a WS-X45-SUP6-E supervisor engine:

```
Switch# show diagnostic result module 3 detail

module 3:   SerialNo : XXXXXXXXXXX

  Overall diagnostic result: PASS

  Test results: (. = Pass, F = Fail, U = Untested)
___________________________________________________________________________

    1) supervisor-bootup --------------->
          Error code ------------------> 0 (DIAG_SUCCESS)
          Total run count -------------> 1
          Last test execution time ----> Oct 01 2007 17:37:04
          First test failure time -----> n/a
          Last test failure time ------> n/a
          Last test pass time ---------> Oct 01 2007 17:37:04
          Total failure count ---------> 0
          Consecutive failure count ---> 0
Power-On-Self-Test Results for ACTIVE Supervisor
prod: WS-X45-SUP6-E part: XXXXXXXXX serial: XXXXXXXXXX
Power-on-self-test for Module 3: WS-X45-SUP6-E
 Test Status: (. = Pass, F = Fail, U = Untested)

CPU Subsystem Tests ...
 seeprom: Pass

Traffic: L3 Loopback ...
 Test Results: Pass
```

```
Traffic: L2 Loopback ...
 Test Results: Pass

Switching Subsystem Memory ...
 Packet Memory Test Results: Pass

Module 3 Passed
___________________________________________________________________________

    2) linecard-online-diag ------------>
          Error code ------------------> 0 (DIAG_SUCCESS)
          Total run count -------------> 1
          Last test execution time ----> Oct 01 2007 17:37:04
          First test failure time -----> n/a
          Last test failure time ------> n/a
          Last test pass time ---------> Oct 01 2007 17:37:04
          Total failure count ---------> 0
          Consecutive failure count ---> 0

Slot Ports Card Type                              Diag Status      Diag Details
---- ----- -------------------------------------- ---------------- ------------
 3     6   Sup 6-E 10GE (X2), 1000BaseX (SFP)     Skipped          Packet memory
Detailed Status
---------------
. = Pass              U = Unknown
L = Loopback failure  S = Stub failure
P = Port failure
E = SEEPROM failure   G = GBIC integrity check failure

Ports 1   2   3   4   5   6
      .   .   .   .   .   .
  ___________________________________________________________________________
Switch#
```

Power-On-Self-Test Results for Supervisor Engine V-10GE

For the Supervisor Engine V-10GE (WS-X4516-10GE), POST tests extra redundancy features on the 10-gigabit ports.

The following topics are discussed:

- POST on the Active Supervisor Engine, page 48-8
- Sample POST Results on an Active Supervisor Engine, page 48-9
- POST on Standby Supervisor Engine, page 48-11
- Sample Display of the POST on Standby Supervisor Engine, page 48-11

POST on the Active Supervisor Engine

The active supervisor engine tests the remote redundant 10-gigabit ports on the standby supervisor engine if it is present when the active supervisor engine is booting. The status of the port is displayed as “Remote TenGigabit Port Status.” If no standby supervisor engine is present, the remote port status is always displayed as “Untested.” It persists even after a new standby supervisor engine is inserted. The remaining tests are conducted using only the gigabit ports’ configuration.

After the active supervisor engine has completed the boot up diagnostics, if the standby supervisor engine is now removed, the remote port status is changed to “Untested” in the overall diagnostic results.

Sample POST Results on an Active Supervisor Engine

```
Switch# show diagnostic result module 1 detail

module 1:

  Overall diagnostic result: PASS

  Test results: (. = Pass, F = Fail, U = Untested)

  ___________________________________________________________________________

    1) supervisor-bootup -----------------------> .

          Error code --------------------------> 0 (DIAG_SUCCESS)
          Total run count ---------------------> 1
          Last test execution time ------------> Jul 19 2005 13:28:16
          First test failure time -------------> n/a
          Last test failure time --------------> n/a
          Last test pass time -----------------> Jul 19 2005 13:28:16
          Total failure count -----------------> 0
          Consecutive failure count -----------> 0

Power-On-Self-Test Results for ACTIVE Supervisor


Power-on-self-test for Module 1:  WS-X4516-10GE
 Port/Test Status: (. = Pass, F = Fail, U = Untested)
 Reset Reason: Software/User



Cpu Subsystem Tests ...
seeprom: . temperature_sensor: . eobc: .

Port Traffic: L3 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .

Local 10GE Port 62: .

Local 10GE Port 63: .

Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
48: . 49: . 50: . 51: .


Port Traffic: L2 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
48: . 49: . 50: . 51: .


Switch Subsystem Memory ...
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
```

```
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: .


Netflow Services Feature ...
se: . cf: . 52: . 53: . 54: . 55: . 56: . 57: . 58: . 59: . 60: . 61: .
62: . 63: . 64: . 65: .


Module 1 Passed

Remote TenGigabitPort status: Passed

  ___________________________________________________________________________

    2) packet-memory-bootup --------------------> 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
packet buffers on free list: 64557 bad: 0 used for ongoing tests: 979


Exhaustive packet memory tests did not run at bootup.
Bootup test results:5
No  errors.

  ___________________________________________________________________________

    3) packet-memory-ongoing -------------------> 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
packet buffers on free list: 64557 bad: 0 used for ongoing tests: 979


Packet memory errors: 0 0
Current alert level: green
Per 5 seconds in the last minute:
    0 0 0 0 0 0 0 0 0 0
    0 0
Per minute in the last hour:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Per hour in the last day:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0
Per day in the last 30 days:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Direct memory test failures per minute in the last hour:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Potential false positives: 0 0
  Ignored because of rx errors: 0 0
  Ignored because of cdm fifo overrun: 0 0
  Ignored because of oir: 0 0
  Ignored because isl frames received: 0 0
  Ignored during boot: 0 0
  Ignored after writing hw stats: 0 0
  Ignored on high gigaport: 0
Ongoing diag action mode: Normal
Last 1000 Memory Test Failures:
Last 1000 Packet Memory errors:
First 1000 Packet Memory errors:

  ___________________________________________________________________________

Switch#
```

POST on Standby Supervisor Engine

Ports 62 and 63 of the supervisor engine always remain Untested or U. Because the Standby supervisor engine never tests the remote 10-gigabit port on the active supervisor engine, the remote 10-gigabit port status on the standby supervisor engine is always Untested. The supervisor engine performs the remaining tests using the gigabit ports' configuration.

Note On redundant chassis, concurrent POST is supported on supervisor engines that are already inserted. However, if a second supervisor engine is inserted while the first one is loading, you might boot the first supervisor engine in a faulty IOS state (POST will abort and some of the POST's tests will be bypassed). This only happens during concurrent bootup of the supervisor engines. So, you should not insert any additional supervisor engines in the empty supervisor engine slot while an already seated supervisor engine is running POST. The Power-On-Self-Test sequence is completed when the "Exiting to ios..." message is displayed.

Sample Display of the POST on Standby Supervisor Engine

```
Switch# show diagnostic result module 2 detail

module 2:

  Overall diagnostic result: PASS

  Test results: (. = Pass, F = Fail, U = Untested)

  ___________________________________________________________________________

    1) supervisor-bootup -----------------------> .
```

```
          Error code --------------------------> 0 (DIAG_SUCCESS)
          Total run count ---------------------> 1
          Last test execution time ------------> Jul 19 2005 13:29:44
          First test failure time -------------> n/a
          Last test failure time --------------> n/a
          Last test pass time -----------------> Jul 19 2005 13:29:44
          Total failure count -----------------> 0
          Consecutive failure count -----------> 0

Power-On-Self-Test Results for ACTIVE Supervisor


Power-on-self-test for Module 2:  WS-X4516-10GE
 Port/Test Status: (. = Pass, F = Fail, U = Untested)
 Reset Reason: OtherSupervisor Software/User




Cpu Subsystem Tests ...
seeprom: . temperature_sensor: . eobc: .

Port Traffic: L3 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .

Local 10GE Port 62: U

Local 10GE Port 63: U

Port Traffic: L2 Serdes Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
48: . 49: . 50: . 51: .


Port Traffic: L2 Asic Loopback ...
 0: .  1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: .
12: . 13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: .
24: . 25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: .
36: . 37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: .
48: . 49: . 50: . 51: .


Switch Subsystem Memory ...
 1: .  2: .  3: .  4: .  5: .  6: .  7: .  8: .  9: . 10: . 11: . 12: .
13: . 14: . 15: . 16: . 17: . 18: . 19: . 20: . 21: . 22: . 23: . 24: .
25: . 26: . 27: . 28: . 29: . 30: . 31: . 32: . 33: . 34: . 35: . 36: .
37: . 38: . 39: . 40: . 41: . 42: . 43: . 44: . 45: . 46: . 47: . 48: .
49: . 50: . 51: .


Netflow Services Feature ...
se: . cf: . 52: . 53: . 54: . 55: . 56: . 57: . 58: . 59: . 60: . 61: .
62: . 63: . 64: . 65: .


Module 2 Passed

Remote TenGigabitPort status: Untested
```

```
  ___________________________________________________________________________

    2) packet-memory-bootup --------------------> 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
packet buffers on free list: 64557 bad: 0 used for ongoing tests: 979


Exhaustive packet memory tests did not run at bootup.
Bootup test results:5
No  errors.

  ___________________________________________________________________________

    3) packet-memory-ongoing -------------------> 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
packet buffers on free list: 64557 bad: 0 used for ongoing tests: 979


Packet memory errors: 0 0
Current alert level: green
Per 5 seconds in the last minute:
    0 0 0 0 0 0 0 0 0 0
    0 0
Per minute in the last hour:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Per hour in the last day:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0
Per day in the last 30 days:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Direct memory test failures per minute in the last hour:
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0
Potential false positives: 0 0
  Ignored because of rx errors: 0 0
  Ignored because of cdm fifo overrun: 0 0
  Ignored because of oir: 0 0
  Ignored because isl frames received: 0 0
  Ignored during boot: 0 0
  Ignored after writing hw stats: 0 0
  Ignored on high gigaport: 0
Ongoing diag action mode: Normal
Last 1000 Memory Test Failures:
Last 1000 Packet Memory errors:
First 1000 Packet Memory errors:

  ___________________________________________________________________________

Switch#
```

Note To ensure that the maximum number of ports are tested, ensure that both supervisor engines are present on power-up.

Causes of Failure and Troubleshooting

A failure of any of the POST tests reflects a problem with the hardware on the supervisor engine. IOS boots the supervisor engine with limited functionality, allowing the user to evaluate and display the diagnostic test results.

To evaluate if the hardware failure is persistent, you can power cycle the supervisor engine to rerun the POST tests.

You can also remove and reinsert the supervisor engine into the chassis to ensure that the seating is correct. Please call the Cisco Systems customer support team for more information.

Note On redundant chassis, concurrent POST is supported on supervisor engines that are already inserted. However, if a second supervisor engine is inserted while the first one is loading, you might boot the first supervisor engine in a faulty IOS state (POST will abort and some of the POST's tests will be bypassed). This only happens during concurrent bootup of the supervisor engines. So, you should not insert any additional supervisor engines in the empty supervisor engine slot while an already seated supervisor engine is running POST. The Power-On-Self-Test sequence is completed when the “Exiting to ios...” message is displayed.

CHAPTER 49

Configuring WCCP Version 2 Services

Note WCCP v2 is *not* supported on Supervisor Engine 6-E.

This chapter describes how to configure the Catalyst 4500 series switches to redirect traffic to content engines (web caches) using the Web Cache Communication Protocol (WCCP) version 2

Note Throughout this chapter, WCCP refers to WCCP version 2. Version 1 is *not* supported.

This chapter consists of these sections:

- Understanding WCCP, page 49-1
- Restrictions for WCCP, page 49-5
- Configuring WCCP, page 49-5
- Verifying and Monitoring WCCP Configuration Settings, page 49-8
- WCCP Configuration Examples, page 49-8

Note The tasks in this chapter assume that you have already configured content engines on your network. For specific information on hardware and network planning associated with Cisco Content Engines and WCCP, see the Product Literature and Documentation links available on the Cisco.com Web Scaling site at this location:

http://www.cisco.com/warp/public/cc/pd/cxsr/ces/index.shtml.

Understanding WCCP

These sections describe WCCP:

- WCCP Overview, page 49-2
- Hardware Acceleration, page 49-2
- Understanding WCCP Configuration, page 49-2
- WCCP Features, page 49-4

WCCP Overview

WCCP is a Cisco-developed content-routing technology that enables you to integrate content engines into your network infrastructure.

The Cisco IOS WCCP feature enables use of Cisco Content Engines (or other content engines 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 of HTTP requests 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 content engine. The word “transparent” is this case means that the end user does not know that a requested file (such as a web page) came from the content engine instead of from the originally specified server.

When a content engine receives a request, it attempts to service it from its own local content. If the requested information is not present, the content engine issues its own request to the originally targeted server to get the required information. When the content 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 content engines, called a content engine cluster, to provide content to a router or multiple routers.Network administrators can easily scale their content engines to handle heavy traffic loads through these clustering capabilities. Cisco clustering technology enables each content member to work in parallel, resulting in linear scalability. Clustering content engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 content engines to scale to your desired capacity.

Hardware Acceleration

Catalyst 4500 series switches provide hardware acceleration for directly connected Cisco Content Engines, which is more efficient than Layer 3 redirection in the software.

You must configure a directly connected Content Engine to negotiate use of the WCCP Layer 2 Redirection feature with load balancing based on the mask assignment table. The **show ip wccp web-cache detail** command displays which redirection method is in use for each cache.

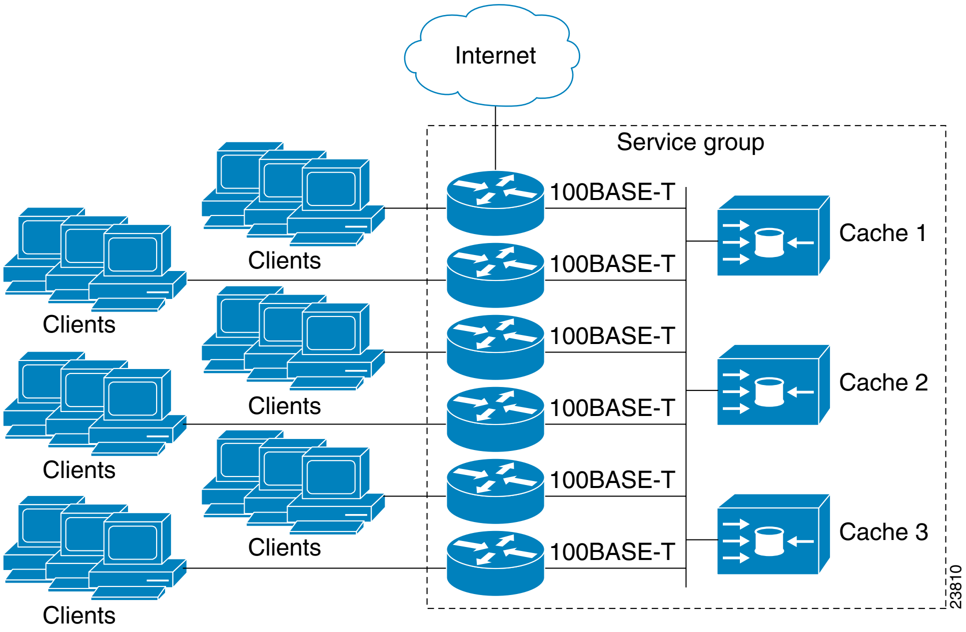
Note You can configure the Cisco Content Engine software release 2.2 or later releases to use the WCCP Layer 2 redirection feature along with the mask assignment table.

Understanding WCCP Configuration

Note WCCPv1 is not supported.

Multiple routers can use WCCP to service a cache cluster. Figure 49-1 illustrates a sample configuration using multiple routers.

Figure 49-1 Cisco Content Engine Network Configuration Using WCCP



The subset of content 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.

WCCP requires that each content 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 content engine. In this case the address of each router in the group must be explicitly specified for each content engine during configuration.
- Multicast—A single multicast address is configured on each content engine. In the multicast address method, the content engine sends a single-address notification that provides coverage for all routers in the service group. For example, a content 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 content engine. This option also enables you to add and remove routers from a service group dynamically without needing to reconfigure the content engines with a different list of addresses each time.

The following sequence of events shows how WCCP configuration works:

1. Each content engine is configured with a list of routers.
2. Each content engine announces its presence and a list of all routers with which it has established communications. The routers reply with their view (list) of content engines in the group.
3. Once the view is consistent across all content engines in the cluster, one content 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 WCCP on routers so they may participate in a service group.

WCCP Features

These sections describe WCCP features:

- Support for HTTP and Non-HTTP Services
- Support for Multiple Routers
- MD5 Security
- Web Content Packet Return

Support for HTTP and Non-HTTP Services

WCCP enables redirection of HTTP traffic (TCP port 80 traffic), as well as non-HTTP traffic (TCP and UDP). WCCP 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, WCCP 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.

Note The Catalyst 4500 series switch supports up to eight service groups.

For information on supported WCCP version 2 services with ACNS version 5.2 software, refer to the *Release Notes for Cisco ACNS Software, Release 5.2.3*.

The content 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 level assigned to it. Packets are matched against service groups in priority order and redirected by the highest priority service group that matches traffic characteristics.

Support for Multiple Routers

WCCP enables you to attach multiple routers to a cluster of cache engines. The use of multiple routers in a service group enables redundancy, interface aggregation, and distribution of the redirection load.

MD5 Security

WCCP provides optional authentication that enables you to control which routers and content 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 Content Packet Return

If a content engine is unable to provide a requested object it has cached due to error or overload, the content engine returns the request to the router for onward transmission to the originally specified destination server. WCCP verifies which requests have been returned from the content engine unserviced. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the content cluster). This provides error handling transparency to clients.

Typical reasons why a content engine would reject packets and initiate the packet return feature include the following:

- Instances when the content engine is overloaded and has no room to service the packets.
- Instances when the content engine is filtering for certain conditions that make caching packets counterproductive (such as, when IP authentication has been turned on).

Restrictions for WCCP

The following limitations apply to WCCP:

- 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 are not relevant or duplicates. Appropriate filtering needs to be performed.
- A service group can comprise up to 32 content engines and 32 routers.
- All content engines in a cluster must be configured to communicate with all routers servicing the cluster.
- Up to 8 service groups are supported at the same time on the same client interface.
- The L2 rewrite forwarding method is supported, but GRE encapsulation is not.
- Direct L3 connectivity to content engines is required; L3 connectivity of one or more hops away is not supported.
- Layer 2 redirection requires that content engines and clients I/Fs be directly connected to a router and should be on separate IP subnetworks
- The TCAM friendly mask-based assignment is supported, but the hash bucket-based method is not.
- Redirect ACL for WCCP on a client interface is not supported.
- Incoming traffic redirection on an interface is supported, but outgoing traffic re-direction is not.
- When TCAM space is exhausted, traffic is not redirected; it is forwarded normally.
- WCCP version 2 standard allows for support of up to 256 distinct masks. However, a Catalyst 4500 series switch only supports mask assignment table with a single mask.

Configuring WCCP

The following configuration tasks assume that you have already installed and configured the content engines you want to include in your network. You must configure the content engines in the cluster before configuring WCCP functionality on your routers. Refer to the *Cisco Content Engine User Guide* for content engine configuration and setup tasks.

IP must be configured on the router interface connected to the cache engines. Examples of router configuration tasks follow this section. For complete descriptions of the command syntax, refer to the *Cisco IOS Configuration Fundamentals Command Reference*, *Cisco IOS Release 12.3*.

These sections describe how to configure WCCP:

- Configuring a Service Group Using WCCP, page 49-6 (Required)
- Using Access Lists for a WCCP Service Group, page 49-7 (Optional)
- Setting a Password for a Router and Cache Engines, page 49-7 (Optional)

Configuring a Service Group Using WCCP

WCCP uses service groups based on logical redirection services, deployed for intercepting and redirecting traffic. The standard service is the content engine, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the content 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 content 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).

For information on supported WCCP services with ACNS version 5.2 software, refer to the *Release Notes for Cisco ACNS Software, Release 5.2.3*.

In addition to the web cache service, there can be up to seven dynamic services running concurrently on the switch.

Note More than one service can run on a switch at the same time, and routers and content engines can be part of multiple service groups at the same time.

The dynamic services are defined by the content engines; the content engine 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 content engine to join the group. In a dynamic service, up to eight ports can be specified within a single protocol TCP or UDP).

Cisco Content Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other content engines may use this service number for some other service. The following configuration information deals with enabling general services on Cisco routers. Refer to the content engine documentation for information on configuring services on content engines.

To enable a service on a Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`ip wccp`** `{`**`web-cache`** `|` *`service-number`*`}` `[`**`group-address`** *`groupaddress`*`]` `[`**`group-list`** *`access-list`*`]` `[`**`password`** *`password`*`]`	Specifies a dynamic service to enable on the switch, specifies the IP multicast address used by the service group (optional), group list to use for content engine membership (optional), specifies whether to use MD5 authentication (optional), and enables the WCCP service.
Step 2	`Switch(config)#` **`interface`** *`type number`*	Specifies a client interface to configure and enters interface configuration mode.
Step 3	`Switch(config-if)#` **`ip wccp`** `{`**`web-cache`** `|` *`service-number`*`}` **`redirect in`**	Enables WCCP redirection for ingress traffic on the specified client interface.

	Command	Purpose
Step 4	`Switch(config)#` **`interface`** *`type number`*	(Only needed to run the multicast feature) Specifies the content engine interface to be configured for multicast reception.
Step 5	`Switch(config-if)#` **`ip wccp`** `{`**`web-cache`** `|` *`service-number`*`}` **`group-listen`**	(Only needed to run the multicast feature) Enables the reception of IP multicast packets (WCCP protocol packets originating from the content engines) on the interface specified in Step 4.

Specifying a Web Cache Service

To configure a web-cache service, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`ip wccp web-cache`**	Enables the web cache service on the switch.
Step 2	`Switch(config)#` **`interface`** *`type number`*	Targets a client interface number for which the web cache service runs, and enters interface configuration mode.
Step 3	`Switch(config-if)#` **`ip wccp web-cache redirect in`**	Enables the check on packets to determine if they qualify to be redirected to a content engine, using the client interface specified in Step 2.

Using Access Lists for a WCCP Service Group

A Catalyst 4500 series switch can use an access list to restrict the content engines that can join a service group.

To restrict a content engine, perform this task:

	Command	Purpose
Step 1	`Switch(config)#` **`access-list`** *`access-list`* **`permit ip host`** *`host-address`* `[`*`destination-address`* `|` *`destination-host`* `|` **`any`**`]`	Creates an access list based on the unicast address of the content engines.
Step 2	`Switch(config)#` **`ip wccp web-cache group-list`** *`access-list`*	Indicates to the switch which content engines are allowed or disallowed to form a service group.

Setting a Password for a Router and Cache Engines

MD5 password security requires that each content engine and Catalyst 4500 series switch that wants to join a service group be configured with the service group password. The password can consist of up to seven characters. Each content engine or Catalyst 4500 series switch in the service group authenticates the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication are discarded.

To configure an MD5 password for use by the Catalyst 4500 series switch in WCCP communications, perform this task:

Command	Purpose
`Switch(config)# ip wccp web-cache password password`	Sets an MD5 password on the Catalyst 4500 series switch.

Verifying and Monitoring WCCP Configuration Settings

To verify and monitor the configuration settings for WCCP, use the following commands in EXEC mode:

Command	Purpose
`Switch# show ip wccp [web-cache | service-number]`	Displays global information related to WCCP, including the protocol version currently running, the number of content engines in the routers service group, which content engine group is allowed to connect to the router, and which access list is being used.
`Switch# show ip wccp {web-cache | service-number} detail`	Queries the router for information on which content 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.
`Switch# show ip interface`	Displays status about whether any **ip wccp** redirection commands are configured on a client interface. For example, "Web Cache Redirect is enabled / disabled."
`Switch# show ip wccp {web-cache | service-number} view`	Displays which devices in a particular service group have been detected and which content engines are having trouble becoming visible to all other switches to which the current switch is connected. The **view** keyword indicates a list of addresses of the service group. The information can be displayed for either the web cache service or the specified dynamic service. For further troubleshooting information, use the **show ip wccp** {**web-cache** | *service number*} **service** command.

WCCP Configuration Examples

This section provides the following configuration examples:

- Performing a General WCCP Configuration Example, page 49-9
- Running a Web Cache Service Example, page 49-9
- Running a Reverse Proxy Service Example, page 49-9

- Using Access Lists Example, page 49-9
- Setting a Password for a Switch and Content Engines Example, page 49-10
- Verifying WCCP Settings Example, page 49-10

Performing a General WCCP Configuration Example

The following example shows a general WCCP configuration session. VLAN 20 is for the client interface. VLAN 50 is for the content engine interface.

```
Switch# configure terminal
Switch(config)# ip wccp web-cache group-address 224.1.1.100 password alaska1
Switch(config)# interface vlan 20
Switch(config-if)# ip wccp web-cache redirect in
Switch(config)# interface vlan 50
Switch(config-if)# ip wccp web cache group-listen
```

Running a Web Cache Service Example

The following example shows a web cache service configuration session:

```
Switch# configure terminal
Switch(config)# ip wccp web-cache
Switch(config)# interface vlan 20
Switch(config-if)# ip wccp web-cache redirect in
Switch(config-if)# ^Z
Switch# copy running-config startup-config
Switch# show ip interface vlan 20 | include WCCP Redirect
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
```

Running a Reverse Proxy Service Example

The following example assumes you a configuring a service group using Cisco Content Engines, which use dynamic service 99 to run a reverse proxy service:

```
Switch# configure terminal
router(config)# ip wccp 99
router(config)# interface vlan 40
router(config-if)# ip wccp 99 redirect in
```

Using Access Lists Example

To achieve better security, you can use a standard access list to notify the Catalyst 4500 series switch to which IP addresses are valid addresses for a content engine attempting to register with the current switch. 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
```

Setting a Password for a Switch and Content Engines Example

The following example shows a WCCP password configuration session where the password is *alaska1*:

```
Switch# 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 Catalyst 4500 series switch:

```
Switch# more system:running-config

    Building configuration...
    Current configuration:
    !
    version 12.2
    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 in
    ip wccp 99 redirect in
    no ip route-cache
    no ip mroute-cache
    !
    interface Vlan300
    ip address 10.4.1.1 255.255.255.0
    !
    interface Serial0
    no ip address
    no ip directed-broadcast
    no ip route-cache
    no ip mroute-cache
    shutdown
    !
```

```
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
!
line con 0
transport input none
line aux 0
transport input all
line vty 0 4
password alaska1
login
!
end
```

CHAPTER 50

Configuring MIB Support

This chapter describes how to configure SNMP and MIB support for the Cisco 4500 series switch. It includes the following sections:

Determining MIB Support for Cisco IOS Releases

Follow these steps to determine which MIBs are included in the Cisco IOS release running on the Cisco 4500 series switch:

Step 1 Go to the Cisco MIBs Support page:

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

Step 2 Under Cisco Access Products, select a **Cisco 4500 switch** to display a list of MIBs supported on the Cisco 4500 switches.

Step 3 Scroll through the list to find the release you are interested in.

Using Cisco IOS MIB Tools

This section describes how to access the Cisco MIB tools page. The MIB Locator finds MIBs in Cisco IOS software releases. You can find general MIB information, instructions about how to use the SNMP Object Navigator which translates SNMP object identifiers (OIDs) into SNMP names, and how to load Cisco MIBs.

Follow these steps to access the Cisco IOS MIB tools site:

Step 1 Go to the Cisco Products and Services page:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

Step 2 Click **MIB Locator** to launch the application. The MIB Locator application allows you to find a MIB in the following three ways.

From the MIB Locator page, you can:

a. Click the drop-down menu and select the desired Cisco IOS software release.

b. From the Platform Family menu, select the appropriate feature set: **CAT4500-SUP2-PLUS**, **CAT4500-SUP2-PLUS-TS**, **CAT4500-SUP3**, **CAT4500-SUP4**, **CAT4500-SUP5**, **CAT4500-SUP5-10gGE2**, and **CAT4948**. If you select the platform first, the system displays only those releases and feature sets that apply to the Cisco 4500 series switch.

c. From the Feature Set menu, select **Service Provider W/VIP**.

Step 3 From the MIB Locator page, you can search by image name. For example, enter the following and click the **Submit** button:

```
c7200-js56i-mz.12.0-1
```

Step 4 From the MIB Locator page, you can search for the MIB from the list of MIBs in the menu. You can select one, or for multiple selections, hold down the **CTRL** key, then click the **Submit** button.

Note After you make a selection, follow the links and instructions.

Downloading and Compiling MIBs

The following sections provide information about how to download and compile MIBs for the Cisco 4500 series switch:

- Considerations for Working with MIBs
- Downloading MIBs
- Compiling MIBs

Considerations for Working with MIBs

While working with MIBs, consider the following:

Mismatches on Datatype Definitions

- Mismatches on datatype definitions might cause compiler errors or warning messages. Although Cisco MIB datatype definitions are not mismatched, some standard RFC MIBs do mismatch. For example:

```
MIB A defines: SomeDatatype ::= INTEGER(0..100)
MIB B defines: SomeDatatype ::= INTEGER(1..50)
```

  This example is considered to be a trivial error and the MIB loads successfully with a warning message.

  The next example is considered as a nontrivial error (even though the two definitions are essentially equivalent), and the MIB is not successfully parsed.

```
MIB A defines: SomeDatatype ::= DisplayString
MIB B defines: SomeDatatype ::= OCTET STRING (SIZE(0..255))
```

If your MIB compiler treats these as errors, or you want to delete the warning messages, edit one of the MIBs that define this same datatype so that the definitions match.

- Many MIBs import definitions from other MIBs. If your management application requires MIBs to be loaded, and you experience problems with undefined objects, you might want to load the following MIBs in this order:

  SNMPv2-SMI.my
  SNMPv2-TC.my
  SNMPv2-MIB.my
  RFC1213-MIB.my
  IF-MIB.my
  CISCO-SMI.my
  CISCO-PRODUCTS-MIB.my
  CISCO-TC.my

- For additional information and SNMP technical tips, from the Locator page, click **SNMP MIB Technical Tips** and follow the links or go to the following URL:

  http://www.cisco.com/pcgi-bin/Support/browse/psp_view.plp=Internetworking:SNMP&s=Implementation_and_Configuration#Samples_and_Tips

- For a list of SNMP OIDs assigned to MIB objects, go to the following URL and click on **SNMP Object Navigator** and follow the links:

  http://tools.cisco.com/ITDIT/MIBS/servlet/index

Note You must have a Cisco CCO name and password to access the MIB Locator.

- For information about how to download and compile Cisco MIBs, go to the following URL:

  http://www.cisco.com/warp/public/477/SNMP/mibcompilers.html

Downloading MIBs

Follow these steps to download the MIBs onto your system if they are not already there:

Step 1 Review the considerations in the previous section (“Considerations for Working with MIBs”).

Step 2 Go to one of the following Cisco URLs. If the MIB you want to download is not there, try the other URL; otherwise, go to one of the URLs in Step 5.

ftp://ftp.cisco.com/pub/mibs/v2

ftp://ftp.cisco.com/pub/mibs/v1

Step 3 Click the link for a MIB to download that MIB to your system.

Step 4 Select **File > Save** or **File > Save As** to save the MIB on your system.

Step 5 You can download industry-standard MIBs from the following URLs:

- http://www.ietf.org
- http://www.atmforum.com

Compiling MIBs

If you plan to integrate the Cisco 4500 series switch with an SNMP-based management application, then you must also compile the MIBs for that platform. For example, if you are running HP OpenView on a UNIX operating system, you must compile Cisco 4500 series switch MIBs with the HP OpenView Network Management System (NMS). For instructions, see the NMS documentation.

Enabling SNMP Support

The following procedure summarizes how to configure the Cisco 4500 series switch for SNMP support.

For detailed information about SNMP commands, see the following Cisco documents:

- *Cisco IOS Release 12.3 Configuration Fundamentals Configuration Guide*, “Monitoring the Router and Network” section, available at the following URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ffun_c/index.htm

- *Cisco IOS Release 12.3 Configuration Fundamentals Command Reference*, Part 3: System Management Commands, “Router and Network Configuration Commands” section, available at the the following URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ffun_r/index.htm

To configure the Cisco 4500 series switch for SNMP support, follow these steps:

Step 1 Set up your basic SNMP configuration through the command line interface (CLI) on the router. Note that these basic configuration commands are issued for SNMP version 2c. For SNMP version 3, you must also set up SNMP users and groups. (See the preceding list of documents for command and setup information.)

a. Define SNMP read-only and read-write communities:

```
Router (config)# snmp-server community Read_Only_Community_Name ro
Router (config)# snmp-server community Read_Write_Community_Name rw
```

b. Configure SNMP views (to limit the range of objects accessible to different SNMP user groups):

```
Router (config)# snmp-server view view_name oid-tree {included | excluded}
```

Step 2 Identify (by IP address) the host to receive SNMP notifications from the router:

```
Router (config)# snmp-server host host
```

Step 3 Configure the router to generate notifications. You can use keywords to limit the number and types of messages generated.

```
Router (config)# snmp-server enable traps [notification-type] [notification-option]
```

Step 4 Optional. Configure the router to generate SNMP notifications released to field replaceable units (FRUs):

```
Router (config)# snmp-server enable traps fru-ctrl
```

Step 5 Optional. Configure the router to generate SNMP notifications related to environmental monitoring:

```
Router (config)# snmp-server enable traps envmon
```

CHAPTER 51

ROM Monitor

This appendix describes the Cisco 806 router ROM monitor (also called the bootstrap program). The ROM monitor firmware runs when the router is powered up or reset. The firmware helps to initialize the processor hardware and boot the operating system software. You can use the ROM monitor to perform certain configuration tasks, such as recovering a lost password or downloading software over the console port. If there is no Cisco IOS software image loaded on the router, the ROM monitor runs the router.

This appendix contains the following sections:

- Entering the ROM Monitor
- ROM Monitor Commands
- Command Descriptions
- Configuration Register
- Console Download
- Debug Commands
- Exiting the ROM Monitor

Entering the ROM Monitor

To use the ROM monitor, you must be using a terminal or PC that is connected to the router over the console port. Refer to the installation chapter in the *Cisco 806 Router Hardware Installation Guide* that came with the router to connect the router to a PC or terminal.

Perform these steps to configure the router to boot up in ROM monitor mode the next time it is rebooted.

	Command	Task
Step 1	**enable**	If an enable password is configured, enter the enable command and the enable password to enter privileged EXEC mode.
Step 2	**configure terminal**	Enter global configuration mode.
Step 3	**config-reg 0x0**	Reset the configuration register.
Step 4	**exit**	Exit global configuration mode.
Step 5	**reload**	Reboot the router with the new configuration register value. The router remains in ROM monitor and does not boot the Cisco IOS software. As long as the configuration value is 0x0, you must manually boot the operating system from the console. See the **boot** command in the “Command Descriptions” section in this appendix. After the router reboots, it is in ROM monitor mode. The number in the prompt increments with each new line.

ROM Monitor Commands

Enter **?** or **help** at the ROM monitor prompt to display a list of available commands and options, as follows:

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
dev                 list the device table
dir                 list files in file system
help                monitor builtin command help
history             monitor command history
meminfo             main memory information
repeat              repeat a monitor command
reset               system reset
set                 display the monitor variables
sysret              print out info from last system return
unalias             unset an alias
unset               unset a monitor variable
```

Commands are case sensitive. You can halt any command by pressing the Break key on a terminal. If you are using a PC, most terminal emulation programs halt a command when you press the Ctrl and the Break keys at the same time. If you are using another type of terminal emulator or terminal emulation software, refer to the documentation for that product for information on how to send a Break command.

Command Descriptions

Table 51-1 describes the most commonly used ROM monitor commands.

Table 51-1 Most Commonly Used ROM Monitor Commands

Command	Description
reset or **i**	Resets and initializes the router, similar to a power up.

dev	Lists boot device identifications on the router; for example:
	`rommon 10> dev`
	`Devices in device table:`
	`        id  name`
	`    flash:  flash`

dir *device***:**	Lists the files on the named device; flash, for example:
	`rommon 4 > dir flash:`
	`      File size           Checksum   File name`
	`2835276 bytes (0x2b434c)   0x2073    c806-oy6-mz`

boot commands	For more information about the ROM monitor boot commands, refer to the *Cisco IOS Configuration Guide* and the *Cisco IOS Command Reference.*
b	Boots the first image in Flash memory.
b flash: [*filename*]	Attempts to boot the image directly from the first partition of Flash memory. If you do not enter a filename, this command will boot this first image in Flash.

Configuration Register

The virtual configuration register is in nonvolatile RAM (NVRAM) and has the same functionality as other Cisco routers. You can view or modify the virtual configuration register from either the ROM monitor or the operating system software. Within ROM monitor, you can change the configuration register by entering the register value in hexadecimal format, or by allowing the ROM monitor to prompt you for the setting of each bit.

Changing the Configuration Register Manually

To change the virtual configuration register from the ROM monitor manually, enter the command **confreg** followed by the new value of the register in hexadecimal, as shown in the following example:

```
rommon 1 > confreg 0x2101


You must reset or power cycle for new config to take effect
rommon 2 >
```

The value is always interpreted as hexadecimal. The new virtual configuration register value is written into NVRAM but does not take effect until you reset or reboot the router.

Changing the Configuration Register Using Prompts

Entering **confreg** without an argument displays the contents of the virtual configuration register and a prompt to alter the contents by describing the meaning of each bit.

In either case, the new virtual configuration register value is written into NVRAM but does not take effect until you reset or reboot the router.

The following display shows an example of entering the **confreg** command:

```
rommon 7> confreg

     Configuration Summary
enabled are:
console baud: 9600
boot: the ROM Monitor

do you wish to change the configuration? y/n  [n]:  y
enable  "diagnostic mode"? y/n  [n]:  y
enable  "use net in IP bcast address"? y/n  [n]:
enable  "load rom after netboot fails"? y/n  [n]:
enable  "use all zero broadcast"? y/n  [n]:
enable  "break/abort has effect"? y/n  [n]:
enable  "ignore system config info"? y/n  [n]:
change console baud rate? y/n  [n]:  y
enter rate: 0 = 9600, 1 = 4800, 2 = 1200, 3 = 2400  [0]:  0
change the boot characteristics? y/n  [n]:  y
enter to boot:
 0 = ROM Monitor
 1 = the boot helper image
 2-15 = boot system
    [0]:  0

Configuration Summary
enabled are:
diagnostic mode
console baud: 9600
boot: the ROM Monitor

do you wish to change the configuration? y/n  [n]:


You must reset or power cycle for new config to take effect
```

Console Download

You can use console download, a ROM monitor function, to download over the router console port either a software image or a configuration file. After download, the file is either saved to the mini-Flash memory module or to main memory for execution (image files only).

Use console download when you do not have access to a Trivial File Transfer Protocol (TFTP) server.

Note If you want to download a software image or a configuration file to the router over the console port, you must use the **ROM monitor** command.

Note If you are using a PC to download a Cisco IOS image over the router console port at 115,200 bps, ensure that the PC serial port is using a 16550 universal asynchronous transmitter/receiver (UART). If the PC serial port is not using a 16550 UART, we recommend using a speed of 38,400 or less when downloading a Cisco IOS image over the console port.

Error Reporting

Because the ROM monitor console download uses the console to perform the data transfer, error messages are only displayed on the console when the data transfer is terminated.

If an error does occur during a data transfer, the transfer is terminated, and an error message is displayed. If you have changed the baud rate from the default rate, the error message is followed by a message telling you to restore the terminal to the baud rate specified in the configuration register.

Debug Commands

Most ROM monitor debugging commands are functional only when Cisco IOS software has crashed or is halted.

The following are ROM monitor debugging commands:

- **frame**—displays an individual stack frame.
- **sysret**—displays return information from the last booted system image. This information includes the reason for terminating the image, a stack dump of up to eight frames, and, if an exception is involved, the address where the exception occurred; for example:

```
rommon 8> sysret
System Return Info:
count: 19,  reason: user break
pc:0x801111b0,  error address: 0x801111b0
Stack Trace:
FP: 0x80005ea8, PC: 0x801111b0
FP: 0x80005eb4, PC: 0x80113694
FP: 0x80005f74, PC: 0x8010eb44
FP: 0x80005f9c, PC: 0x80008118
FP: 0x80005fac, PC: 0x80008064
FP: 0x80005fc4, PC: 0xfff03d70
FP: 0x80005ffc, PC: 0x00000000
FP: 0x00000000, PC: 0x00000000
```

- **meminfo**—displays size in bytes, starting address, available range of main memory, the starting point and size of packet memory, and size of NVRAM; for example:

```
rommon 9> meminfo

Main memory size: 40 MB.
Available main memory starts at 0x10000, size 40896KB
IO (packet) memory size: 5 percent of main memory.
NVRAM size: 32KB
```

Exiting the ROM Monitor

You must set the configuration register to a value from 0x2 to 0xF for the router to boot a Cisco IOS image from Flash memory upon startup or reloading.

The following example shows how to reset the configuration register and cause the router to boot a Cisco IOS image stored in Flash memory:

```
rommon 1 > confreg 0x2101


You must reset or power cycle for new config to take effect
rommon 2 >boot
```

The router will boot the Cisco IOS image in Flash memory. The configuration register will change to 0x2101 the next time the router is reset or power cycled.

INDEX

APPENDIX A

Acronyms and Abbreviations

Table A-1 defines the acronyms and abbreviations used in this publication.

Table A-1 Acronyms

Acronym	Expansion
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
AV	attribute value
AVVID	Architecture for Voice, Video and Integrated Data
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
CEF	Cisco Express Forwarding
CGMP	Cisco Group Management Protocol

Table A-1 Acronyms (continued)

Acronym	Expansion
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
DBL	Dynamic Buffer Limiting
DCC	Data Country Code
dCEF	distributed Cisco Express Forwarding
DDR	dial-on-demand routing
DE	discard eligibility
DEC	Digital Equipment Corporation
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
DRAM	dynamic RAM
DSAP	destination service access point
DSCP	differentiated services code point
DSPU	downstream SNA Physical Units
DTP	Dynamic Trunking Protocol
DTR	data terminal ready
DXI	data exchange interface

Table A-1 Acronyms (continued)

Acronym	Expansion
EAP	Extensible Authentication Protocol
EARL	Enhanced Address Recognition Logic
EEPROM	electrically erasable programmable read-only memory
EHSA	enhanced high system availability
EHT	Explicit Host Tracking
EIA	Electronic Industries Association
ELAN	Emulated Local Area Network
EOBC	Ethernet out-of-band channel
ESI	end-system identifier
FECN	forward explicit congestion notification
FM	feature manager
FRU	field replaceable unit
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
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
IS-IS	Intermediate System-to-Intermediate System Intradomain Routing Protocol
ISL	Inter-Switch Link
ISO	International Organization of Standardization
LAN	local area network
LANE	LAN Emulation
LAPB	Link Access Procedure, Balanced

Table A-1 Acronyms (continued)

Acronym	Expansion
LDA	Local Director Acceleration
LCP	Link Control Protocol
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
MACL	MAC Access Control
MD5	Message Digest 5
MFD	multicast fast drop
MIB	Management Information Base
MII	media-independent interface
MLS	Multilayer Switching
MLSE	maintenance loop signaling entity
MOP	Maintenance Operation Protocol
MOTD	message-of-the-day
MLSE	maintenance loops signaling entity
MRM	multicast routing monitor
MSDP	Multicast Source Discovery Protocol
MST	Multiple Spanning Tree
MSTI	MST instance
MTU	maximum transmission unit
MVAP	multiple VLAN access port
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
NTP	Network Time Protocol
NVRAM	nonvolatile RAM

Table A-1 Acronyms (continued)

Acronym	Expansion
OAM	Operation, Administration, and Maintenance
ODM	order dependent merge
OSI	Open System Interconnection
OSPF	open shortest path first
PACL	Port Access Control List
PAE	port access entity
PAgP	Port Aggregation Protocol
PBD	packet buffer daughterboard
PBR	Policy Based Routing
PC	Personal Computer
PCM	pulse code modulation
PCR	peak cell rate
PDP	policy decision point
PDU	protocol data unit
PEP	policy enforcement point
PGM	Pragmatic General Multicast
PHY	physical sublayer
PIB	policy information base
PIM	Protocol Independent Multicast
PoE	Power over Internet
PPP	Point-to-Point Protocol
PRID	Policy Rule Identifiers
PVST+	Per VLAN Spanning Tree+
QM	QoS manager
QoS	quality of service
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

Table A-1 Acronyms (continued)

Acronym	Expansion
RPF	reverse path forwarding
RPR	Route Processor Redundancy
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
SPAN	Switched Port Analyzer
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

Table A-1 Acronyms (continued)

Acronym	Expansion
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
WFQ	weighted fair queueing
WRED	weighted random early detection
WRR	weighted round-robin
XNS	Xerox Network System