Cisco Reader Comment Card

General Information

1 Years of networking experience: _____ Years of experience with Cisco products: _____

2 I have these network types: ☐ LAN ☐ Backbone ☐ WAN
☐ Other: _____

3 I have these Cisco products: ☐ Switches ☐ Routers
☐ Other (specify models): _____

4 I perform these types of tasks: ☐ H/W installation and/or maintenance ☐ S/W configuration
☐ Network management ☐ Other: _____

5 I use these types of documentation: ☐ H/W installation ☐ H/W configuration ☐ S/W configuration
☐ Command reference ☐ Quick reference ☐ Release notes ☐ Online help
☐ Other: _____

6 I access this information through: ____% Cisco.com (CCO) ____% CD-ROM
____% Printed docs ____% Other: _____

7 I prefer this access method: _____

8 I use the following three product features the most:

Document Information

Document Title: Software Configuration Guide for the Catalyst 4006 Switch with Supervisor Engine III

Part Number: 78-14348-01 S/W Release: Cisco IOS Release 12.1(11b)EW

On a scale of 1–5 (5 being the best), please let us know how we rate in the following areas:

_____ The document is written at my technical level of understanding.

_____ The information is accurate.

_____ The document is complete.

_____ The information I wanted was easy to find.

_____ The information is well organized.

_____ The information I found was useful to my job.

Please comment on our lowest scores:

Mailing Information

Company Name Date

Contact Name Job Title

Mailing Address

City State/Province ZIP/Postal Code

Country Phone ( ) Extension

Fax ( ) E-mail

Can we contact you further concerning our documentation? ☐ Yes ☐ No

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

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 4631 SAN JOSE CA

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DOCUMENT RESOURCE CONNECTION
CISCO SYSTEMS INC
170 WEST TASMAN DRIVE
SAN JOSE CA 95134-9883

Software Configuration Guide for the Catalyst 4006 Switch with Supervisor Engine III

Cisco IOS Release 12.1(11b)EW

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7814348=
Text Part Number: 78-14348-01

Software Configuration Guide for the Catalyst 4006 Switch with Supervisor Engine III

CONTENTS

Preface

This preface describes this document, how it is organized, and its document conventions. The book also tells you how you can 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 4000 family 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 4000 family switches
Chapter 2	Command-Line Interfaces	Describes how to use the command-line interface (CLI)
Chapter 3	Configuring the Switch for the First Time	Describes how to perform a baseline configuration of the switch
Chapter 4	Configuring Interfaces	Describes how to configure non-layer-specific features on Fast Ethernet and Gigabit Ethernet interfaces
Chapter 5	Checking Port Status and Connectivity	Describes how to check module and interface status
Chapter 6	Configuring Layer 2 Ethernet Interfaces	Describes how to configure interfaces to support Layer 2 features, including VLAN trunks
Chapter 7	Understanding and Configuring VLANs	Describes how to set up and modify VLANs
Chapter 8	Understanding and Configuring Private VLANs	Describes how to set up and modify private VLANs

Chapter	Title	Description
Chapter 9	Understanding and Configuring VTP	Describes how to configure the VLAN Trunking Protocol
Chapter 10	Configuring Layer 3 Interfaces	Describes how to configure interfaces to support Layer 3 features
Chapter 11	Understanding and Configuring EtherChannel	Describes how to configure Layer 2 and Layer 3 EtherChannel port bundles
Chapter 12	Understanding and Configuring STP	Describes how to configure the Spanning Tree Protocol (STP) and explains how spanning tree works
Chapter 13	Configuring STP Features	Describes how to configure the spanning-tree PortFast, UplinkFast, and BackboneFast features
Chapter 14	Configuring Cisco Express Forwarding	Describes how to configure Cisco Express Forwarding (CEF) for IP unicast traffic
Chapter 15	Understanding and Configuring IP Multicast	Describes how to configure IP Multicast Multilayer Switching (MMLS)
Chapter 16	Configuring Network Security	Describes how to configure access control lists
Chapter 17	Understanding and Configuring IGMP Snooping	Describes how to configure Internet Group Management Protocol (IGMP) snooping
Chapter 18	Understanding and Configuring CDP	Describes how to configure Cisco Discovery Protocol (CDP)
Chapter 19	Understanding and Configuring UDLD	Describes how to configure the UniDirectional Link Detection (UDLD) protocol
Chapter 20	Understanding and Configuring QoS	Describes how to configure quality of service (QoS)
Chapter 21	Configuring SPAN	Describes how to configure the Switched Port Analyzer (SPAN)
Chapter 22	Power Management and Environmental Monitoring	Describes how to configure environmental monitoring, power redundancy, and inline power features
Chapter 23	Configuring Voice Interfaces	Describes how to configure multi-VLAN access ports for use with Cisco IP phones
Appendix A	Acronyms	Definitions for many commonly used Acronyms in this book

Related Documentation

The following publications are available for the Catalyst 4006 switch with Supervisor Engine III:

- *Catalyst 4000 Family Installation Guide*
- *Catalyst 4000 Family Module Installation Guide*
- *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*
- *System Message Guide for the Catalyst 4006 Switch with Supervisor Engine III*
- *Release Notes for the Catalyst 4006 Switch with Supervisor Engine III, Cisco IOS Release 12.1(11)EW*

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

  The Cisco IOS Configuration Guides and Command References are at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm
- For information about MIBs, refer to http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Conventions

This document uses the following typographical conventions:

Convention	Description
boldface font	Commands 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.

Convention	Description
[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.
⟶	This pointer highlights an important line of text in an example.
Ctrl-D	Ctrl represents the Control key—for example, the key combination **Ctrl-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.

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, please refer to the command in the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

Notes use the following conventions:

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

Cautions use the following conventions:

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

Obtaining Documentation

The following sections explain how to obtain documentation from Cisco Systems.

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at the following URL:

http://www.cisco.com

Translated documentation is available at the following URL:

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

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which is shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Ordering Documentation

Cisco documentation is available in the following ways:

- Registered Cisco Direct Customers can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

If you are reading Cisco product documentation on Cisco.com, you can submit technical comments electronically. Click **Leave Feedback** at the bottom of the Cisco Documentation home page. After you complete the form, print it out and fax it to Cisco at 408 527-0730.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

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

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools by using the Cisco Technical Assistance Center (TAC) Web Site. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC Web Site.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com is a highly integrated Internet application and a powerful, easy-to-use tool that provides a broad range of features and services to help you to

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

You can self-register on Cisco.com to obtain customized information and service. To access Cisco.com, go to the following URL:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two types of support are available through the Cisco TAC: the Cisco TAC Web Site and the Cisco TAC Escalation Center.

Inquiries to Cisco TAC are categorized according to the urgency of the issue:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

Which Cisco TAC resource you choose is based on the priority of the problem and the conditions of service contracts, when applicable.

Cisco TAC Web Site

The Cisco TAC Web Site allows you to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC Web Site, go to the following URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco services contract have complete access to the technical support resources on the Cisco TAC Web Site. The Cisco TAC Web Site requires a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to the following URL to register:

http://www.cisco.com/register/

If you cannot resolve your technical issues by using the Cisco TAC Web Site, and you are a Cisco.com registered user, you can open a case online by using the TAC Case Open tool at the following URL:

http://www.cisco.com/tac/caseopen

If you have Internet access, it is recommended that you open P3 and P4 cases through the Cisco TAC Web Site.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses issues that are classified as priority level 1 or priority level 2; these classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer will automatically open a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to the following URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled; for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). In addition, please have available your service agreement number and your product serial number.

CHAPTER 1

Product Overview

This chapter provides an overview of the Catalyst 4006 switch with Supervisor Engine III. This chapter includes these two major sections:

Understanding Layer 3 Switching

A Layer 3 switch is a high-performance switch that has been optimized for a campus LAN or intranet and that 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.

Supported Features

The following sections describes the key supported features:

Note For more information about the chassis, modules, and software features supported by the Catalyst 4006 switch with Supervisor Engine III, refer to the *Release Notes for Catalyst 4006 switch with Supervisor Engine III* at http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/relnotes/

Supported Hardware

The Catalyst 4006 switch supports the Supervisor Engine III and the following modules:

- WS-X4124-FX-MT—24-port 100BASE-FX Fast Ethernet switching module
- WS-X4148-FX-MT—48-port 100BASE-FX Fast Ethernet switching module
- WS-X4148-RJ—48-port 10/100 Fast Ethernet RJ-45
- WS-X4148-RJ21—48-port 10/100-Mbps Fast Ethernet switching module
- WS-X4148-RJ45-V—48-port inline power 10/100BASE-TX switching module
- WS-X4232-GB-RJ—32-port 10/100 Fast Ethernet RJ-45, plus 2-port 1000BASE-X (GBIC) Gigabit Ethernet
- WS-X4232-RJ-XX—32-port 10/100 Fast Ethernet RJ-45
- WS-X4306-GB—6-port 1000BASE-X (GBIC) Gigabit Ethernet
- WS-X4418-GB—18-port 1000BASE-X (GBIC) Gigabit Ethernet switching module
- WS-X4412-2GB-T—12-port 1000BASE-T Gigabit Ethernet switching module
- WS-X4424-GB-RJ45—24-port 10/100/1000BASE-T Gigabit Ethernet switching module
- WS-X4448-GB-LX—48-port 1000BASE-LX Gigabit Ethernet Fiber Optic interface module
- WS-X4448-GB-RJ45—48-port 10/100/1000BASE-T Gigabit Ethernet switching module
- WS-X4095-PEM—Catalyst 4000 DC Power Entry Module
- WS-P4603-2PSU—Catalyst 4000 Auxiliary Power Shelf (3-slot) including two WS-X4608 power supplies
- WS-X4608—Catalyst 4603 Power Supply Unit for WS-P4603

Layer 2 Software Features

The following sections describe the key Layer 2 switching software features on the Catalyst 4006 switch with Supervisor Engine III:

CDP

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.

EtherChannel Bundles

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

For more information on configuring EtherChannel, see Chapter 11, "Understanding and Configuring EtherChannel."

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.

The Catalyst 4006 switch with Supervisor Engine III supports the following spanning-tree enhancements:

- Spanning tree PortFast—PortFast allows a port with a directly attached host to transition to the forwarding state immediately, 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.

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, which 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, see Chapter 7, "Understanding and Configuring VLANs."

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.

  For more information about VTP, see Chapter 9, "Understanding and Configuring VTP."

- 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 more information about private VLANs, see Chapter 8, "Understanding and Configuring Private VLANs."

UniDirectional Link Detection

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 more information about UDLD, see Chapter 19, "Understanding and Configuring UDLD."

Layer 3 Software Features

The following sections describe the key Layer 3 switching software features on the Catalyst 4006 switch with Supervisor Engine III:

IP Routing Protocols

The following sections describe the routing protocols supported on the Catalyst 4006 switch with Supervisor Engine III:

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.)

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 grouping 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.

IGRP

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

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, partial 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 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).

BGP

The Border Gateway Protocol (BGP) is an Exterior Gateway Protocol (EGP) 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 4006 switch with Supervisor Engine III 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.

- For BGP configuration information, refer to “Configuring BGP” in the *Cisco IOS IP and IP Routing Configuration Guide* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt2/1cdbgp.htm

- For a complete description of the BGP commands, refer to "BGP Commands" in the *Cisco IOS IP and IP Routing Command Reference* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_r/iprprt2/1rdbgp.htm

- For multiprotocol BGP configuration information, refer to "Configuring Multiprotocol BGP Extensions for IP Multicast" in the *Cisco IOS IP and IP Routing Configuration Guide* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt2/1cdmbgp.htm

- For multiprotocol BGP command descriptions, refer to "Multiprotocol BGP Extensions for IP Multicast Commands" in the *Cisco IOS IP and IP Routing Command Reference* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_r/iprprt2/1rdmbgp.htm

HSRP

Hot Standby Routing Protocol (HSRP) provides high network availability by routing IP traffic from hosts on Ethernet networks without relying on the availability of any single 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 more information on configuring HSRP, refer to the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt1/1cdip.htm

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 makes forwarding decisions based on their content. Multicast traffic is forwarded only to ports with attached hosts that request multicast traffic.
- Protocol Independent Multicast (PIM)—PIM is a protocol-independent multicast service 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.

For information on configuring multicast services, see Chapter 15, "Understanding and Configuring IP Multicast."

CEF

The Catalyst 4006 switch with Supervisor Engine III features Cisco Express Forwarding (CEF). 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 characterized by 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.

Network Security (ACLs)

Access control lists (ACLs) filter network traffic by controlling whether routed packets are forwarded or blocked at the router's interfaces. The Catalyst 4006 switch with Supervisor Engine III examines each packet to determine whether to forward or drop the packet, based on the criteria you specified within the access lists.

For information on configuring access lists, see Chapter 16, "Configuring Network Security."

QoS Features

The quality of service (QoS) features select network traffic, prioritize it according to its relative importance, and use this information to avoid congestion. Implementing QoS in your network makes network performance more predictable and bandwidth use more effective.

The Catalyst 4006 switch with Supervisor Engine III supports the following QoS features:

- Classification and marking
- Ingress and egress policing
- Sharing and shaping

For information on configuring QoS, see Chapter 20, "Understanding and Configuring QoS."

Management and Security Features

The Catalyst 4006 switch with Supervisor Engine III offers network management and control through the CLI or through alternative access methods, such as SNMP. The switch software supports these network management and security features:

- Password-protected access (read-only and read-write)—This feature protects management interfaces against unauthorized configuration changes.
- Local, RADIUS, and TACACS+ authentication—These authentication methods control access to the switch. For additional information, refer to "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/ios121/121cgcr/secur_c/scprt1/index.htm
- Visual port status information—The switch LEDs provide visual management of port- and switch-level status.
- Switched Port Analyzer (SPAN)—SPAN allows you to monitor traffic on any port for analysis by a network analyzer or remote monitoring (RMON) probe. For information on SPAN, see Chapter 21, "Configuring SPAN."
- Simple Network Management Protocol (SNMP)—This protocol facilitates the exchange of management information between network devices. The Catalyst 4006 switch with Supervisor Engine III supports these SNMP types and enhancements:
  - SNMP—A full internet standard
  - SNMP v2—Community-based administrative framework for version 2 of SNMP
  - SNMP trap message enhancements—Additional information with certain SNMP trap messages, including spanning-tree topology change notifications and configuration change notifications

For information on SNMP, refer to the IOS *Configuration Fundamentals Configuration Guide* and *Configuration Fundamentals Command Reference* at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

- Dynamic Host Control Protocol (DHCP) server— DHCP enables you to automatically assign reusable IP addresses to DHCP clients. 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.

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

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t1/easyip2.htm

- Debugging features—The Catalyst 4006 switch with Supervisor Engine III has several commands to help you debug your initial setup. These commands include the following groups:
  - **show platform**
  - **platform**
  - **debug platform**

  For more information on these commands, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

CHAPTER 2

Command-Line Interfaces

This chapter describes the command-line interfaces (CLIs) you use to configure the Catalyst 4006 switch with Supervisor Engine III.

This chapter includes the following major sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Accessing the Switch CLI

The following sections describe how to access the switch CLI:

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 4000 Family Module Installation Guide* for console interface cable connection procedures.

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

	Task	Command
Step 1	At the prompt, press **Return.**	
Step 2	From the user EXEC prompt >, enter **enable** to change to enable mode (also known as privileged mode or privileged EXEC mode).	`Switch> enable`

Step 3	At the password prompt, enter the system password. The # prompt appears, indicating that you have accessed the CLI in enabled mode.	`Password: password`
		
		`Switch#`

Step 4	Enter the necessary commands to complete tasks.	
Step 5	When finished, exit the session.	`Switch# quit`

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 can telnet to the switch, you must set the IP address for the switch (see the "Configuring Physical Layer 3 Interfaces" section on page 10-5).

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

To telnet to the switch, perform this procedure:

	Task	Command
Step 1	From the remote host, enter the **telnet** command and the name or IP address of the switch you want to access.	**telnet** {*hostname* | *ip_addr*}

Step 2	At the prompt, enter the password for the CLI. If no password has been configured, press **Return**.	`Password: password`
		
		`Switch#`

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

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

Table 2-2 History Substitution Commands

Command	Result
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 successively 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.

IOS Command Modes

Note For complete information about 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/ios121/121cgcr/index.htm

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

When you start a session on the switch, you begin in user mode, also called user EXEC mode. Only a limited subset of commands are available in EXEC mode. To have access to all commands, you must enter privileged EXEC 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 at global configuration mode. From global configuration mode, you can enter interface configuration mode, subinterface configuration mode, and a variety of protocol-specific modes.

ROM-monitor mode is a separate mode used when the switch cannot boot up properly. For example, the switch might enter ROM-monitor mode if it does not find a valid system image when it is booting, or if its configuration file is corrupted. For more information, see the "ROM-Monitor Command-Line Interface" section on page 2-6.

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

Table 2-3 Frequently Used IOS Command Modes

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

The 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 IOS 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. For example, for the **configure** command, enter:

```
Switch# co?
```

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.

ROM-Monitor Command-Line Interface

The ROM-monitor (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 rebooting 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, see *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

CHAPTER 3

Configuring the Switch for the First Time

This chapter describes how to initially configure the Catalyst 4006 switch with Supervisor Engine III. This chapter supplements the administration information and procedures in these publications:

- *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.1, at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/index.htm
- *Cisco IOS Configuration Fundamentals Configuration Command Reference*, Release 12.1, at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/index.htm

This chapter includes the following major sections:

Note For complete syntax and usage information for the commands used in this publication, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

Default Switch Configuration

This sections describes the default configurations for the Catalyst 4006 switch with Supervisor Engine III. Table 3-1 shows the default switch configuration.

Table 3-1 Default Switch Configuration

Feature	Default Value
Administrative connection	Normal mode
Global switch information	No value for the following: • System name • System contact • Location

Table 3-1 Default Switch Configuration (continued)

Feature	Default Value
System clock	No value for system clock time
Passwords	No passwords are configured for normal mode or enable mode (press the **Return** key)
Switch prompt	`Switch>`
Interfaces	Enabled, with speed and flow control autonegotiated, and without IP addresses

Configuring the Switch

The following sections describe how to configure your switch:

Using Configuration Mode to Configure Your Switch

You can configure your switch from configuration mode with the following 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>`). 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 privileged EXEC prompt (#):

```
Switch#
```

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

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

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

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

In either of these configuration modes, you can enter changes to the switch configuration.

Step 5 Enter the **end** command to exit configuration mode.

Step 6 Save your settings. (See the "Saving the Running Configuration Settings" section on page 3-3.)

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

Checking the Running Configuration Settings

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

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

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.

Reviewing the Configuration in NVRAM

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

The following sample output 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 procedure:

	Task	Command
Step 1	Configure a default gateway.	`Switch(config)# ip default-gateway A.B.C.D`
Step 2	Verify that the default gateway is correctly displayed in the IP routing table.	`Switch# show ip route`

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, use this procedure:

	Task	Command
Step 1	Configure a static route to the remote network.	`Switch(config)# ip route dest_IP_address mask {forwarding_IP | vlan vlan_ID}`
Step 2	Verify that the static route is displayed correctly.	`Switch# show running-config`

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 171.20.52.3 255.255.255.255 Vlan1
no ip http server
!
!
x25 host z
!
line con 0
 transport input none
line vty 0 4
 exec-timeout 0 0
 password lab
 login
 transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end

Switch#
```

Protecting Access to Privileged EXEC Commands

The procedures in the following sections give you a way to control access to the system configuration file and privileged EXEC commands:

Setting or Changing a Static Enable Password

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

Command	Purpose
`Switch(config)# enable password password`	Sets a new password or change 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 3-11.

Using the enable password and enable secret Commands

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a TFTP server, you can use either the **enable password** or **enable secret** commands. Both commands configure an encrypted password that you must enter to access 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 of these tasks:

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

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 4006 switch with Supervisor Engine III 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 3-11 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 3-11.

Setting or Changing a Privileged Password

To set or change a privileged password, enter this command:

Command	Purpose
`Switch(config-line)# password password`	Sets a new password or change 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 3-11.

Setting TACACS+ Password Protection for Privileged EXEC Mode

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

- The "Authentication, Authorization, and Accounting (AAA)," chapter in *Cisco IOS Security Configuration Guide*, Release 12.1, at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt1/index.htm
- *Cisco IOS Security Command Reference*, Release 12.1, at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/index.htm

To set the TACACS+ protocol to determine whether a user can access privileged EXEC mode, enter this command:

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

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

Note When used without extended TACACS, the **enable use-tacacs** command allows anyone with a valid username and password to access the privileged EXEC mode, creating a potential security risk. This problem occurs because the query resulting from entering the **enable** command is indistinguishable from an attempt to log in without extended TACACS+.

Encrypting Passwords

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

To configure the IOS software to encrypt passwords, enter this command:

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 3-11 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 3-11.

Configuring Multiple Privilege Levels

By default, the 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 a smaller group of users.

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

- Setting the Privilege Level for a Command, page 3-10
- Changing the Default Privilege Level for Lines, page 3-10
- Logging In to a Privilege Level, page 3-10

Setting the Privilege Level for a Command

To set the privilege level for a command, use this procedure:

	Task	Command
Step 1	Set the privilege level for a command.	`Switch(config)# privilege mode level level command`
Step 2	Specify the enable password for a privilege level.	`Switch(config)# enable password level level [encryption-type] password`

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 3-11.

Changing the Default Privilege Level for Lines

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

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 3-11.

Logging In to a Privilege Level

To log in at a specified privilege level, enter this command:

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

Exiting a Privilege Level

To exit to a specified privilege level, enter this command:

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 procedure:

	Task	Command
Step 1	Display the password and access level and the privilege level configuration.	`Switch# show running-config`
Step 2	Show the privilege level configuration.	`Switch# show privilege`

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

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 boot-up.

Step 3 Configure the switch to boot-up without reading the configuration memory (NVRAM). See "Configuring the Software Configuration Register" section on page 3-13 for more information.

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 configuration register and BOOT variable:

Understanding the Supervisor Engine Boot Configuration

The supervisor engine boot-up 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 3-14. The BOOT environment variable is described in the "Specifying the Startup System Image" section on page 3-16.

Understanding the ROM Monitor

The ROM monitor (ROMMON) is invoked at switch boot-up, 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 r 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 described in the "Setting the BOOTLDR Environment Variable" section on page 3-18.

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 rebooting 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 boot up 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 written into NVRAM.

Following are some reasons for changing 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 allow you to manually boot-up the system using the **boot** command at the bootstrap program prompt
- To force an automatic boot-up 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 confusion and possibly halting the Catalyst 4006 switch with Supervisor Engine III, remember that valid configuration register settings might be combinations of settings and not just the individual settings listed in Table 3-2. For example, the factory default value of 0x0102 is a combination of settings.

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

Table 3-2 Software Configuration Register Bit Meaning

Bit Number[1]	Hexadecimal	Meaning
00 to 03	0x0000 to 0x000F	Boot field (see Table 3-3)
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	Load ROM monitor after netboot fails
14	0x4000	IP broadcasts do not have network numbers

1. The factory default value for the configuration register is 0x0102. This value is a combination of the following: binary bit 8 = 0x0100 and binary bits 00 through 03 = 0x0002 (see Table 3-3).
2. OEM = original equipment manufacturer.

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

Boot Field	Meaning
00	Stays at the system bootstrap prompt (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 in the BOOT variable, the switch attempts to boot from the next image listed in the BOOT variable. When the end of the BOOT variable is reached and the switch cannot boot successfully, the switch starts attempting 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 using and setting 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 0x0102.

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

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

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

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

	Task	Command
Step 1	Determine the current configuration register setting.	`Switch# show version`
Step 2	Enter configuration mode, selecting the terminal option.	`Switch# configure terminal`
Step 3	Modify the existing configuration register setting to reflect the way in which you want the switch to load a system image.	`Switch(config)# config-register value`
Step 4	Exit configuration mode.	`Switch(config)# end`
Step 5	Reboot the switch to make your changes take effect.	`Switch# reload`

Perform the following procedure to modify the configuration register while the switch is running IOS:

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 entering the **config-register** *value* configuration command, where *value* is a hexadecimal number preceded by 0x (see Table 3-2 on page 3-13).

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

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

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

Step 6 Save your settings. (See the "Saving the Running Configuration Settings" section on page 3-3. 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.

This completes the procedure for making configuration register changes.

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 value of the configuration register boot field.

To verify the configuration register setting for the switch, enter this command:

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 IOS *Configuration Fundamentals Configuration Guide*.

Use the following sections to configure your switch to boot from Flash memory. Flash memory can be either be 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 includes the following features:

- Can be remotely loaded with multiple system software images through TFTP or rcp transfers (one transfer for each file loaded).
- Allows you to 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 The system image stored in Flash memory can be changed only from privileged EXEC level on the console terminal.

Configuring Flash Memory

To configure your switch to boot from Flash memory, perform this 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 12.1, at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/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 3-14, for more information on modifying the configuration register.

Step 3 Save your configurations.

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

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. You must save the environment variable settings to your startup configuration to place the information under ROM monitor control and for the environment variables to function as expected. Enter the **copy system:running-config nvram:startup-config** command to save the environment variables from your running configuration to your startup configuration.

You can view the contents of the BOOT and 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#
```

Setting the BOOTLDR Environment Variable

The BOOTLDR environment specifies the Flash file system and file name that contains the boot loader image required to load system software. It defines the primary Cisco IOS image that will load the final image from another source.

To set the BOOTLDR environment variable, perform this procedure:

	Task	Command
Step 1	Verify that bootflash contains the boot loader image.	`Switch# dir bootflash:`
Step 2	Enter the configuration mode from the terminal.	`Switch# configure terminal`
Step 3	Set the BOOTLDR environment variable to specify the Flash device and file name of the boot loader image. This step modifies the runtime BOOTLDR environment variable.	`Switch(config)# boot bootldr bootflash:`*`boot_loader`*
Step 4	Exit configuration mode.	`Switch# end`
Step 5	Save this runtime BOOTLDR environment variable to your startup configuration.	`Switch# copy system:running-config nvram:startup-config`
Step 6	(Optional) Verify the contents of the BOOTLDR environment variable.	`Switch# show bootvar`

This example shows how to set the BOOTLDR variable:

```
Switch# dir bootflash:
Directory of bootflash:/

  1  -rw-     1599488   Nov 29 1999 11:12:29 cat4000-is-mz.XE.bin

15990784 bytes total (14391168 bytes free)
Switch# configure terminal
```

```
Switch (config)# boot bootldr bootflash:cat4000-is-mz.XE.bin
Switch (config)# end
Switch# copy system:running-config nvram:startup-config
[ok]
Switch# show bootvar
BOOTLDR variable = bootflash:cat4000-is-mz,1
Configuration register is 0x0
```

CHAPTER 4

Configuring Interfaces

This chapter describes how to configure interfaces for the Catalyst 4000 family 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 commands used in this publication, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

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)
- 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 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. Begin interface configuration in global configuration mode and perform the following procedure:

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 the 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#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 at the global config mode prompt:

```
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 "Monitoring and Maintaining the Interface" section on page 4-9.

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, enter this command:

Command	Purpose
`Switch(config)# interface range {vlan vlan_ID - vlan_ID}	{{fastethernet	gigabitethernet	macro macro_name} slot/interface - interface} [, {vlan vlan_ID - vlan_ID} {{fastethernet	gigabitethernet	macro macro_name} slot/interface - interface}]`	Selects the range of interfaces to be configured. Note the following: • The space before the dash is required. • 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 interface numbers 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 reenable all Fast Ethernet interfaces in the range 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, enter this command:

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, enter this command:

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, enter this command:

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)#
```

Configuring Optional Interface Features

The following sections describe optional procedures that you can perform on most interfaces:

Configuring Interface Speed and Duplex Mode

The following sections describe how to configure the interface speed and duplex mode:

Speed and Duplex Mode Configuration Guidelines

You can configure the interface speed and duplex mode parameters to **auto** and allow the Catalyst 4006 switch with Supervisor Engine III 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 set the interface **speed** to **auto**, the switch automatically sets the **duplex** mode to **auto**.
- 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, the duplex mode is full duplex. You cannot change the duplex mode.
- If the interface speed is set to 10 or 100 mbps, 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.

To set the port speed for a 10/100-Mbps Ethernet interface, perform this procedure:

	Task	Command
Step 1	Select the interface to be configured.	`Switch(config)# interface fastethernet slot/interface`
Step 2	Set the interface speed of the interface.	`Switch(config-if)# speed [10 | 100 | auto]`

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
```

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 procedure:

	Task	Command
Step 1	Select the interface to be configured.	`Switch(config)# interface gigabitethernet1/1`
Step 2	Disable autonegotiation on the interface.	`Switch(config-if)# speed nonegotiate`

To restore autonegotiation, enter a **no speed nonegotiate** command in the interface configuration mode.

For blocking ports on the WS-X4416 module, you cannot 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.

Note If you set the port speed to **auto** on a 10/100-Mbps Ethernet interface, both speed and duplex mode are autonegotiated. The configured duplex mode is not applied on autonegotiation interfaces.

To set the duplex mode of a Fast Ethernet interface, perform this procedure:

	Task	Command
Step 1	Select the interface to be configured.	`Switch(config)# interface fastethernet slot/interface`
Step 2	Set the duplex mode of the interface.	`Switch(config-if)# duplex [auto | full | half]`

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, enter the following command:

Command	Purpose
Switch# **show interfaces** [**fastethernet** | **gigabitethernet**] *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"
```

Understanding Online Insertion and Removal

The online insertion and removal (OIR) feature supported on the Catalyst 4000 family switches allows you to remove and replace modules while the system is online. You can shut down the interface processor 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 an interface processor. The system notifies the route processor 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

Cisco IOS for the Catalyst 4000 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 using the **show ?** command at the EXEC prompt.) These commands are fully described in the *Cisco IOS Interface Command Reference*.

To display information about the interface, perform this procedure:

	Task	Command
Step 1	Display the status and configuration of all interfaces or of a specific interface.	`Switch# show interfaces [type slot/interface]`
Step 2	Display the currently running configuration in RAM.	`Switch# show running-config`
Step 3	Display the global (system-wide) and interface-specific status of any configured protocol.	`Switch# show protocols [type slot/interface]`
Step 4	Display the hardware configuration, software version, the names and sources of configuration files, and the boot images.	`Switch# show version`

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 using SNMP; it clears only those counters seen 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 procedure:

	Task	Command
Step 1	Select the interface to be configured.	Switch(config)# **interface** {**vlan** *vlan_ID*} | {{**fastethernet** | **gigabitethernet**} *slot/port*} | {**port-channel** *port_channel_number*}
Step 2	Shut down the interface.	Switch(config-if)# **shutdown**
Step 3	Reenable the interface.	Switch(config-if)# **no shutdown**

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 check whether an interface is disabled, enter the EXEC **show interfaces** command. An interface that has been shut down is shown as administratively down when you enter the **show interfaces** command.

CHAPTER 5

Checking Port Status and Connectivity

This chapter describes how to check switch port status and connectivity on the Catalyst 4006 switch with Supervisor Engine III.

This chapter includes the following sections:

Note For complete syntax and usage information for the commands used in this publication, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm.

Checking Module Status

The Catalyst 4006 switch with Supervisor Engine III 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 of the 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 5-1.

This example shows how to display the status of all interfaces on a Catalyst 4000 Family switch:

```
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#
```

Checking 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#
```

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 3, "Configuring the Switch for the First Time."

Note To **telnet** to a host using the hostname, configure and enable DNS.

To **telnet** to another device on the network from the switch, enter this command in privileged EXEC mode:

Command	Purpose
`Switch# telnet host [port]`	Opens a Telnet session to a remote host.

This example shows how to **telnet** 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, enter the following command in privileged EXEC mode:

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, enter the following command 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, enter the following command in privileged EXEC mode:

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 5-5
- Running Ping, page 5-6

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, enter the following command in normal or 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 5-6
- Running IP Traceroute, page 5-7

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, enter the following command 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#
```

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.

To limit the rate that Internet Control Message Protocol (ICMP) destination unreachable messages are generated, enter the following command in global configuration mode:

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, use the **ip mask-reply** interface configuration command.

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 6

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 the Catalyst 4000 family 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:

Note To configure Layer 3 interfaces, see Chapter 10, "Configuring Layer 3 Interfaces."

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Overview of Layer 2 Ethernet Switching

The following sections describe how Layer 2 Ethernet switching works on the Catalyst 4000 family switches:

Understanding Layer 2 Ethernet Switching

The Catalyst 4006 switch with Supervisor Engine III supports 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

The Catalyst 4006 switch with Supervisor Engine III 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 4006 switch with Supervisor Engine III are full-duplex mode only, providing 2-Gbps effective bandwidth.

Switching Frames Between Segments

Each Ethernet interface on a Catalyst 4000 family 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 4006 switch with Supervisor Engine III 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)—ISL is a Cisco-proprietary trunking encapsulation.

Note Blocking Gigabit ports on the WS-X4418-GB and WS-X4412-2GB-T modules do not support ISL. Ports 3 - 18 are blocking Gigabit ports on the WS-X4418-GB module. Ports 1 - 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 11, "Understanding and Configuring EtherChannel."

Ethernet trunk interfaces support different trunking modes (see Table 6-2 on page 6-4). You can specify whether the trunk uses ISL encapsulation, 802.1Q encapsulation, or if the encapsulation type is autonegotiated.

To autonegotiate trunking, the interfaces must be 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 Chapter 9, "Understanding and Configuring VTP."

Trunk negotiation is managed by the Dynamic Trunking Protocol (DTP). DTP supports autonegotiation of both ISL and 802.1Q trunks.

Encapsulation Types

Table 6-1 lists the Ethernet trunk encapsulation types.

Table 6-1 Ethernet Trunk Encapsulation Types

Encapsulation Type	Encapsulation Command	Function
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 6-2 lists the Layer 2 interface modes and describes how they function on Ethernet interfaces.

Table 6-2 Layer 2 Interface Modes

Mode	Function
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.

Layer 2 Interface Configuration Guidelines and Restrictions

The following guidelines and restrictions apply when configuring 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.
- Ensure 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 potentially cause spanning tree loops.

Default Layer 2 Ethernet Interface Configuration

Table 6-3 shows the Layer 2 Ethernet interface default configuration.

Table 6-3 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
Native VLAN (for 802.1Q only trunks)	VLAN 1
Spanning Tree Protocol (STP)	Enabled for all VLANs
STP port priority	128
STP port cost	• 19 for 10/100-Mbps Fast Ethernet interfaces • 19 for 100-Mbps Fast Ethernet interfaces • 4 for 1000-Mbps Gigabit Ethernet interfaces

Configuring Ethernet Interfaces for Layer 2 Switching

The following sections describe how to configure Layer 2 switching on the Catalyst 4006 switch with Supervisor Engine III:

- Configuring an Ethernet Interface as a Layer 2 Trunk, page 6-5
- Configuring an Interface as a Layer 2 Access Port, page 6-7
- Clearing Layer 2 Configuration, page 6-9

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.

To configure an interface as a Layer 2 trunk, perform this procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 2	(Optional) Shut down the interface to prevent traffic flow until configuration is complete.	`Switch(config-if)# shutdown`
Step 3	(Optional) Specify 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**).	`Switch(config-if)# switchport trunk encapsulation {isl | dot1q | negotiate}`
Step 4	Configure the interface as a Layer 2 trunk. (Required only if the interface is a Layer 2 access port or to specify the trunking mode.)	`Switch(config-if)# switchport mode {dynamic {auto | desirable} | trunk}`
Step 5	(Optional) Specify the access VLAN, which is used if the interface stops trunking. The access VLAN is not used as the native VLAN.	`Switch(config-if)# switchport access vlan vlan_num`
Step 6	For 802.1Q trunks, specify the native VLAN. **Note** If you do not set the native VLAN, the default is used (VLAN 1).	`Switch(config-if)# switchport trunk native vlan vlan_num`
Step 7	(Optional) Configure 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.	`Switch(config-if)# switchport trunk allowed vlan {add | except | all | remove} vlan_num`[1]`[,vlan_num[,vlan_num[,....]]]`
Step 8	(Optional) Configure the list of VLANs allowed to be pruned from the trunk (see the "Understanding VTP Pruning" section on page 9-3). The default list of VLANs allowed to be pruned contains all VLANs, except for VLAN 1.	`Switch(config-if)# switchport trunk pruning vlan {add | except | none | remove} vlan_num`[1]`[,vlan_num[,vlan_num[,....]]]`
Step 9	Activate the interface. (Required only if you shut down the interface.)	`Switch(config-if)# no shutdown`
Step 10	Exit configuration mode.	`Switch(config-if)# end`
Step 11	Display the running configuration of the interface.	`Switch# show running-config interface {fastethernet | gigabitethernet} slot/port`
Step 12	Display the switch port configuration of the interface.	`Switch# show interfaces [fastethernet | gigabitethernet] slot/port switchport`
Step 13	Display the trunk configuration of the interface.	`Switch# show interfaces [{fastethernet | gigabitethernet} slot/port] trunk`

1. 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.

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 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

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

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 Mode" section on page 7-6).

To configure an interface as a Layer 2 access port, perform this procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 2	(Optional) Shut down the interface to prevent traffic flow until configuration is complete.	`Switch(config-if)# shutdown`
Step 3	Configure 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.	`Switch(config-if)# switchport`
Step 4	Configure the interface as a Layer 2 access port.	`Switch(config-if)# switchport mode access`
Step 5	Place the interface in a VLAN.	`Switch(config-if)# switchport access vlan vlan_num`
Step 6	Activate the interface. (Required only if you had shut down the interface.)	`Switch(config-if)# no shutdown`
Step 7	Exit configuration mode.	`Switch(config-if)# end`
Step 8	Display the running configuration of the interface.	`Switch# show running-config interface {fastethernet | gigabitethernet} slot/port`
Step 9	Display the switch port configuration of the interface.	`Switch# show interfaces [{fastethernet | gigabitethernet} slot/port] switchport`

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 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


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 procedure:

	Task	Command
Step 1	Select the interface to clear.	`Switch(config)# default interface {fastethernet | gigabitethernet} slot/port`
Step 2	Exit configuration mode.	`Switch(config-if)# end`
Step 3	Display the running configuration of the interface.	`Switch# show running-config interface {fastethernet | gigabitethernet} slot/port`
Step 4	Display the switch port configuration of the interface.	`Switch# show interfaces [{fastethernet | gigabitethernet} slot/port] switchport`

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 the Layer 2 configuration was cleared:

```
Switch# show running-config interface fastethernet 5/6
Building configuration...
Current configuration:
!
interface FastEthernet5/6
end

Switch# show interfaces fastethernet 5/6 switchport
Name: Fa5/6
Switchport: Enabled
Switch#
```

CHAPTER 7

Understanding and Configuring VLANs

This chapter describes VLANs on the Catalyst 4000 family 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 commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Overview of VLANs

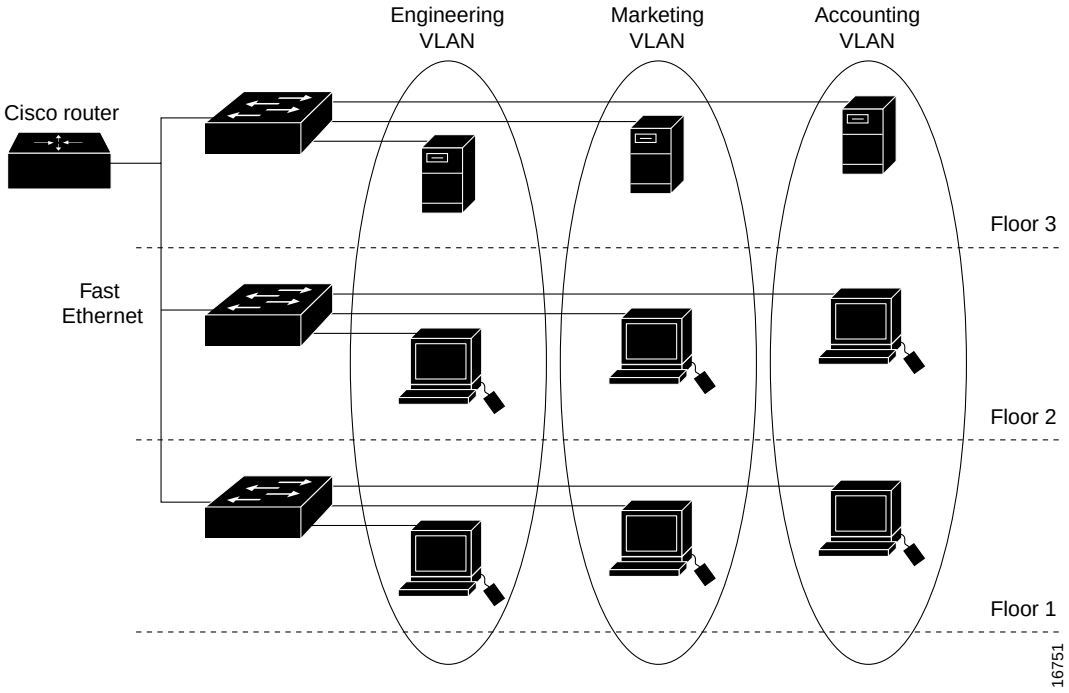
In the technical definition set forth by the IEEE, VLANs define broadcast domains in a Layer 2 network. A broadcast domain is the extent that a frame propagates through a network. Legacy networks use routers to define broadcast domain boundaries. Layer 2 switches create broadcast domains based on the configuration of the switch. Switches are multi-port 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 "Overview of Layer 3 Interfaces" section on page 10-1 for information on inter-VLAN routing on the Catalyst 4000 family switches.

VLANs have the same attributes as a physical LAN, but VLANs allow you to group end stations even when they are not connected physically to the same LAN segment.

Figure 7-1 shows an example of three VLANs in logically defined networks.

Figure 7-1 VLANs in Logically Defined Networks



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. LAN interface VLAN membership is assigned manually on an interface-by-interface basis. When you assign LAN interfaces to VLANs manually, it 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 (Ethernet, FDDI [Fiber Distributed Data Interface], FDDI network entity title [NET], Token Ring Bridge Relay Function [TrBRF], or Token Ring Concentrator Relay Function [TrCRF])
- VLAN state (active or suspended)
- Security Association Identifier (SAID)
- Bridge identification number for TrBRF VLANs
- Ring number for FDDI and TrCRF VLANs
- Parent VLAN number for TrCRF VLANs
- Spanning Tree Protocol (STP) type for TrCRF VLANs
- VLAN number to use when translating from one VLAN type to another

Note When translating from one VLAN type to another, the software 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 you can create a VLAN, the Catalyst 4000 family switch must be in VTP server mode or VTP transparent mode. If the Catalyst 4000 family switch is a VTP server, you must define a VTP domain. For information on configuring VTP, see Chapter 9, "Understanding and Configuring VTP."
- The Cisco IOS **end** command is not supported in VLAN database mode.
- You cannot enter **Ctrl-Z** to exit VLAN database mode.

VLAN Default Configuration

Tables 7-1 through 7-5 show the default configurations for the different VLAN media types.

Table 7-1 Ethernet VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1	1–1005
VLAN name	default	No range
802.10 SAID	100,001	1–4,294,967,294
MTU size	1500	No range
Translational bridge 1	1002	0–1005
Translational bridge 2	1003	0–1005
VLAN state	active	active; suspend; shutdown

Note Catalyst 4000 family 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.

Table 7-2 FDDI VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1002	1–1005
VLAN name	fddi-default	No range
802.10 SAID	101,002	1–4,294,967,294
MTU size	1500	No range
Ring number	0	1–4095
Parent VLAN	0	0–1005
Translational bridge 1	0	0–1005
Translational bridge 2	0	0–1005
VLAN state	active	active; suspend

Table 7-3 TrCRF VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1003	1–1005
VLAN name	VTPv1 token-ring-default; VTPv2 trcrf-default	No range
Parent VLAN	VTPv1 0; VTPv2 1005	No range
802.10 SAID	101,003	1–4,294,967,294
Ring Number	VTPv1 0; VTPv2 3276	1–4095
MTU size	VTPv1 default 1500; VTPv2 default 4472	No range
Translational bridge 1	0	0–1005
Translational bridge 2	0	0–1005
VLAN state	active	active; suspend
Bridge mode	VTPv1 none; VTPv2 srb	srb, srt
ARE max hops	7	0–13
STE max hops	7	0–13
Backup CRF	disabled	disable; enable

Table 7-4 FDDI-Net VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1004	1–1005
VLAN name	fddinet-default	No range
802.10 SAID	101,004	1–4,294,967,294
MTU size	1500	No range
Bridge number	0	0–15
STP type	ieee	auto; ibm; ieee
VLAN state	active	active; suspend

Table 7-5 TrBRF VLAN Defaults and Ranges

Parameter	Default	Range
VLAN ID	1005	1–1005
VLAN name	VTPv1 trnet-default; VTPv2 trbrf-default	No range
802.10 SAID	101,005	1–4,294,967,294
MTU size	VTPv1 1500; VTPv2 4472	No range
Bridge number	VTPv1 0; VTPv2 15	0–15
STP type	ibm	auto; ibm; ieee
VLAN state	active	active; suspend

Configuring VLANs

Note Before you configure VLANs, you must decide whether to use VLAN Trunking Protocol (VTP) to maintain global VLAN configuration information for your network. For complete information on VTP, see Chapter 9, "Understanding and Configuring VTP."

Note VLANs support a number of parameters that are not discussed in detail in this section. For complete information, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

These sections describe how to configure VLANs:

VLAN Configuration Options

These sections describe the VLAN configuration options:

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 *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

VLAN Configuration in Global Configuration Mode

If the switch is in VTP server or transparent mode (see the "Configuring VTP" section on page 9-6), you can configure VLANs in global and config-vlan configuration modes. When you configure VLANs in global and config-vlan configuration modes, the VLAN configuration is saved in the **vlan.dat** files. To display the VLAN configuration, enter the **show vlan** command.

If the switch is in VLAN transparent mode, the **copy running-config startup-config** command saves 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.

VLAN Configuration in VLAN Database Mode

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

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

Configuring VLANs in Global Mode

User-configured VLANs have unique IDs from 1 to 1001. To create a VLAN, enter the **vlan** command with an unused ID. To modify a VLAN, enter the **vlan** command for an existing VLAN.

See the "VLAN Default Configuration" section on page 7-3 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:

	Task	Command
Step 1	Enter VLAN configuration mode.	`Switch# configure terminal`
Step 2	Add an Ethernet VLAN. **Note** You cannot delete the default VLANs for the different media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005. When you delete a VLAN, any LAN 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. Use the **no** keyword to delete a VLAN.	`Switch(config)# [no] vlan vlan_ID` `Switch(config-vlan)#`
Step 3	Return to privileged EXEC mode.	`Switch(config-vlan)# end`
Step 4	Verify the VLAN configuration.	`Switch# show vlan [id | name] vlan_name`

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#
```

Configuring VLANs in VLAN Database Mode

User-configured VLANs have unique IDs from 1 to 1001. To create a VLAN, enter the **vlan** command with an unused ID. To modify a VLAN, enter the **vlan** command for an existing VLAN.

See the "VLAN Default Configuration" section on page 7-3 for a listing of the 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 procedure:

	Task	Command
Step 1	Enter VLAN configuration mode.	`Switch# vlan database`
Step 2	Add an Ethernet VLAN. **Note** You cannot delete the default VLANs for the different media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005. When you delete a VLAN, any LAN 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. Use the **no** keyword to delete a VLAN.	`Switch(vlan)# vlan vlan_ID`
Step 3	Return to privileged EXEC mode.	`Switch(vlan)# exit`
Step 4	Verify the VLAN configuration.	`Switch# show vlan [id | name] vlan_name`

This example shows how to create an Ethernet VLAN in VLAN database mode and verify the configuration:

```
Switch# vlan database
Switch(vlan)# vlan 3
VLAN 3 added:
    Name: VLAN0003
Switch(vlan)# exit
APPLY completed.
Exiting....
Switch# show vlan name VLAN0003
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------

3    VLAN0003                         active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- ------ ------
3    enet  100003     1500  -      -      -        -    0      0
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 Ensure you assign LAN interfaces to a VLAN of the proper type. Assign Fast Ethernet and 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 6-5.

CHAPTER 8

Understanding and Configuring Private VLANs

This chapter describes private VLANs on the Catalyst 4000 family 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 commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Overview of Private VLANs

Note To configure private VLANs, make sure the switch is in VTP transparent mode.

Private VLANs provide Layer 2 isolation between ports within the same private VLAN. There are three types of private VLAN ports:

- Promiscuous—A promiscuous port can communicate with all interfaces, including the community and isolated ports within a private VLAN.
- Isolated—An isolated port has complete Layer 2 separation from other ports within the same private VLAN except for the promiscuous port. Private VLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports.
- Community—Community ports communicate among themselves and with their promiscuous ports. These interfaces are isolated at Layer 2 from all other interfaces in other communities or isolated ports within their private VLAN.

Note Because trunks can support the VLANs carrying traffic between isolated, community, and promiscuous ports, isolated and community port traffic might enter or leave the switch through a trunk interface.

Private VLAN ports are associated with a set of supporting VLANs that are used to create the private VLAN structure. A private VLAN uses VLANs three ways:

- As a primary VLAN—Carries traffic from promiscuous ports to isolated, community, and other promiscuous ports in the same primary VLAN.
- As an isolated VLAN—Carries traffic from isolated ports to a promiscuous port.
- As a community VLAN—Carries traffic between community ports and to promiscuous ports. You can configure multiple community VLANs in a private VLAN.

Isolated and community VLANs are called secondary VLANs. You can extend private VLANs across multiple devices by trunking the primary, isolated, and community VLANs to other devices that support private VLANs.

In a switched environment, you can assign an individual private VLAN and associated IP subnet to each individual or common group of end stations. The end stations need to communicate with a default gateway only to gain access outside the private VLAN. With end stations in a private VLAN, you can do the following:

- Designate selected ports connected to end stations. For example, interfaces connected to servers as isolated to prevent any communication at Layer 2. Or, if the end stations are servers, this configuration prevents Layer 2 communication between the servers.
- Designate the interfaces to which the default gateway(s) and selected end stations (for example, backup servers or LocalDirector) are attached as promiscuous ports to allow all end stations access.
- Reduce VLAN and IP subnet consumption, because you can prevent traffic between end stations even though they are in the same VLAN and IP subnet.

Note A promiscuous port can service only one primary VLAN. A promiscuous port can service one isolated or many community VLANs.

With a promiscuous port, you can connect a wide range of devices as access points to a private VLAN. For example, you can connect a promiscuous port to the server port of 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 private VLAN servers from an administration workstation.

Private VLAN Configuration Guidelines

Note This release does not support the community VLANs.

Follow these guidelines to configure private VLANs:

- Set VTP to transparent mode. After you configure a private VLAN, you cannot change the VTP mode to client or server.
- Do not include VLAN 1 or VLANs 1002–1005 in the private VLAN configuration.
- Use only the private VLAN configuration commands to assign ports to primary, isolated, or community VLANs.

  Layer 2 interfaces assigned to the VLANs that you configure as primary, isolated, or community VLANs are inactive while the VLAN is part of the private VLAN configuration. Layer 2 trunk interfaces remain in the STP forwarding state.

- Configure Layer 3 VLAN interfaces only for primary VLANs.

  Layer 3 VLAN interfaces for isolated and community VLANs are inactive while the VLAN is configured as an isolated or community VLAN.

- Do not configure private VLAN ports as EtherChannel.

  While a port is part of the private VLAN configuration, any EtherChannel configuration for it is inactive.

- Do not configure a destination SPAN port as a private VLAN port.

  While a port is part of the private VLAN configuration, any destination SPAN configuration for it is inactive.

- You can configure a private VLAN port as a SPAN source port.
- You can use VLAN-based SPAN (VSPAN) on primary, isolated, and community VLANs, or use SPAN on only one VLAN to separately monitor egress or ingress traffic.
- A primary VLAN can have one isolated VLAN and multiple community VLANs associated with it.
- An isolated or community VLAN can have only one primary VLAN associated with it.
- Enable PortFast and BPDU guard on isolated and community ports to prevent spanning tree loops due to misconfigurations.

  When enabled, STP applies the BPDU guard feature to all PortFast-configured Layer 2 LAN ports.

- If you delete a VLAN used in the private VLAN configuration, the private VLAN ports associated with the VLAN become inactive.
- Private VLAN ports do not have to be on the same network device as long as the devices are trunk connected and the primary and secondary VLANs have not been removed from the trunk.
- VTP does not support private VLANs. You must configure private VLANs on each device where you want private VLAN ports.
- To maintain the security of your private VLAN configuration and avoid other use of the VLANs configured as private VLANs, configure private VLANs on all intermediate devices, even if devices that have no private VLAN ports.
- We recommend that you prune the private VLANs from the trunks on devices that carry no traffic in the private VLANs.
- You can apply different quality of service (QoS) configuration to primary, isolated, and community VLANs.
- To apply Cisco IOS output ACLs to all outgoing private VLAN traffic, configure them on the Layer 3 VLAN interface of the primary VLAN.
- Cisco IOS ACLs applied to the Layer 3 VLAN interface of a primary VLAN automatically apply to the associated isolated and community VLANs.
- Do not apply Cisco IOS ACLs to isolated or community VLANs.

  Cisco IOS ACL configuration applied to isolated and community VLANs is inactive while the VLANs are part of the private VLAN configuration.

- Do not apply dynamic access control entries (ACEs) to primary VLANs.

  Cisco IOS dynamic ACL configuration applied to a primary VLAN is inactive while the VLAN is part of the private VLAN configuration.

- You can stop Layer 3 switching on an isolated VLAN by deleting the mapping of that VLAN with its primary VLAN.

Configuring Private VLANs

To configure a private VLAN, follow this procedure:

Step 1 Set VTP mode to transparent. See the "Disabling VTP (VTP Transparent Mode)" section on page 9-9.

Step 2 Create the secondary VLANs (isolated or community VLANs). See the "Configuring a VLAN as a Private VLAN" section on page 8-5.

Note Isolated and community VLANs are called *secondary* VLANs.

Step 3 Create the primary VLAN. See the "Configuring a VLAN as a Private VLAN" section on page 8-5.

Step 4 Associate the secondary VLAN to the primary VLAN. See the "Associating Secondary VLANs with a Primary VLAN" section on page 8-5.

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 to an isolated or community port. See the "Configuring a Layer 2 Interface as a Private VLAN Host Port" section on page 8-7.

Step 6 Associate the isolated port or community port to the primary-secondary VLAN pair. See the "Associating Secondary VLANs with a Primary VLAN" section on page 8-5.

Step 7 Configure an interface as a promiscuous port. See the "Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port" section on page 8-6.

Step 8 Map the promiscuous port to the primary-secondary VLAN pair. See the "Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port" section on page 8-6.

These sections describe how to configure private VLANs:

- "Configuring a VLAN as a Private VLAN" section on page 8-5
- "Associating Secondary VLANs with a Primary VLAN" section on page 8-5
- "Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port" section on page 8-6
- "Configuring a Layer 2 Interface as a Private VLAN Host Port" section on page 8-7
- "Permitting Routing of Secondary VLAN Ingress Traffic" section on page 8-8

Configuring a VLAN as a Private VLAN

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

	Task	Command
Step 1	Enter configuration mode.	`Switch# configure terminal`
Step 2	Configure a VLAN as a private VLAN. • Use the **no** keyword to clear private VLAN status. • The command does not take effect until you exit VLAN configuration submode.	`Switch(config)# vlan vlan_ID` `Switch(config-vlan)# [no] private-vlan {isolated | primary}`
Step 3	Exit configuration mode.	`Switch(config-vlan)# end`
Step 4	Verify the configuration.	`Switch# show vlan private-vlan [type]`

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 type

Primary Secondary Type              Interfaces
------- --------- ----------------- ----------------------------------------
202               primary
```

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 type

Primary Secondary Type              Interfaces
------- --------- ----------------- ----------------------------------------
202               primary
440               isolated
```

Associating Secondary VLANs with a Primary VLAN

To associate secondary VLANs with a primary VLAN, perform this procedure:

	Task	Command
Step 1	Enter configuration mode.	`Switch# configure terminal`
Step 2	Enter VLAN configuration mode for the primary VLAN.	`Switch(config)# vlan primary_vlan_ID`
Step 3	Associate the secondary VLAN with the primary VLAN. The list can contain only one VLAN. Use the **no** keyword to delete all associations from the primary VLAN.	`Switch(config-vlan)# [no] private-vlan association {secondary_vlan_list | add secondary_vlan_list | remove secondary_vlan_list}`

	Task	Command
Step 4	Exit VLAN configuration mode.	`Switch(config-vlan)# end`
Step 5	Verify the configuration.	`Switch# show vlan private-vlan [type]`

When you associate secondary VLANs with a primary VLAN, note the following:

- The *secondary_vlan_list* parameter contains only one isolated VLAN ID.
- Use the **remove** keyword with the *secondary_vlan_list* variable to clear the association between the secondary VLAN and the primary VLAN. The list can contain only one VLAN.
- Use the **no** keyword to clear all associations from the primary VLAN.
- The command does not take effect until you exit VLAN configuration submode.

This example shows how to associate isolated VLAN 440 with primary VLAN 202 and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 202
Switch(config-vlan)# private-vlan association 440
Switch(config-vlan)# end
Switch# show vlan private-vlan

Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202     440       isolated
```

Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port

To configure a Layer 2 interface as a private VLAN promiscuous port, perform this procedure:

	Task	Command
Step 1	Enter configuration mode.	`Switch# configure terminal`
Step 2	Select the LAN interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 3	Configure a Layer 2 interface as a private VLAN promiscuous port.	`Switch(config-if)# switchport mode private-vlan {host | promiscuous}`
Step 4	Map the private VLAN promiscuous port to a primary VLAN and to selected secondary VLANs. Use the **no** keyword to delete all associations from the primary VLAN.	`Switch(config-if)# [no] switchport private-vlan mapping primary_vlan_ID {secondary_vlan_list | add secondary_vlan_list | remove secondary_vlan_list}`
Step 5	Exit configuration mode.	`Switch(config-if)# end`
Step 6	Verify the configuration.	`Switch# show interfaces {fastethernet | gigabitethernet} slot/port switchport`

When you configure a Layer 2 interface as a private VLAN 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 private VLAN ID or a hyphenated range of private VLAN IDs.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to map the secondary VLANs to the private VLAN promiscuous port.

- Use the **remove** keyword with a *secondary_vlan_list* to clear the mapping between secondary VLANs and the private VLAN promiscuous port.
- Use the **no** keyword to clear all mapping from the private VLAN promiscuous port.

This example shows how to configure interface FastEthernet 5/2 as a private VLAN promiscuous port, map it to a private VLAN, 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 202 440
Switch(config-if)# end
Switch# show interfaces fastethernet 5/2 switchport
Name: Fa5/2
Switchport: Enabled
Administrative Mode: private-vlan promiscuous
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: none ((Inactive))
Administrative private-vlan mapping: 202 (VLAN0202) 440 (VLAN0440)
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

Configuring a Layer 2 Interface as a Private VLAN Host Port

To configure a Layer 2 interface as a private VLAN host port, perform this procedure:

	Task	Command
Step 1	Enter configuration mode.	`Switch# configure terminal`
Step 2	Select the LAN port to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 3	Configure a Layer 2 interface as a private VLAN host port.	`Switch(config-if)# switchport mode private-vlan {host | promiscuous}`
Step 4	Associate the Layer 2 interface with a private VLAN. Use the **no** keyword to delete all associations from the primary VLAN.	`Switch(config-if)# [no] switchport private-vlan host-association primary_vlan_ID secondary_vlan_ID`
Step 5	Exit configuration mode.	`Switch(config-if)# end`
Step 6	Verify the configuration.	`Switch# show interfaces {fastethernet | gigabitethernet} slot/port switchport`

This example shows how to configure interface FastEthernet 5/1 as a private VLAN 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: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: 202 (VLAN0202)
Administrative private-vlan mapping: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

Permitting Routing of Secondary VLAN Ingress Traffic

To permit routing of secondary VLAN ingress traffic, perform this procedure:

	Task	Command
Step 1	Enter configuration mode.	`Switch# configure terminal`
Step 2	Enter interface configuration mode for the primary VLAN.	`Switch(config)# interface vlan primary_vlan_ID`
Step 3	To permit routing on the secondary VLAN ingress traffic, map the secondary VLAN to the primary VLAN. Use the **no** keyword to delete all associations from the primary VLAN.	`Switch(config-if)# [no] private-vlan mapping primary_vlan_ID {secondary_vlan_list | add secondary_vlan_list | remove secondary_vlan_list}`
Step 4	Exit configuration mode.	`Switch(config-if)# end`
Step 5	Verify the configuration.	`Switch# show interface private-vlan mapping`

When you permit routing on the secondary VLAN ingress traffic, note the following:

- Enter a value for the *secondary_vlan_list* variable or use the **add** keyword with the *secondary_vlan_list* variable to map the secondary VLANs to the primary VLAN.
- Use the **remove** keyword with the *secondary_vlan_list* variable to clear the mapping between secondary VLANs and the primary VLAN.
- Use the **no** keyword to clear all mapping from the primary VLAN.

This example shows how to permit routing of secondary VLAN ingress traffic from private VLAN 440 and verify the configuration:

```
Switch# configure terminal
Switch(config)# interface vlan 202
Switch(config-if)# private-vlan mapping add 440
Switch(config-if)# end
Switch# show interfaces private-vlan mapping
Interface Secondary VLAN Type
--------- -------------- -----------------
vlan202   440            isolated

Switch#
```

CHAPTER 9

Understanding and Configuring VTP

This chapter describes the VLAN Trunking Protocol (VTP) on the Catalyst 4000 family 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 commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

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.

Note For complete information on configuring VLANs, see Chapter 7, "Understanding and Configuring VLANs."

The following sections describe how VTP works:

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 4000 family switch is in VTP server mode and is in the no-management domain state until the switch receives an advertisement for a domain over a trunk link or you configure a management domain. 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 4000 family switch to operate in any one of these VTP modes:

- Server—In VTP server mode, you can create, modify, and delete VLANs and specify other configuration parameters (such as VTP version and VTP pruning) for the entire VTP domain. VTP servers advertise their VLAN configuration to other network devices in the same VTP domain and synchronize their VLAN configuration with other network devices based on advertisements received over trunk links. VTP server is the default mode.
- Client—VTP clients behave the same way as VTP servers, but you cannot create, change, or delete VLANs on a VTP client.
- Transparent—VTP transparent network devices do not participate in VTP. A VTP transparent network device does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements. However, in VTP version 2, transparent network devices do forward VTP advertisements that they receive on their trunking LAN interfaces.

Note Catalyst 4000 family 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 4000 family 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 9-1 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 4000 family switch (see the "Enabling VTP Pruning" section on page 9-6).

Figure 9-1 Flooding Traffic without VTP Pruning

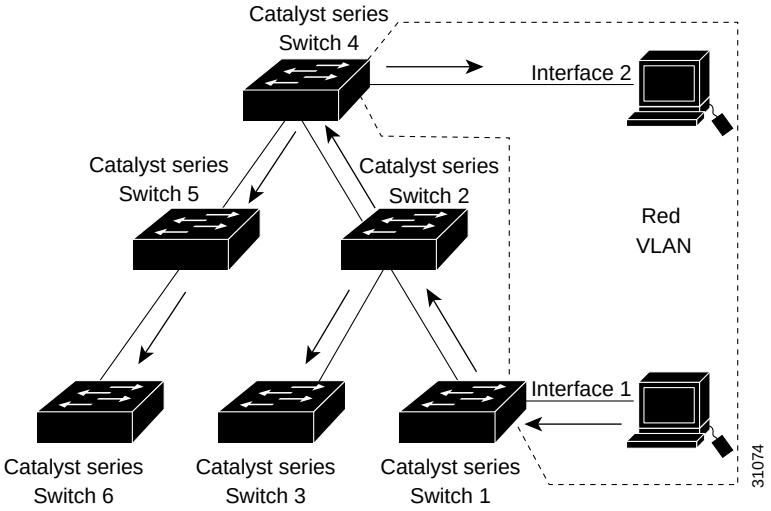
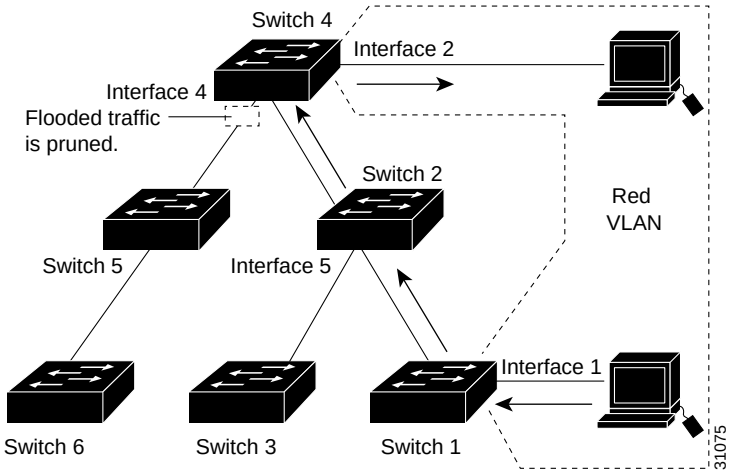


Figure 9-2 shows the same switched network with VTP pruning enabled. The broadcast traffic from Switch 1 is not forwarded to Switches 3, 5, and 6 because traffic for the Red VLAN has been pruned on the links indicated (Interface 5 on Switch 2 and Interface 4 on Switch 4).

Figure 9-2 Flooding Traffic with VTP Pruning



Enabling VTP pruning on a VTP server enables pruning for the entire management domain. VTP pruning takes effect several seconds after you enable it. By default, VLANs 2 through 1000 are eligible for pruning. VTP pruning does not prune traffic from pruning-ineligible VLANs. VLAN 1 is always eligible 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 4000 family switch affects pruning eligibility for those VLANs on that switch only, not on all network devices in the VTP domain.

VTP Default Configuration

Table 9-1 shows the default VTP configuration.

Table 9-1 VTP Default Configuration

Feature	Default Value
VTP domain name	Null
VTP mode	Server
VTP version 2 enable state	Version 2 is disabled
VTP password	None
VTP pruning	Disabled

Configuring VTP

The following sections describe how to configure VTP:

Configuring VTP Global Parameters

The following sections describe configuring the VTP global parameters:

Configuring a VTP Password

To configure the VTP global parameters, enter the following command:

Purpose	Command
Sets a password for the VTP domain. The password can be from 8 to 64 characters. Use the **no** keyword to remove the password.	`Switch# [no] vtp password password_string`

This example shows how to configure a VTP password:

```
Switch# vtp password WATER
Setting device VLAN database password to WATER.
Switch#
```

Enabling VTP Pruning

To enable VTP pruning in the management domain, perform this procedure:

	Task	Command
Step 1	Enable VTP pruning in the management domain. Use the **no** keyword to disable VTP pruning in the management domain.	`Switch# [no] vtp pruning`
Step 2	Verify the configuration.	`Switch# show vtp status`

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 procedure:

	Task	Command
Step 1	Enable VTP version 2. Use the **no** keyword to revert to the default.	`Switch# [no] vtp version {1 | 2}`
Step 2	Verify the configuration.	`Switch# show vtp status`

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 4000 family switch as a VTP server, perform this procedure:

	Task	Command
Step 1	Enter configuration mode.	`Switch# configuration terminal`
Step 2	Configure the switch as a VTP server.	`Switch(config)# vtp mode server`
Step 3	Define the VTP domain name, which can be up to 32 characters long.	`Switch(config)# vtp domain domain_name`

	Task	Command
Step 4	Exit VLAN configuration mode.	`Switch(config)# end`
Step 5	Verify the configuration.	`Switch# show vtp status`

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

Configuring the Switch as a VTP Client

To configure the Catalyst 4000 family switch as a VTP client, perform this procedure:

	Task	Command
Step 1	Enter configuration mode.	`Switch# configuration terminal`
Step 2	Configure the switch as a VTP client. Use the **no** keyword to return to the default setting (server mode).	`Switch(config)# [no] vtp mode client`
Step 3	Exit configuration mode.	`Switch(config)# end`
Step 4	Verify the configuration.	`Switch# show vtp status`

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                     : 2
```

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

Disabling VTP (VTP Transparent Mode)

To disable VTP on the Catalyst 4000 family switch, perform this procedure:

	Task	Command
Step 1	Enter configuration mode.	`Switch# configuration terminal`
Step 2	Disable VTP on the switch.Use the **no** keyword to return to the default setting (server mode).	`Switch(config)# [no] vtp mode transparent`
Step 3	Exit configuration mode.	`Switch(config)# end`
Step 4	Verify the configuration.	`Switch# show vtp status`

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, enter the following command:

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
```

CHAPTER 10

Configuring Layer 3 Interfaces

This chapter describes the Layer 3 interfaces on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Overview of Layer 3 Interfaces

The Catalyst 4006 switch with Supervisor Engine III 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 4006 switch with Supervisor Engine III 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 4006 switch with Supervisor Engine III to be configured like a traditional router.

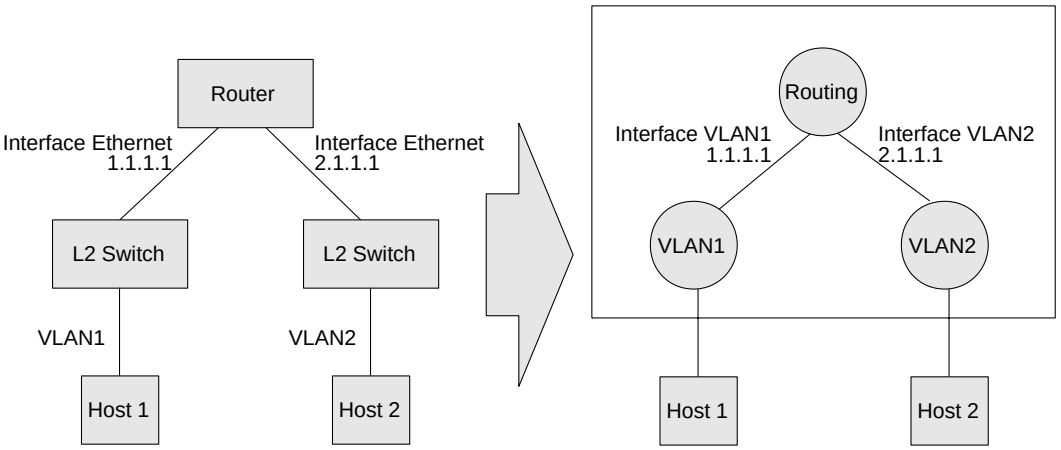
This section contains the following subsections:

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 4006 switch with Supervisor Engine III supports inter-VLAN routing by integrating the routing and bridging functions on a single Catalyst 4006 switch with Supervisor Engine III.

Figure 10-1 shows how the routing and bridging functions in the three physical devices of the traditional network are performed logically on one Catalyst 4006 switch with Supervisor Engine III.

Figure 10-1 Logical Layer 3 VLAN Interfaces for the Catalyst 4006 Switch with Supervisor Engine III

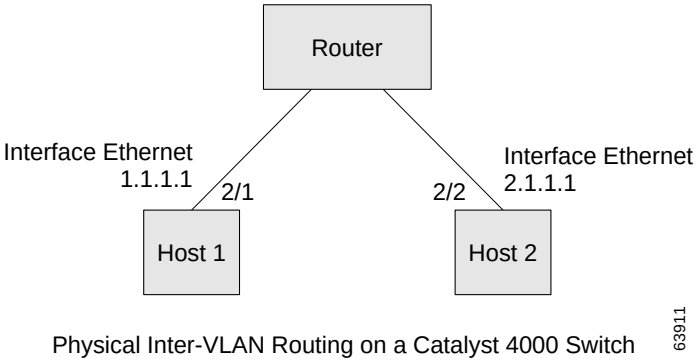


Physical Layer 3 Interfaces

The physical Layer 3 interfaces support capabilities equivalent to a traditional router. These Layer 3 interfaces provide hosts with physical routing interfaces to the Catalyst 4006 switch with Supervisor Engine III.

Figure 10-2 shows how the Catalyst 4006 switch with Supervisor Engine III functions as a traditional router.

Figure 10-2 Physical Layer 3 Interfaces for the Catalyst 4006 with Supervisor Engine III



Physical Inter-VLAN Routing on a Catalyst 4000 Switch

Configuration Guidelines

The Catalyst 4006 switch with Supervisor Engine III does not support:

- Subinterfaces or the **encapsulation** keyword on Layer 3 Fast Ethernet or Gigabit Ethernet interfaces
- IPX routing and Appletalk routing

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 procedure:

	Task	Command
Step 1	Create the VLAN.	`Switch(config)# vlan vlan_ID`
Step 1	Select an interface to configure.	`Switch(config)# interface vlan vlan_ID`
Step 2	Configure the IP address and IP subnet.	`Switch(config-if)# ip address ip_address subnet_mask`
Step 3	Enable the interface.	`Switch(config-if)# no shutdown`
Step 4	Exit configuration mode.	`Switch(config-if)# end`
Step 5	Save your configuration changes to NVRAM.	`Switch# copy running-config startup-config`

Step 6	Verify the configuration.	`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`

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 10.255.255.248
!
ip classless
no ip http server
!
!
line con 0
line aux 0
line vty 0 4
!
end
```

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 procedure:

	Task	Command
Step 1	Enable IP routing (Required only if disabled.)	`Switch(config)#ip routing`
Step 1	Select an interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port} | {port-channel port_channel_number}`
Step 2	Convert this port from physical Layer 2 port to physical Layer 3 port.	`Switch(config-if)#no switchport`
Step 3	Configure the IP address and IP subnet.	`Switch(config-if)# ip address ip_address subnet_mask`
Step 4	Enable the interface.	`Switch(config-if)# no shutdown`
Step 5	Exit configuration mode.	`Switch(config-if)# end`
Step 6	Save your configuration changes to NVRAM.	`Switch# copy running-config startup-config`

Step 7	Verify the configuration.	`Switch# show interfaces [type slot/interface]`
		`Switch# show ip interfaces [type slot/interface]`
		`Switch# show running-config interfaces [type slot/interface]`

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 10.255.255.248
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 10.255.255.248
!
…
ip classless
no ip http server
!
!
line con 0
```

```
line aux 0
line vty 0 4
!
end
```

CHAPTER 11

Understanding and Configuring EtherChannel

This chapter describes how to use the command-line interface (CLI) to configure EtherChannel on the Catalyst 4006 switch with Supervisor Engine III Layer 2 or Layer 3 interfaces. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note The commands in the following sections can be used on all Ethernet interfaces in a Catalyst 4006 switch with Supervisor Engine III, including the uplink ports on the supervisor engine.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Overview of EtherChannel

EtherChannel bundles individual Ethernet links into a single logical link that provides bandwidth up to 1600 Mbps (Fast EtherChannel full duplex) or 16 Gbps (Gigabit EtherChannel) between a Catalyst 4006 switch with Supervisor Engine III and another switch or host.

A Catalyst 4006 switch with Supervisor Engine III supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module and across modules in a Catalyst 4006 switch with Supervisor Engine III. All interfaces in each EtherChannel must be the same speed and must all be configured as either Layer 2 or Layer 3 interfaces.

Note The network device to which a Catalyst 4006 switch with Supervisor Engine III 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. Once 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.

The following sections describe how EtherChannel works:

Understanding 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.

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).

Understanding the Port Aggregation Protocol

The Port Aggregation Protocol (PAgP) expedites the automatic creation of EtherChannels by exchanging packets between Ethernet interfaces. PAgP packets are exchanged only between interfaces in **auto** and **desirable** modes. Interfaces configured in the **on** mode do not exchange PAgP packets.

The protocol learns the capabilities of interface groups dynamically and informs the other interfaces. When PAgP identifies correctly matched Ethernet links, it groups the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

EtherChannel includes three user-configurable modes: **on**, **auto**, and **desirable** (see Table 11-1). Only **auto** and **desirable** are PAgP modes.

Table 11-1 EtherChannel Modes

Mode	Description
on	Mode that forces the interface to channel without PAgP. In the **on** mode, a usable EtherChannel exists only when an interface group in the **on** mode is connected to another interface group in the **on** mode.
auto	PAgP mode that places an interface in a passive negotiating state in which the interface responds to PAgP packets it receives but does not initiate PAgP negotiation.
desirable	PAgP mode that places an interface in an active negotiating state in which the interface initiates negotiations with other interfaces by sending PAgP packets.

Both the **auto** and **desirable** modes allow interfaces to negotiate with partner interfaces to determine whether they can form an EtherChannel, based on criteria such as interface speed and, for Layer 2 EtherChannels, trunking state and VLAN numbers.

Interfaces between two connected switches can form an EtherChannel when they are in different PAgP modes as long as the modes are compatible. This compatibility of the modes is shown in Table 11-2.

Table 11-2 Compatibility of EtherChannel Modes

	desirable	auto	on
desirable	Yes	Yes	No
auto	Yes	No	No
on	No	No	Yes

Understanding Load Balancing

EtherChannel balances traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel.

EtherChannel load balancing can use MAC addresses, IP addresses, or Layer 4 port numbers; either source or destination or both source and destination.

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 operates at the switch level rather than per-channel, applying globally for all channels on the switch.

For additional information on load balancing, see the "Configuring EtherChannel Load Balancing" section on page 11-9.

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. If you shut down an interface in an EtherChannel, it is treated as a link failure and its traffic is transferred to one of the remaining interfaces in the EtherChannel.
- An EtherChannel will 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 is the same on all the trunks. Interfaces in an EtherChannel with different trunk modes 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 range of VLANs is not the same, the interfaces do not form an EtherChannel even when set to the **auto** or **desirable** mode.
  - 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 where you apply the configuration.

Configuring EtherChannel

These sections describe how to configure EtherChannel:

Note Ensure that the interfaces are configured correctly (see the "EtherChannel Configuration Guidelines and Restrictions" section on page 11-3).

Configuring Layer 3 EtherChannels

To configure Layer 3 EtherChannels, create the port-channel logical interface and then put the Ethernet interfaces into the port-channel.

These sections describe Layer 3 EtherChannel configuration:

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 procedure:

	Task	Command
Step 1	Create the port-channel interface. The value for *port_channel_number* can range from 1 to 64	`Switch(config)# interface port-channel port_channel_number`
Step 2	Assign an IP address and subnet mask to the EtherChannel.	`Switch(config-if)# ip address ip_address mask`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show running-config interface port-channel port_channel_number`

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 procedure for each interface:

	Task	Command
Step 1	Select a physical interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 2	Make this a Layer 3 routed port.	`Switch(config-if)# no switchport`
Step 3	Ensure that there is no IP address assigned to the physical interface.	`Switch(config-if)# no ip address`
Step 4	Configure the interface in a port-channel and specify the PAgP mode.	`Switch(config-if)# channel-group port_channel_number mode {auto | desirable | on}`

	Task	Command
Step 5	Exit configuration mode.	`Switch(config-if)# end`

Step 6	Verify the configuration.	`Switch# show running-config interface port-channel port_channel_number`
		
		`Switch# show running-config interface {fastethernet | gigabitethernet} slot/port`
		
		`Switch# show interfaces {fastethernet | gigabitethernet} slot/port etherchannel`
		
		`Switch# show etherchannel 1 port-channel`

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 4-4 for information about the **range** keyword.

The following two examples shows 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 EtherChannel

To configure Layer 2 EtherChannels, configure the Ethernet interfaces with the **channel-group** command. This creates the port-channel logical interface.

Note IOS 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 procedure for each interface:

	Task	Command
Step 1	Select a physical interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 2	Configure the interface in a port-channel and specify the PAgP mode.	`Switch(config-if)# channel-group port_channel_number mode {auto | desirable | on}`
Step 3	Exit configuration mode.	`Switch(config-if)# end`

Step 4	Verify the configuration.	`Switch# show running-config interface {fastethernet | gigabitethernet} slot/port`
		
		`Switch# show interface {fastethernet | gigabitethernet} slot/port etherchannel`

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
```

Note See the "Configuring a Range of Interfaces" section on page 4-4 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.
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 EtherChannel Load Balancing

Note Load balancing operates at the switch level rather than per-channel, applying globally for all channels on the switch.

To configure EtherChannel load balancing, perform this procedure:

	Task	Command
Step 1	Configure EtherChannel load balancing. Use the **no** keyword to return EtherChannel load balancing to the default configuration.	`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}`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show etherchannel load-balance`

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(config)#
```

This example shows how to verify the configuration:

```
Switch# show etherchannel load-balance
Source XOR Destination IP address
Switch#
```

Removing an Interface from an EtherChannel

To remove an Ethernet interface from an EtherChannel, perform this procedure:

	Task	Command
Step 1	Select a physical interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 2	Remove the interface from the port-channel interface.	`Switch(config-if)# no channel-group`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show running-config interface {fastethernet | gigabitethernet} slot/port` `Switch# show interface {fastethernet | gigabitethernet} slot/port etherchannel`

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.

To remove an EtherChannel, perform this procedure:

	Task	Command
Step 1	Remove the port-channel interface.	`Switch(config)# no interface port-channel` *port_channel_number*
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show etherchannel summary`

This example shows how to remove port-channel 1:

```
Switch# configure terminal
Switch(config)# no interface port-channel 1
Switch(config)# end
```

CHAPTER 12

Understanding and Configuring STP

This chapter describes how to configure the spanning tree protocol (STP) on Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note For information on configuring the PortFast, UplinkFast, and BackboneFast spanning tree enhancements, see Chapter 13, "Configuring STP Features."

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/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. Spanning tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

Catalyst 4000 family switches use the STP (the IEEE 802.1D bridge protocol) on all VLANs. By default, a single instance of spanning tree runs on each configured VLAN (provided you do not manually disable spanning tree). You can enable and disable 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.

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. 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.

The following sections describe how STP works:

Bridge Protocol Data Units

The following items 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 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 bridge and MAC addresses, bridge priority, port priority, and path cost. To communicate and compute the spanning tree topology, BPDUs are transmitted from each switch (configuration BPDUs) and in one direction from the root switch. Each configuration BPDU contains at least the following information:

- 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
- Message age
- 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 switch.
- The shortest distance to the root switch 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 switch. 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 switch.

The spanning tree root switch is the logical center of the spanning tree topology in a switched network. All paths that are not required to reach the root switch from anywhere in the switched network are placed in spanning tree blocking mode.

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 12-1 describes the STP timers that affect the entire spanning tree performance.

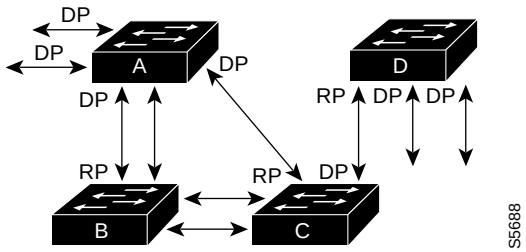
Table 12-1 Spanning Tree Protocol Timers

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

Creating the STP Topology

In Figure 12-1, Switch A is elected as the root bridge because 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 12-1 Spanning Tree Topology



RP = Root Port
DP = Designated Port

When the spanning tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change. The goal is to make the fastest link the root port.

For example, assume that one port on Switch B is a fiber-optic link, and another port on Switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the 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 using the old topology.

Each Layer 2 interface on a switch using spanning tree exists in one of the following five states:

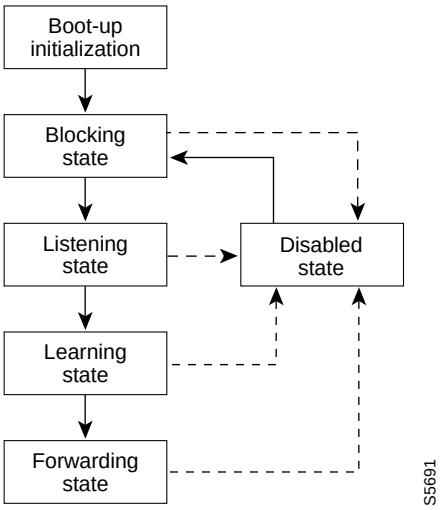
- Blocking—In this state, the Layer 2 interface does not participate in frame forwarding
- Listening—First transitional state after the blocking state when spanning tree determines that the Layer 2 interface should participate in frame forwarding
- Learning—In this state, the Layer 2 interface prepares 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

A Layer 2 interface moves through these five states as follows:

- From initialization to blocking
- From blocking to listening or to disabled
- From listening to learning or to disabled
- From learning to forwarding or to disabled
- From forwarding to disabled

Figure 12-2 illustrates how a Layer 2 interface moves through the five states.

Figure 12-2 Spanning Tree Layer 2 Interface States



When you enable spanning tree, every port in the switch, VLAN, or network goes through the blocking state and the transitory states of listening and learning at power up. If properly configured, each Layer 2 interface stabilizes to the forwarding or blocking state.

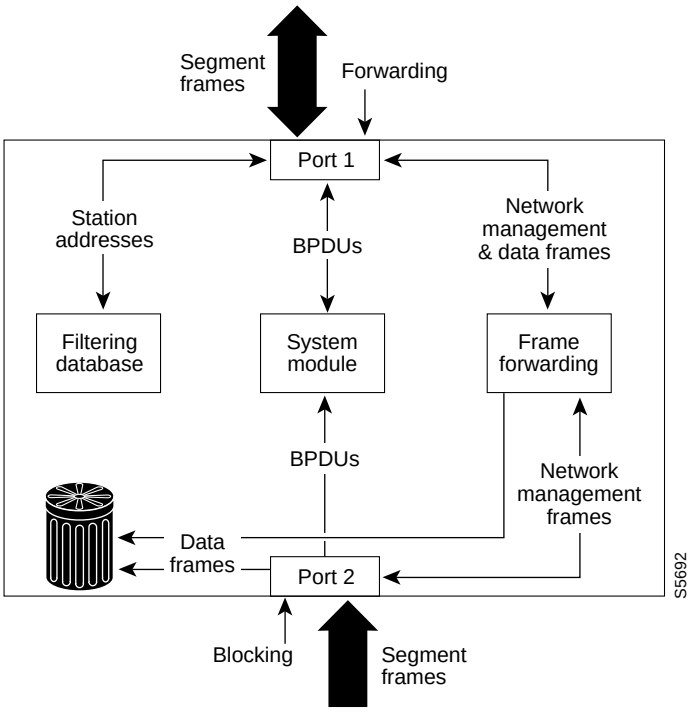
When the spanning tree algorithm places a Layer 2 interface in the forwarding state, the following process occurs:

1. The Layer 2 interface is put into the listening state while it waits for protocol information that suggests it should go to the blocking state.
2. The Layer 2 interface waits for the forward delay timer to expire, moves the Layer 2 interface to the learning state, and resets the forward delay timer.
3. In the learning state, the Layer 2 interface continues to block frame forwarding as it learns end station location information for the forwarding database.
4. The Layer 2 interface waits for the forward delay timer to expire and then moves the Layer 2 interface to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A Layer 2 interface in the blocking state does not participate in frame forwarding, as shown in Figure 12-3. After initialization, a BPDU is sent out to each Layer 2 interface in the switch. A switch initially assumes it is the root until it exchanges BPDUs with other switches. This exchange establishes which switch in the network is the root or root bridge. If only one switch is in the network, no exchange occurs, the forward delay timer expires, and the ports move to the listening state. A port always enters the blocking state following switch initialization.

Figure 12-3 Layer 2 Interface in Blocking State



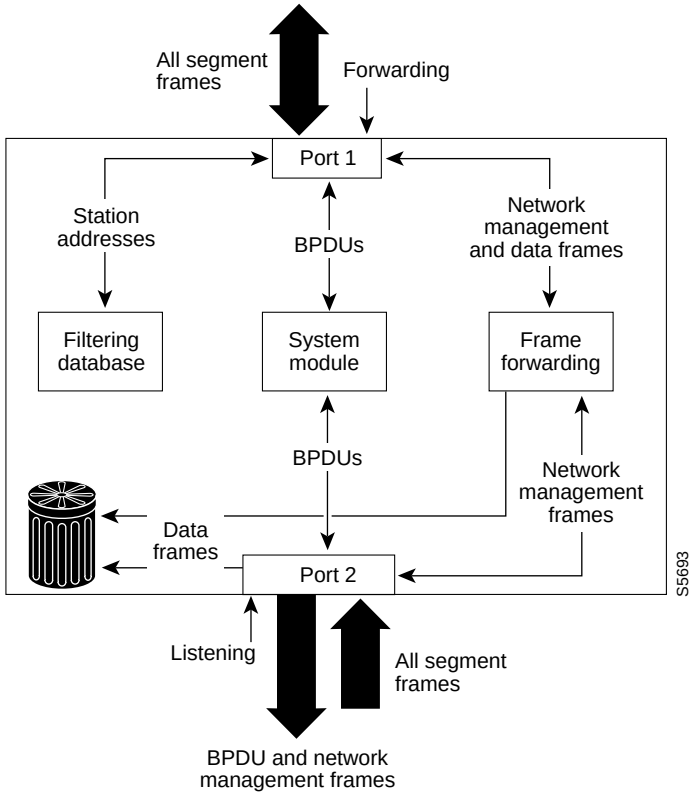
A Layer 2 interface in the blocking state behaves as follows:

- Discards frames received from the attached segment
- Discards frames switched from another interface for forwarding
- Does not incorporate end station location into its address database (there is no learning on a blocking Layer 2 interface, so there is no address database update)
- Receives BPDUs and directs them to the system module
- Does not transmit BPDUs received from the system module
- Receives and responds to network management messages

Listening State

The listening state is the first transitional state that a Layer 2 interface enters after the blocking state. The Layer 2 interface enters this state when spanning tree determines that the Layer 2 interface should participate in frame forwarding. Figure 12-4 shows a Layer 2 interface in the listening state.

Figure 12-4 Layer 2 Interface in Listening State



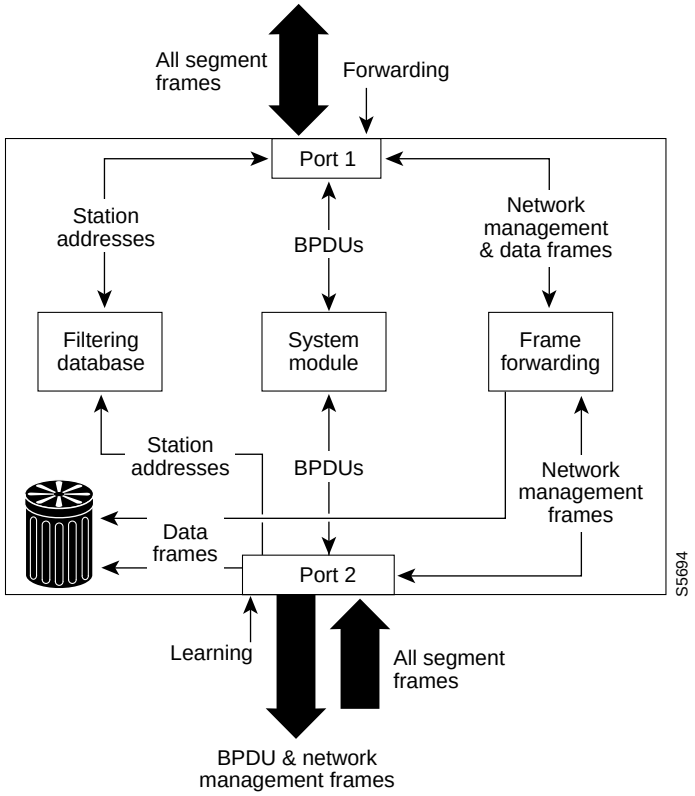
A Layer 2 interface in the listening state behaves as follows:

- Discards frames received from the attached segment
- Discards frames switched from another interface for forwarding
- Does not incorporate end station location into its address database (there is no learning at this point, so there is no address database update)
- Receives BPDUs and directs them to the system module
- Receives, processes, and transmits BPDUs received from the system module
- Receives and responds to network management messages

Learning State

A Layer 2 interface in the learning state prepares to participate in frame forwarding. The Layer 2 interface enters the learning state from the listening state. Figure 12-5 shows a Layer 2 interface in the learning state.

Figure 12-5 Layer 2 Interface in Learning State



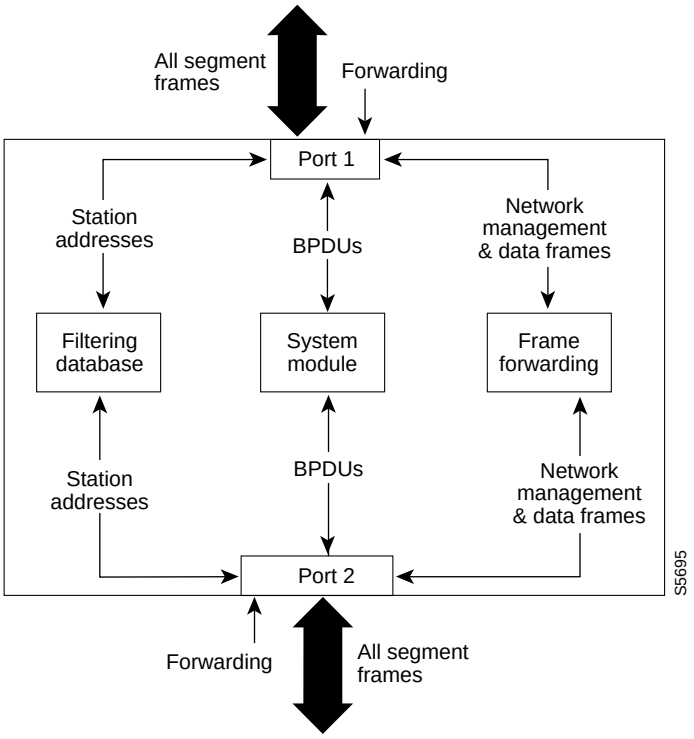
A Layer 2 interface in the learning state behaves as follows:

- Discards frames received from the attached segment
- Discards frames switched from another interface for forwarding
- Incorporates end station location into its address database
- Receives BPDUs and directs them to the system module
- Receives, processes, and transmits BPDUs received from the system module
- Receives and responds to network management messages

Forwarding State

A Layer 2 interface in the forwarding state forwards frames, as shown in Figure 12-6. The Layer 2 interface enters the forwarding state from the learning state.

Figure 12-6 Layer 2 Interface in Forwarding State



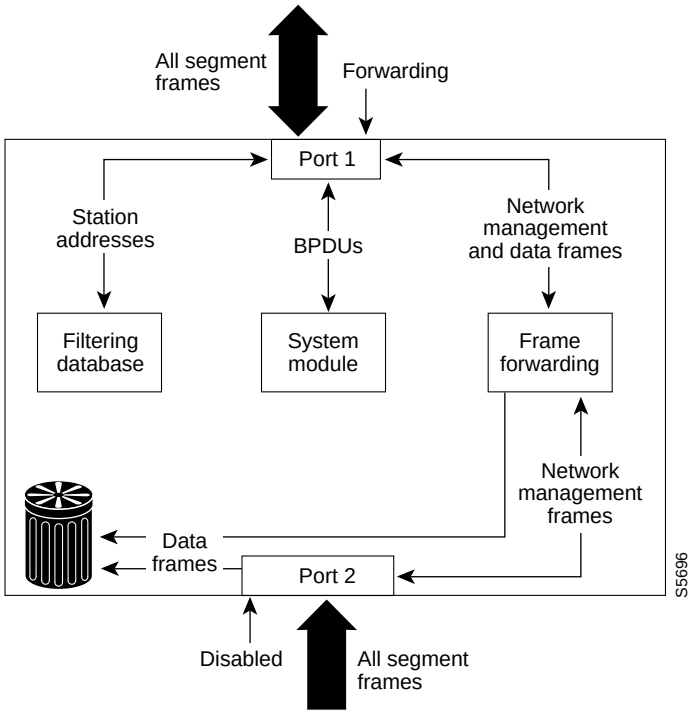
A Layer 2 interface in the forwarding state behaves as follows:

- Forwards frames received from the attached segment
- Forwards frames switched from another Layer 2 interface for forwarding
- Incorporates end station location information into its address database
- Receives BPDUs and directs them to the system module
- Processes BPDUs received from the system module
- Receives and responds to network management messages

Disabled State

A Layer 2 interface in the disabled state does not participate in frame forwarding or spanning tree, as shown in Figure 12-7. A Layer 2 interface in the disabled state is virtually nonoperational.

Figure 12-7 Layer 2 Interface in Disabled State



A disabled Layer 2 interface behaves as follows:

- Discards frames received from the attached segment
- Discards frames switched from another Layer 2 interface for forwarding
- Does not incorporate end station location into its address database (there is no learning, so there is no address database update)
- Does not receive BPDUs
- Does not receive BPDUs for transmission from the system module

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 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.

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 6, "Configuring Layer 2 Ethernet Interfaces."

Default STP Configuration

Table 12-2 shows the default spanning tree configuration.

Table 12-2 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)	• Gigabit Ethernet: 4 • Fast Ethernet: 19 • Fast Ethernet 10/100: 19
Spanning tree VLAN port priority value (configurable on a per-VLAN basis—used on interfaces configured as Layer 2 trunk ports)	128
Spanning tree VLAN port cost (configurable on a per-VLAN basis—used on interfaces configured as Layer 2 trunk ports)	• 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:

Note The spanning tree commands described in this chapter can be configured on any interface, except interfaces that are configured with the **no switchport** command.

Enabling STP

Note By default, spanning tree is enabled on all the VLANs.

You can enable 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 disable spanning tree).

To enable spanning tree on a per-VLAN basis, perform this procedure:

	Task	Command
Step 1	Enable spanning tree for VLAN *vlan_id*. The *vlan_ID* value can be from 1 to 1005.	`Switch(config)# spanning-tree vlan vlan_ID`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify that spanning tree is enabled.	`Switch# show spanning-tree vlan vlan_ID`

This example shows how to enable 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#
```

Configuring the Root Bridge

A Catalyst 4006 switch with Supervisor Engine III 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 re-computation 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, let's 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 procedure:

	Task	Command
Step 1	Configure a switch as the root switch. You can use the **no** keyword to restore the defaults.	`Switch(config)# [no] spanning-tree vlan vlan_ID root primary [diameter hops [hello-time seconds]]`
Step 2	Exit configuration mode.	`Switch(config)# end`

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#
```

The following 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 Observe that the bridge priority is now set at 8192, making this switch 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 procedure:

	Task	Command
Step 1	Configure a switch as the secondary root switch. You can use the **no** keyword to restore the defaults.	`Switch(config)# [no] spanning-tree vlan vlan_ID root secondary [diameter hops [hello-time seconds]]`
Step 2	Exit configuration mode.	`Switch(config)# end`

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 10:

```
Switch#show spanning-tree vlan 10

 VLAN10 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 16384, address 0050.3e7c.7809
  Configured hello time 2, max age 14, forward delay 10
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set
  Number of topology changes 9 last change occurred 00:00:51 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, aging 300

 Port 149 (FastEthernet3/21) of VLAN10 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.149.
   Designated root has priority 16384, address 0050.3e7c.7809
   Designated bridge has priority 16384, address 0050.3e7c.7809
   Designated port id is 128.149, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   BPDU:sent 1718, received 2

 Port 150 (FastEthernet3/22) of VLAN10 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.150.
   Designated root has priority 16384, address 0050.3e7c.7809
   Designated bridge has priority 16384, address 0050.3e7c.7809
   Designated port id is 128.150, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   BPDU:sent 1719, received 4
```

```
Port 173 (FastEthernet3/45) of VLAN10 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.173.
   Designated root has priority 16384, address 0050.3e7c.7809
   Designated bridge has priority 16384, address 0050.3e7c.7809
   Designated port id is 128.173, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   BPDU:sent 41, received 1695

Port 174 (FastEthernet3/46) of VLAN10 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.174.
   Designated root has priority 16384, address 0050.3e7c.7809
   Designated bridge has priority 16384, address 0050.3e7c.7809
   Designated port id is 128.174, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   BPDU:sent 34, received 1686
```

Configuring STP Port Priority

In the event of a loop, 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 spanning tree to select first and lower priority values to interfaces that you want spanning tree to select last. If all interfaces have the same priority value, 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 IOS 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:

	Task	Command
Step 1	Select an interface to configure.	`Switch(config)# interface {{fastethernet |gigabitethernet} slot/port} | {port-channel port_channel_number}`
Step 2	Configure 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.	`Switch(config-if)# [no] spanning-tree port-priority port_priority`
Step 3	Configure 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.	`Switch(config-if)# [no] spanning-tree vlan vlan_ID port-priority port_priority`
Step 4	Exit configuration mode.	`Switch(config-if)# end`
Step 5	Verify the configuration.	`Switch# show spanning-tree interface {{fastethernet | gigabitethernet} slot/port} | {port-channel port_channel_number}` `show spanning-tree vlan vlan_ID`

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 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 19, 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#
```

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 procedure:

	Task	Command
Step 1	Select an interface to configure.	`Switch(config)# interface {{fastethernet |gigabitethernet} slot/port} | {port-channel port_channel_number}`
Step 2	Configure 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.	`Switch(config-if)# [no] spanning-tree cost port_cost`
Step 3	Configure 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.	`Switch(config-if)# [no] spanning-tree vlan vlan_ID cost port_cost`
Step 4	Exit configuration mode.	`Switch(config-if)# end`
Step 5	Verify the configuration.	`Switch# show spanning-tree interface {{fastethernet | gigabitethernet} slot/port} | {port-channel port_channel_number}` `show spanning-tree vlan vlan_ID`

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 procedure:

	Task	Command
Step 1	Configure the bridge priority of a VLAN. The *bridge_priority* value can be from 1 to 65,535. You can use the **no** keyword to restore the defaults.	`Switch(config)# [no] spanning-tree vlan vlan_ID priority bridge_priority`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show spanning-tree vlan vlan_ID bridge [brief]`

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 procedure:

	Task	Command
Step 1	Configure 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.	`Switch(config)# [no] spanning-tree vlan vlan_ID hello-time hello_time`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show spanning-tree vlan vlan_ID bridge [brief]`

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 procedure:

	Task	Command
Step 1	Configure 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.	`Switch(config)# [no] spanning-tree vlan vlan_ID max-age max_age`

	Task	Command
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show spanning-tree vlan vlan_ID bridge [brief]`

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, use this procedure:

	Task	Command
Step 1	Configure 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.	`Switch(config)# [no] spanning-tree vlan vlan_ID forward-time forward_time`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show spanning-tree vlan vlan_ID bridge [brief]`

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#
```

Disabling Spanning Tree Protocol

To disable spanning tree on a per-VLAN basis, perform this procedure:

	Task	Command
Step 1	Disable spanning tree on a per-VLAN basis.	`Switch(config)# no spanning-tree vlan vlan_ID`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify that spanning tree is disabled.	`Switch# show spanning-tree vlan vlan_ID`

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#
```

CHAPTER 13

Configuring STP Features

This chapter describes the Spanning Tree Protocol (STP) features supported on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note For information on configuring STP, see Chapter 12, "Understanding and Configuring STP."

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/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(s). If a port goes into the root-inconsistent state, it automatically goes into the listening state.

When a switch that has ports with root guard enabled detects a new root, the ports will go into root-inconsistent state. Then, when the switch no longer detects a new root, its ports will automatically go into the listening state.

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. If the interface receives a bridge protocol data unit (BPDU), which should not happen if the interface is connected to a single workstation or server, spanning tree puts the port into the blocking state.

Note Because the purpose of PortFast is to minimize the time 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.

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 enabled, spanning tree applies the BPDU guard feature to all PortFast-configured interfaces.

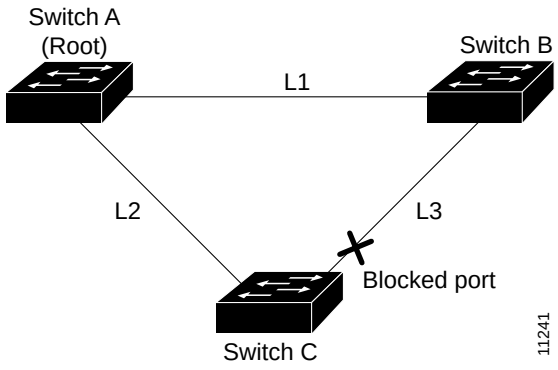
Overview of UplinkFast

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 achieves load balancing between redundant Layer 2 links using uplink groups. 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.

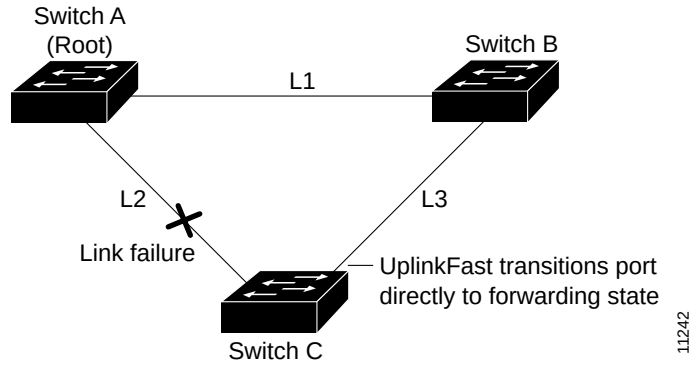
Figure 13-1 shows an example of a topology with no link failures. Switch A, the root switch, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 13-1 UplinkFast Example Before Direct Link Failure



If Switch C detects a link failure on the currently active link L2 on the root port (a *direct* link failure), UplinkFast unblocks the blocked port on Switch C and transitions it to the forwarding state without going through the listening and learning states, as shown in Figure 13-2. This switchover takes approximately one to five seconds.

Figure 13-2 UplinkFast Example After Direct Link Failure



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 is a Max Age optimization. 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 will ignore the inferior BPDU for the Max Age time.

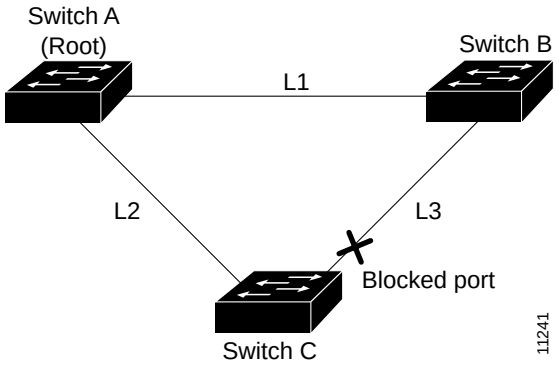
After receiving inferior BPDUs, the receiving switch will try to determine if there is an alternate path to the root bridge.

- 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 Max Age time expires, which turns the switch into the root switch.

If the switch finds an alternate path to the root bridge, it will use this new alternate path. This new path, and any other alternate paths, will be used to send a Root Link Query (RLQ) BPDU. By turning on BackboneFast, 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 13-3 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 13-3 BackboneFast Example Before Indirect Link Failure



Next, assume that L1 fails. Switch A and Switch B, the switches directly connected to this segment, instantly know that the link is down. To repair the network, it is necessary that the blocking interface on Switch C enter the forwarding state. 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 Max Age timer has expired.

Here is what happens when you use BackboneFast to eliminate the Max Age timer (20-second) delay:

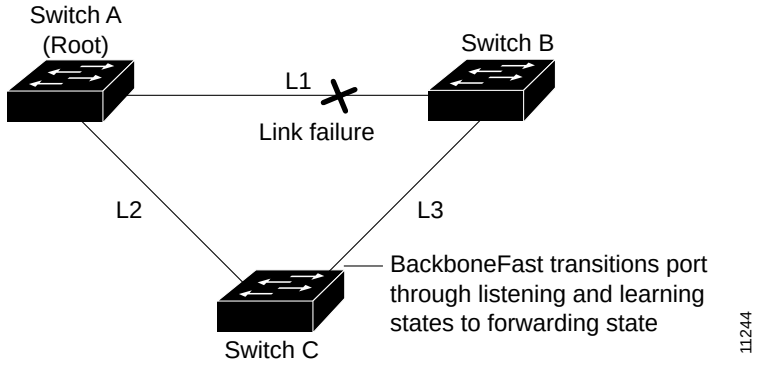
1. L1 fails.
2. Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root switch over L1, Switch B detects the failure and elects itself the root.
3. Switch B begins sending configuration BPDUs to Switch C, listing itself as the root. This is part of the normal STP behavior. The following events are specific to BackboneFast.
4. When Switch C receives the inferior configuration BPDUs from Switch B, Switch C infers that an indirect failure has occurred.
5. Switch C then sends out an RLQ request.
6. Switch A receives the RLQ request. Because Switch A is the root bridge, it replies with an RLQ response, listing itself as the root bridge.

7. 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.
8. BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the Max Age timer for the port to expire.
9. 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.

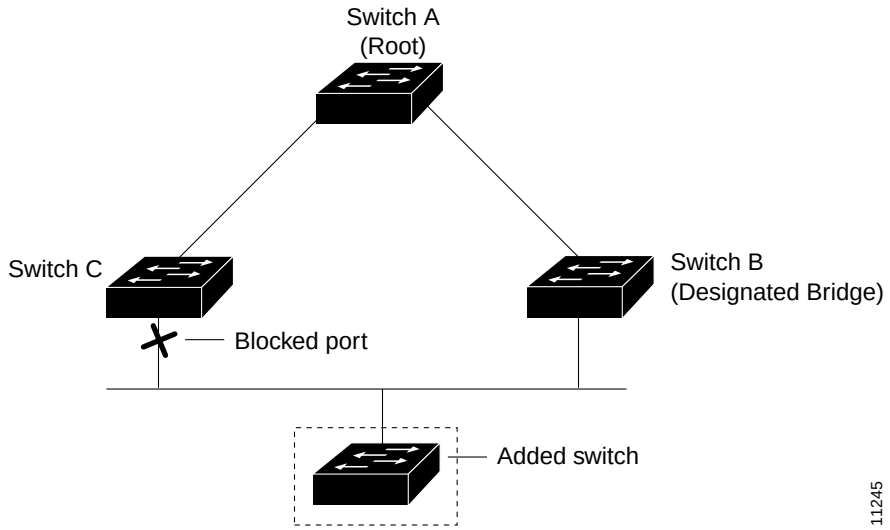
Figure 13-4 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 13-4 BackboneFast Example After Indirect Link Failure



If a new switch is introduced into a shared-medium topology as shown in Figure 13-5, 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 13-5 Adding a Switch in a Shared-Medium Topology



Enabling Root Guard

To enable Root Guard on a Layer 2 access port (to force it to become a designated port), perform this procedure:

	Task	Command
Step 1	Select an interface to configure.	`Switch(config)# interface {{fastethernet | gigabitethernet} slot/port}`
Step 2	Enable root guard. You can use the **no** keyword to disable Root Guard.	`Switch(config-if)# [no] spanning-tree guard root`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show spanning-tree`

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

```
Switch# configure terminal
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 previous 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 display ports that are in the root-inconsistent state:

```
Switch# show spanning-tree vlan 1 inconsistentports

Name                 Interface              Inconsistency
-------------------- ---------------------- ------------------
VLAN1                FastEthernet3/47       Port Type Inconsistent
VLAN1                FastEthernet3/48       Port Type Inconsistent

Number of inconsistent ports (segments) in VLAN1 :2
```

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 procedure:

	Task	Command
Step 1	Select an interface to configure.	`Switch(config)# interface {{fastethernet | gigabitethernet} slot/port} | {port-channel port_channel_number}`
Step 2	Enable PortFast on a Layer 2 access port connected to a single workstation or server. You can use the **no** keyword to disable PortFast.	`Switch(config-if)# [no] spanning-tree portfast`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show running interface {{fastethernet | gigabitethernet} slot/port} | {port-channel port_channel_number}`

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

```
Switch# configure terminal
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#
```

Enabling BPDU Guard

To enable BPDU guard to shut down PortFast-configured interfaces that receive BPDUs, perform this procedure:

	Task	Command
Step 1	Enable BPDU guard on all the switch's PortFast-configured interfaces. You can use the **no** keyword to disable BPDU guard.	`Switch(config)# [no] spanning-tree portfast bpduguard`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show spanning-tree summary totals`

This example shows how to enable BPDU guard:

```
Switch# configure terminal
Switch(config)# spanning-tree portfast bpduguard
Switch(config)# end
Switch#
```

This example shows how to verify the 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#
```

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 will become the root switch. However, for spanning tree with non-default bridge priority value and non-default port cost, Uplinkfast does not increase the bridge priority nor increment the port cost. 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 procedure:

	Task	Command
Step 1	Enable UplinkFast as follows: To disable UplinkFast and restore the default rate, use the command **no spanning-tree uplinkfast max-update-rate**	`Switch(config)# [no] spanning-tree uplinkfast [max-update-rate max_update_rate]`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify that UplinkFast is enabled.	`Switch# show spanning-tree vlan vlan_ID`

This example shows how to enable UplinkFast with an update rate of 400 pps:

```
Switch# configure terminal
Switch(config)# spanning-tree uplinkfast max-update-rate 400
Switch(config)# exit
Switch#
```

This example shows how to verify that UplinkFast is 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#
```

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 procedure:

	Task	Command
Step 1	Enable BackboneFast. You can use the **no** keyword to disable BackboneFast.	`Switch(config)# [no] spanning-tree backbonefast`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify that BackboneFast is enabled.	`Switch# show spanning-tree backbonefast`

This example shows how to enable BackboneFast:

```
Switch# configure terminal
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#
```

CHAPTER 14

Configuring Cisco Express Forwarding

This chapter describes Cisco Express Forwarding (CEF) on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and examples to configure this feature.

This chapter includes the following major sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

Overview 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.

Note Policy-based routing is not supported in this release.

CEF Components

CEF stores information in several data structures rather than the route cache of multilayer switches. The data structures optimize lookup for efficient packet forwarding. Two primary components comprise the CEF operation:

- Forwarding Information Base
- Adjacency Tables

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 4006 Supervisor Engine III, 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 14-1.

Table 14-1 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.

CEF Operation Modes

CEF can be enabled in one of two modes:

- Central CEF Mode

  When Central CEF mode is enabled, the CEF FIB and adjacency tables reside on the Integrated Switching Engine hardware, and the hardware performs the express forwarding.

- Distributed CEF Mode

  The Supervisor Engine III does not support distributed CEF switching.

Catalyst 4006 Implementation of CEF

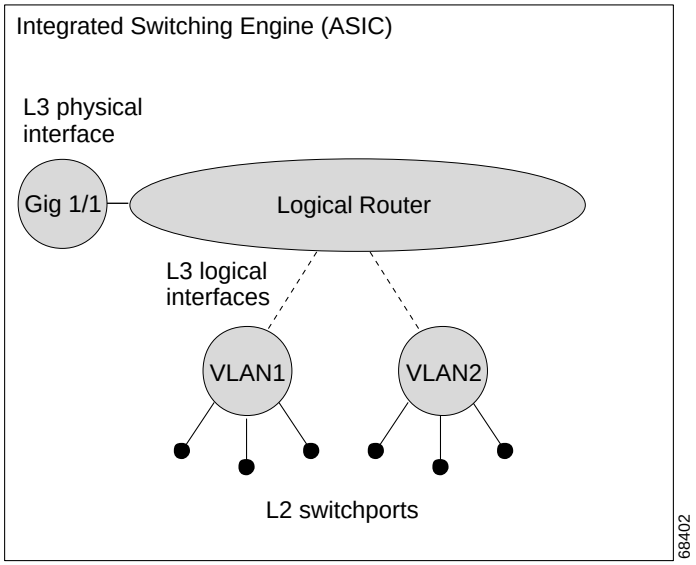
The Catalyst 4006 switch with Supervisor Engine III supports an ASIC-based Integrated Switching Engine that provides:

- 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 14-1 shows a high-level view of the ASIC-based Layer 2 and Layer 3 switching process on the Integrated Switching Engine.

Figure 14-1 Logical L2/L3 Switch Components



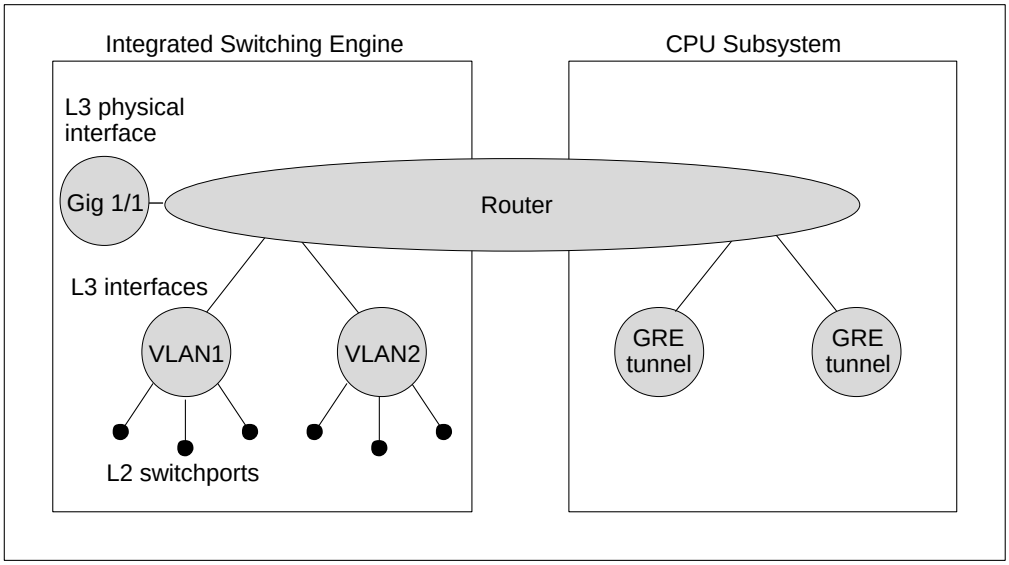
The Integrated Switching Engine performs inter-VLAN routing on logical Layer 3 interfaces with the ASIC hardware. The ASIC hardware also supports a physical Layer 3 interface that can be configured to connect with a host, a switch, or a router.

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 14-2 shows a logical view of the Integrated Switching Engine and the CPU subsystem switching components.

Figure 14-2 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 a generic routing encapsulation (GRE) tunnel.

Hardware Switching

Hardware switching is the normal operation of the Supervisor Engine III.

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 4006 with Supervisor Engine III 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 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 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 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.

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

The following sections describe how to configure CEF:

- Default Configuration, page 14-6
- Enabling CEF, page 14-7
- Configuring Load Balancing for CEF, page 14-7

Note The **ip mtu** command is not supported in this release.

Default Configuration

Table 14-2 shows the default IP unicast configuration.

Table 14-2 CEF Default Configuration Values

Feature	Default Value
CEF	Enabled globally

Enabling CEF

By default, CEF is enabled on Supervisor Engine III. No configuration is required.

To disable CEF, enter the **no IP cef** command in global configuration mode. When you disable CEF, Cisco IOS software forwards packets using the CPU subsystem software. Do not disable CEF for normal operation.

Command	Purpose
`Switch(config)# no ip cef`	Disables CEF operation.

To reenable CEF, enter the **IP cef** command in global configuration mode:

Command	Purpose
`Switch(config)# ip cef`	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. When you configure load balancing, configure it on outbound interfaces.

You can configure two types of load balancing for CEF by performing the following optional tasks:

- Configuring Per-Destination Load Balancing
- Viewing CEF Information

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.

Disabling Per-Destination Load Balancing

To disable per-destination load balancing, enter the following command in interface configuration mode:

Command	Purpose
`Switch# no ip load-sharing per-destination`	Disables per-destination load balancing.

Viewing CEF Information

You can view the collected CEF information. To do so, enter the following command in EXEC mode:

Command	Purpose
`Switch# show ip cef`	Displays the collected CEF information.

Monitoring and Maintaining CEF

To display information about IP traffic, enter the following command:

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 runs, 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, enter the following command:

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, enter the following command:

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 15

Understanding and Configuring IP Multicast

This chapter describes IP multicast routing on the Catalyst 4006 switch with Supervisor Engine III. It also provides procedures and examples to configure IP multicast routing.

Note For more information on the syntax and usage for the switch commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

This chapter includes the following major sections:

Overview of IP Multicast

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 along the path between the source and the 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 4006 switch with Supervisor Engine III, 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.

IP Multicast Protocols

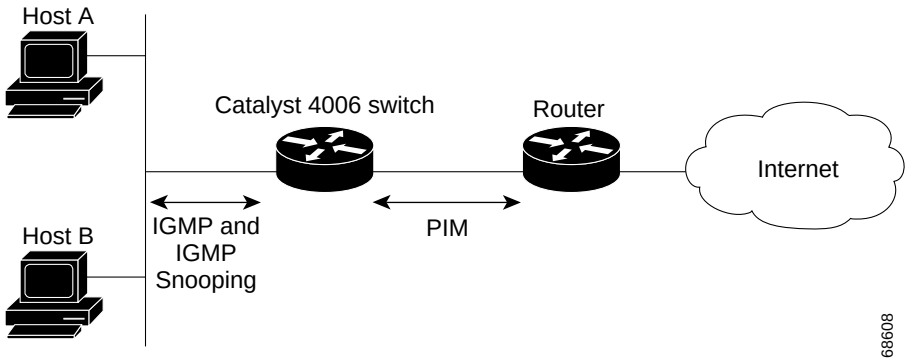
The Catalyst 4006 switch with Supervisor Engine III primarily uses the following protocols to implement IP multicast routing:

- Internet Group Management Protocol (IGMP)
- Protocol Independent Multicast (PIM)
- IGMP snooping and Cisco Group Management Protocol

Note The Catalyst 4006 switch with Supervisor Engine III supports dynamic discovery of Distance Vector Multicast Routing Protocol (DVMRP) routers and can interoperate with them using Ethernet or DVMRP tunnels.

Figure 15-1 shows where these protocols operate within the IP multicast environment.

Figure 15-1 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-DM initially floods multicast traffic throughout the network. It uses reverse path forwarding to send traffic to all Layer 3 switch or router interfaces except the one on which it arrived. Downstream routers that do not need the multicast (either because they have no receivers on their interfaces or because they are already receiving the multicast from another port) reply with a prune message, requesting to be removed from the forwarding list. This process repeats every 3 minutes.

Layer 3 switches and routers create routing state information with the flood and prune mechanism. Routing state is the source and group information that downstream routers use to build their multicast forwarding tables. PIM-DM can support only source trees—(S,G) entries. It cannot be used to build a shared distribution tree.

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.

In PIM-SM, senders and receivers first register with a single router, which is designated as an RP. Traffic is sent by the sender to the RP, which then forwards it to the registered receivers.

As intermediate routers see the source and destination of the multicast traffic (it is unlikely that the best path from source to destination goes through the RP), they optimize the paths so that the traffic takes a more direct route (likely bypassing the RP). But traffic is still sent to the RP, in case new receivers want to register.

PIM-SM uses a shared tree to distribute the information about active sources. Depending on the configuration options, the traffic can remain on the shared tree or be changed to an optimized source distribution tree. The latter is the default behavior for PIM-SM on Cisco routers. The traffic starts to flow down the shared tree, and then routers along the path determine whether there is a better path to the source. If a more direct path exists, the designated router (the router closest to the receiver) sends a join message toward the source and then reroutes the traffic along this path.

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 4006 switch with Supervisor Engine III, IGMP snooping is implemented in the forwarding ASIC, so it does not impact the forwarding rate.

Note A Catalyst 4006 switch with Supervisor Engine III can act as a CGMP server for switches that do not support IGMP snooping, such as the Catalyst 4000 family switches with Supervisor Engines I and II. You cannot configure the Catalyst 4006 switch with Supervisor Engine III as a CGMP client. To configure a Catalyst 4006 switch with Supervisor Engine III as a client, use IGMP snooping.

CGMP is a proprietary 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.

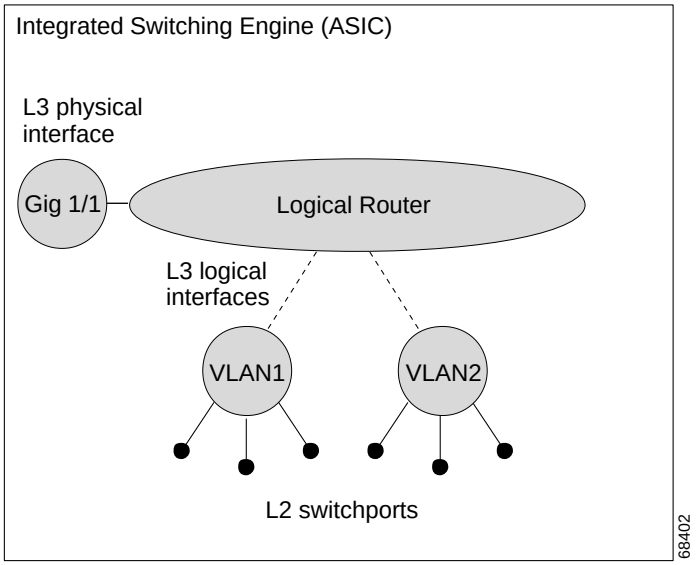
IP Multicast on the Catalyst 4006 Switch with Supervisor Engine III

The Catalyst 4006 switch with Supervisor Engine III 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 15-2 shows a logical view of Layer 2 and Layer 3 forwarding in the Integrated Switching Engine hardware.

Figure 15-2 Logical View of Layer 2 and Layer 3 Forwarding in Hardware



CEF, MFIB, and Layer 2 Forwarding

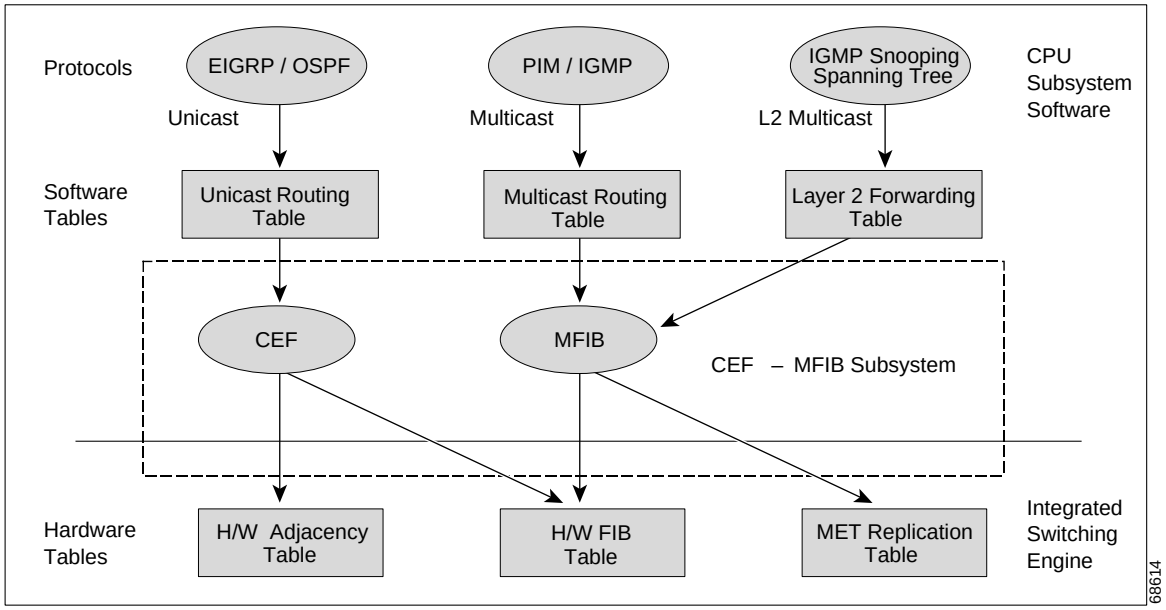
The implementation of IP multicast on the Catalyst 4006 switch with Supervisor Engine III is an extension of centralized Cisco Express Forwarding (CEF). CEF extracts information from the unicast routing table, which is created by unicast routing protocols, such as BGP, OSPF, and EIGR, and loads it into 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 4006 switch with Supervisor Engine III 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 the Layer 3 MFIB information with the Layer 2 forwarding information and stores it in the hardware MET for packet replication.

Figure 15-3 shows a functional overview of how the Catalyst 4006 switch with Supervisor Engine III combines unicast routing, multicast routing, and Layer 2 bridging information to forward in hardware.

Figure 15-3 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.

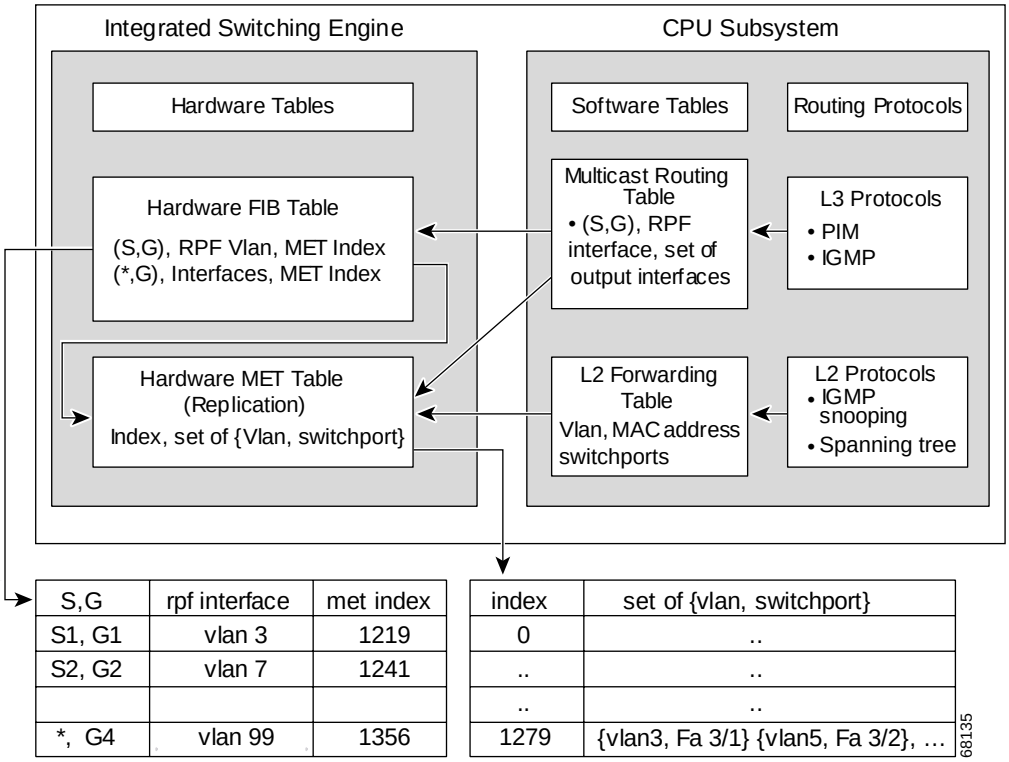
If VLAN 1 contains 1/1 and 1/2, VLAN 2 contains 2/1 and 2/2, and VLAN 3 contains 3/1 and 3/2, the MET chain for this route would contain these switchports: (1/1,1/2,2/1,2/2,3/1, and 3/2).

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 15-4 shows some key data structures that the Catalyst 4006 switch with Supervisor Engine III uses to forward IP multicast packets in hardware.

Figure 15-4 IP Multicast Tables and Protocols



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.

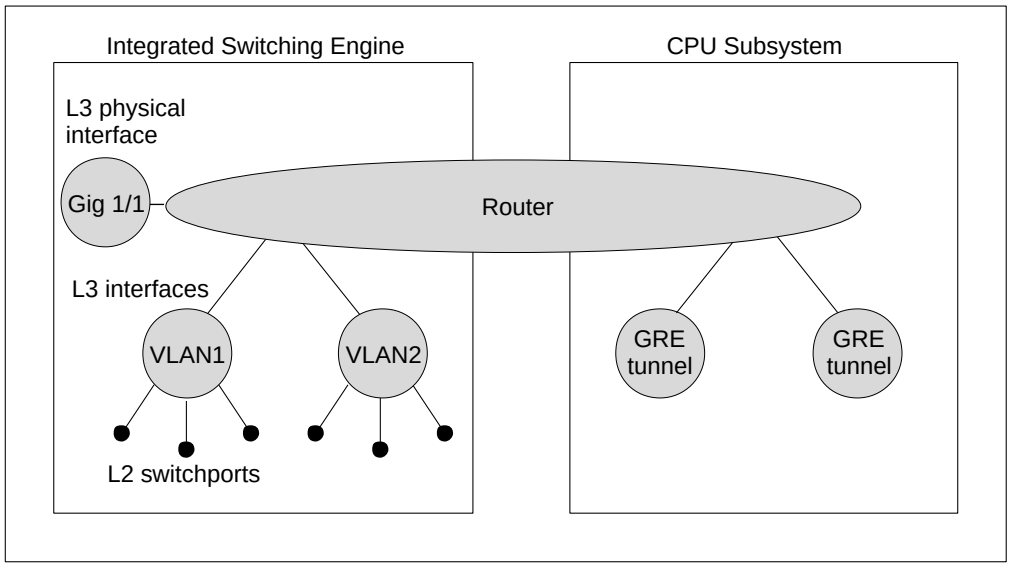
Output interface lists are stored in the multicast expansion table (MET). The MET has room for up to 32,000 output interface lists. The MET resources are shared by both Layer 3 multicast routes and by Layer 2 multicast entries. The actual number of output interface lists available in hardware depends 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. The Catalyst 4006 switch with Supervisor Engine III is not designed to forward packets in the CPU subsystem software.

Figure 15-5 shows a logical view of the hardware and software forwarding components.

Figure 15-5 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-ARPA encapsulation.

The following packets are always forwarded in software:

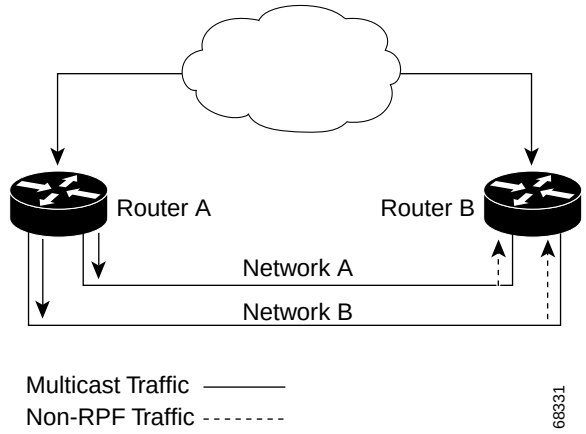
- Packets sent to multicast groups that fall into the range 224.0.0.* (where * is in the range from 0 to 255). This range is used by routing protocols. Layer 3 switching supports all other multicast group addresses.
- Packets with IP options.

Non-RPF Traffic

Traffic that fails an 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 15-6 shows how Non-RPF traffic can occur in a common network configuration.

Figure 15-6 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 4006 switch with Supervisor Engine III. 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. To display the information in the hardware tables, use the **show platform hardware** command.

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 4006 switch with Supervisor Engine III 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.

These sections describe required IP multicast routing configuration tasks:

- Enabling IP Multicast Routing, page 15-13
- Enabling PIM on an Interface, page 15-14

These sections describe optional IP multicast routing configuration tasks:

- Configuring Auto-RP, page 15-16
- Configuring PIM Version 2, page 15-18

These sections describe optional advanced IP multicast routing configuration tasks:

- Configuring Advanced PIM Features, page 15-23
- Configuring an IP Multicast Static Route, page 15-27

To see more complete information on IP multicast routing, refer to the *Cisco IOS IP and IP Routing Configuration Guide, Release 12.1*.

Default Configuration

Table 15-1 shows the IP multicast default configuration.

Table 15-1 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 4006 switch with Supervisor Engine III. 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/ios121/121newft/121t/121t5/dtssm5t.htm

Enabling IP Multicast Routing

Enabling IP multicast routing allows the Catalyst 4006 switch with Supervisor Engine III to forward multicast packets. To enable IP multicast routing on the router, enter the following command 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, enter the following command in interface configuration mode:

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, enter the following command in interface configuration mode:

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, enter the following command in interface configuration mode:

Command	Purpose
`Switch(config-if)# ip pim sparse-dense-mode`	Enables PIM to operate in sparse or dense mode, depending on the group.

Configuring a Rendezvous Point

If you configure PIM to operate in sparse mode, you must also choose one or more routers to be a rendezvous point (RP). You need not configure the routers to be RPs; they learn this themselves. RPs are used by senders to a multicast group to announce their existence and by receivers of multicast packets to learn about new senders. The Cisco IOS software can be configured so that packets for a single multicast group can use one or more RPs.

The RP address is used by first-hop routers to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop routers to send PIM join and prune messages to the RP to inform it about group membership. You must configure the RP address on all routers (including the RP router).

A PIM router can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain. The conditions specified by the access list determine for which groups the router is an RP.

To configure the address of the RP, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip pim rp-address` *`ip-address`* [*`access-list-number`*] [**`override`**]	Configures the address of a PIM RP.

Configuring Auto-RP

Auto-RP is a feature that automates the distribution of group-to-RP mappings in a PIM network. This feature has the following benefits:

- It is easy to use multiple RPs within a network to serve different group ranges.
- Auto-RP allows load splitting among different RPs and arrangement of RPs according to the location of group participants.
- Auto-RP avoids inconsistent, manual RP configurations that can cause connectivity problems.

Multiple RPs can be used to serve different group ranges or serve as hot backups of each other. To make Auto-RP work, a router must be designated as an *RP-mapping agent*, which receives the RP-announcement messages from the RPs and arbitrates conflicts. The RP-mapping agent then sends the consistent group-to-RP mappings to all other routers. Thus, all routers automatically discover which RP to use for the groups they support.

Note If you configure PIM in sparse mode or sparse-dense mode and do not configure Auto-RP, you must statically configure an RP as described in the section "Assigning an RP to Multicast Groups" later in this chapter.

Note If router interfaces are configured in sparse mode, Auto-RP can still be used if all routers are configured with a static RP address for the Auto-RP groups.

Setting Up Auto-RP in a New Internetwork

If you are setting up Auto-RP in a new internetwork, you do not need a default RP because you configure all the interfaces for sparse-dense mode. Follow the process described in the section "Adding Auto-RP to an Existing Sparse-Mode Cloud," omitting the first step of choosing a default RP.

Adding Auto-RP to an Existing Sparse-Mode Cloud

The following sections contain some suggestions for the initial deployment of Auto-RP into an existing sparse-mode cloud, to minimize disruption of the existing multicast infrastructure.

Choosing a Default RP

Sparse-mode environments need a default RP; sparse-dense-mode environments do not. If you have sparse-dense mode configured everywhere, you do not choose a default RP.

Adding Auto-RP to a sparse-mode cloud requires a default RP. In an existing PIM sparse-mode region, at least one RP is defined across the network that has good connectivity and availability. That is, the **ip pim rp-address** command is already configured on all routers in this network.

Use that RP for the global groups (for example, 224.x.x.x and other global groups). There is no need to reconfigure the group address range that RP serves. RPs discovered dynamically through Auto-RP take precedence over statically configured RPs. Typically, you would use a second RP for the local groups.

Announcing the RP and the Group Range It Serves

Find another router to serve as the RP for the local groups. The RP-mapping agent can double as an RP itself. Assign the whole range of 239.x.x.x to that RP, or assign a subrange of that (for example, 239.2.x.x).

To designate a router as the RP, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip pim send-rp-announce` *type number* `scope` *ttl* `group-list` *access-list-number*	Configures a router to be the RP.

To change the group ranges that this RP optimally serves in the future, change the announcement setting on the RP. If the change is valid, all other routers automatically adopt the new group-to-RP mapping.

The following example advertises the IP address of Ethernet 0 as the RP for the administratively scoped groups:

```
ip pim send-rp-announce ethernet0 scope 16 group-list 1
access-list 1 permit 239.0.0.0 0.255.255.255
```

Assigning the RP Mapping Agent

The RP mapping agent is the router that sends the authoritative Discovery packets notifying other routers which group-to-RP mapping to use. Such a role is necessary in the event of conflicts (such as overlapping group-to-RP ranges).

Find a router for which connectivity is not likely to be interrupted and assign it the role of RP-mapping agent. All routers within the TTL number of hops from the source router receive the Auto-RP Discovery messages. To assign the role of RP mapping agent in that router, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip pim send-rp-discovery scope` *ttl*	Assigns the RP mapping agent.

Verifying the Group-to-RP Mapping

To learn whether the group-to-RP mapping has arrived, enter one of the following commands in EXEC mode on the designated routers:

Command	Purpose
`Switch# show ip pim rp mapping`	Displays active RPs that are cached with associated multicast routing entries. Information learned by configuration or Auto-RP.
`Switch# show ip pim rp` [*group-name* | *group-address*] [`mapping`]	Displays information actually cached in the routing table.

Preventing Join Messages to False RPs

Note the **ip pim accept-rp** commands previously configured throughout the network. If the **ip pim accept-rp** command is not configured on any router, this problem can be addressed later. In those routers already configured with the **ip pim accept-rp** command, you must specify the command again to accept the newly advertised RP.

To accept all RPs advertised with Auto-RP and reject all other RPs by default, use the **ip pim accept-rp auto-rp** command.

If all interfaces are in sparse mode, a default configured RP to support the two well-known groups 224.0.1.39 and 224.0.1.40. Auto-RP relies on these two well-known groups to collect and distribute RP-mapping information. When this is the case and the **ip pim accept-rp auto-rp** command is configured, another **ip pim accept-rp** command accepting the default RP must be configured. The following is an example of this configuration:

```
ip pim accept-rp default RP address 1
access-list 1 permit 224.0.1.39
access-list 1 permit 224.0.1.40
```

Filtering Incoming RP Announcement Messages

To filter incoming RP announcement messages, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip pim rp-announce-filter rp-list` *access-list-number* `group-list` *access-list-number*	Filters incoming RP announcement messages.

Configuring PIM Version 2

PIM Version 2 includes the following improvements over PIM Version 1:

- A single, active RP exists per multicast group, with multiple backup RPs. This single RP compares to multiple active RPs for the same group in PIM Version 1.
- A bootstrap router (BSR) provides a fault-tolerant, automated RP discovery and distribution mechanism. Thus, routers dynamically learn the group-to-RP mappings.
- Sparse mode and dense mode are properties of a group, as opposed to an interface. We strongly recommend sparse-dense mode, as opposed to either sparse mode or dense mode only.
- PIM join and prune messages have more flexible encodings for multiple address families.
- A more flexible Hello packet format replaces the Query packet to encode current and future capability options.
- Register messages to an RP indicate whether they were sent by a border router or a designated router.
- PIM packets are no longer inside IGMP packets; they are standalone packets.

PIM Version 1, used with the Auto-RP feature, can perform the same tasks as the PIM Version 2 BSR. However, Auto-RP is a standalone protocol, separate from PIM Version 1, and is Cisco proprietary. PIM Version 2 is a standards track protocol in the IETF. We recommend that you use PIM Version 2.

Choose either the BSR or Auto-RP for a given range of multicast groups. If there are PIM Version 1 routers in the network, do not use the BSR.

Cisco's PIM Version 2 implementation allows interoperability and transition between Version 1 and Version 2, although there might be some minor problems. You can upgrade to PIM Version 2 incrementally. PIM Versions 1 and 2 can be configured on different routers within one network. Internally, all routers on a shared media network must run the same PIM version. Therefore, if a PIM Version 2 router detects a PIM Version 1 router, the Version 2 router downgrades itself to Version 1 until all Version 1 routers have been shut down or upgraded.

PIM uses the BSR to discover and announce RP-set information for each group prefix to all the routers in a PIM domain. This is the same function accomplished by Auto-RP, but the BSR is part of the PIM Version 2 specification. The BSR mechanism interoperates with Auto-RP on Cisco routers.

To avoid a single point of failure, you can configure several candidate BSRs in a PIM domain. A BSR is elected among the candidate BSRs automatically; they use bootstrap messages to discover which BSR has the highest priority. This router then announces to all PIM routers in the PIM domain that it is the BSR.

Routers that are configured as candidate RPs then unicast to the BSR the group range for which they are responsible. The BSR includes this information in its bootstrap messages and disseminates it to all PIM routers in the domain. Based on this information, all routers will be able to map multicast groups to specific RPs. As long as a router is receiving the bootstrap message, it has a current RP map.

Prerequisites

When PIM Version 2 routers interoperate with PIM Version 1 routers, Auto-RP should have already been deployed. A PIM Version 2 BSR that is also an Auto-RP mapping agent will automatically advertise the RP elected by Auto-RP. That is, Auto-RP prevails in its single RP being imposed on every router in the group. All routers in the domain refrain from trying to use the PIM Version 2 hash function to select multiple RPs.

Because bootstrap messages are sent hop by hop, a PIM Version1 router will prevent these messages from reaching all routers in your network. Therefore, if your network has a PIM Version 1 router in it, and only Cisco routers, it is best to use Auto-RP rather than the bootstrap mechanism. If you have a network that includes routers from other vendors, configure the Auto-RP mapping agent and the BSR on a Cisco PIM Version 2 router. Also ensure that no PIM Version 1 router is located on the path between the BSR and a non-Cisco PIM Version 2 router.

Transitioning to PIM Version 2

On each LAN, the Cisco implementation of PIM Version 2 automatically enforces the rule that all PIM messages on a shared LAN are in the same PIM version. To accommodate that rule, if a PIM Version 2 router detects a PIM Version 1 router on the same interface, the Version 2 router downgrades itself to Version 1 until all Version 1 routers have been shut down or upgraded.

Deciding When to Configure a BSR

If there are only Cisco routers in your network (no routers from other vendors), there is no need to configure a BSR. Configure Auto-RP in the mixed PIM Version 1/Version 2 environment.

However, if you have non-Cisco, PIM Version 2 routers that need to interoperate with Cisco routers running PIM Version 1, both Auto-RP and a BSR are required. We recommend that a Cisco PIM Version 2 router be both the Auto-RP mapping agent and the BSR.

Dense Mode

Dense mode groups in a mixed Version 1/Version 2 region need no special configuration; they will interoperate automatically.

Sparse Mode

Sparse mode groups in a mixed Version 1/Version 2 region are possible because the Auto-RP feature in Version 1 interoperates with the RP feature of Version 2. Although all PIM Version 2 routers are also capable of using Version 1, we recommend that the RPs be upgraded to Version 2 (or at least upgraded to PIM Version 1 in the Cisco IOS Release 11.3 software).

To ease the transition to PIM Version 2, we also recommend the following:

- Use Auto-RP throughout the region
- Configure sparse-dense mode throughout the region

If Auto-RP was not already configured in the PIM Version 1 regions, configure Auto-RP.

PIM Version 2 Configuration Tasks

There are two approaches to using PIM Version 2. You can use Version 2 exclusively in your network, or migrate to Version 2 by employing a mixed PIM version environment.

- If your network is all Cisco routers, you may use either Auto-RP or the bootstrap mechanism (BSR).
- If you have non-Cisco routers in your network, you need to use the bootstrap mechanism.
- If you have PIM Version 1 and PIM Version 2 Cisco routers and routers from other vendors, then you must use both Auto-RP and the bootstrap mechanism.

The tasks to configure PIM Version 2 are described in the following sections:

- Specifying the PIM Version, page 15-20
- Configuring PIM Version 2 Only, page 15-20
- Transitioning to PIM Version 2, page 15-19
- Monitoring the RP Mapping Information, page 15-23
- Troubleshooting, page 15-23

Specifying the PIM Version

All systems using Cisco IOS Release 11.3(2)T or later start in PIM Version 2 mode by default. In case you need to reenable PIM Version 2 or specify PIM Version 1 for some reason, you can control the PIM version by entering the following command in interface configuration mode:

Command	Purpose
`Switch(interface)# ip pim version [1 | 2]`	Configures the PIM version used.

Configuring PIM Version 2 Only

To configure PIM Version 2 exclusively, perform the tasks in this section. It is assumed that no PIM Version 1 system exists in the PIM domain.

The first task is recommended, configuring sparse-dense mode. If you configure Auto-RP, none of the other tasks is required to run PIM Version 2. To configure Auto-RP, see the section "Configuring Auto-RP" earlier in this chapter.

If you want to configure a BSR, complete the tasks in the following sections:

- Configuring PIM Sparse-Dense Mode, page 15-21
- Defining the PIM Domain Border, page 15-21
- Configuring Candidate BSRs, page 15-21
- Configuring Candidate RPs, page 15-22

Configuring PIM Sparse-Dense Mode

To configure PIM sparse-dense mode, enter the following commands on all PIM routers inside the PIM domain, beginning in global configuration mode:

	Command	Purpose
Step 1	`Switch(config)# ip multicast-routing`	Enables IP multicast routing.
Step 2	`Switch(interface)# interface type number`	Configures an interface.
Step 3	`Switch(config)# ip pim sparse-dense-mode`	Enables PIM on the interface. The sparse-dense mode is identical to the implicit interface mode in the PIM Version 2 specification.

Repeat Steps 2 and 3 above for each interface on which you want to run PIM.

Defining the PIM Domain Border

Configure a border for the PIM domain, so that bootstrap messages do not cross this border in either direction. Therefore, different BSRs will be elected on the two sides of the PIM border. Use the following command on the interface of a border router peering with one or more neighbors outside the PIM domain. To configure a PIM domain boundary, enter the following command in interface configuration mode:

Command	Purpose
`Switch(config)# ip pim border`	Configures a PIM domain boundary.

Configuring Candidate BSRs

You should configure one or more candidate BSRs. The routers that serve as candidate BSRs should be well connected and be in the backbone portion of the network, as opposed to the dialup portion of the network. On the candidate BSRs, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip pim bsr-candidate hash-mask-length [priority]`	Configure the router to be a candidate bootstrap router.

Configuring Candidate RPs

Configure one or more candidate RPs. Similar to BSRs, the RPs should also be well connected and in the backbone portion of the network. An RP can serve the entire IP multicast address space or a portion of it. Candidate RPs send candidate RP advertisements to the BSR. Consider the following when deciding which routers should be RPs:

- In a network of Cisco routers where only Auto-RP is used, any router can be configured as an RP.
- In a network of routers that includes only Cisco PIM Version 2 routers and routers from other vendors, any router can be used as an RP.
- In a network of Cisco PIM Version 1 routers, Cisco PIM Version 2 routers, and routers from other vendors, only Cisco PIM Version 2 routers should be configured as RPs.

On the candidate RPs, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip pim rp-candidate type number ttl group-list access-list-number`	Configures the router to be a candidate RP.

For examples of configuring PIM Version 2, see the section "BSR Configuration Example" at the end of this chapter.

Using Auto-RP and a BSR

If you must have one or more BSRs, as described earlier in the prior section "Deciding When to Configure a BSR," we recommend the following:

- Configure the candidate BSRs as the RP-mapping agents for Auto-RP.
- For group prefixes advertised via Auto-RP, the Version 2 BSR mechanism should not advertise a subrange of these group prefixes served by a different set of RPs. In a mixed Version 1/Version 2 PIM domain, it is preferable to have backup RPs serve the same group prefixes. This prevents the Version 2 DRs from selecting a different RP from those Version 1 DRs, due to longest match lookup in the RP-mapping database.

To verify the consistency of group-to-RP mappings, perform the following tasks in EXEC mode:

	Task	Purpose
Step 1	`Switch# show ip pim rp [[group-name | group-address] | mapping]`	Displays the available RP mappings on any router.
Step 2	`Switch# show ip pim rp-hash group`	Confirms that the same RP appears that a PIM Version 1 system chooses on a PIM Version 2 router.

Monitoring the RP Mapping Information

To monitor the RP mapping information, you can enter the following commands in EXEC mode:

Command	Purpose
`Switch# show ip pim bsr`	Displays information about the currently elected BSR.
`Switch# show ip pim rp-hash group`	Displays the RP that was selected for the specified group.
`Switch# show ip pim rp [group-name | group-address | mapping]`	Displays how the router learns of the RP (via bootstrap or Auto-RP mechanism).

Troubleshooting

When debugging interoperability problems between PIM Version 1 and Version 2, perform the following tasks:

Step 1 Verify RP mapping with the **show ip pim rp-hash** command, making sure that all systems agree on the same RP for the same group.

Step 2 Verify interoperability between different versions of DRs and RPs. Ensure that the RPs are interacting with the DRs properly (by responding with register-stops and forwarding de-encapsulated data packets from registers).

Configuring Advanced PIM Features

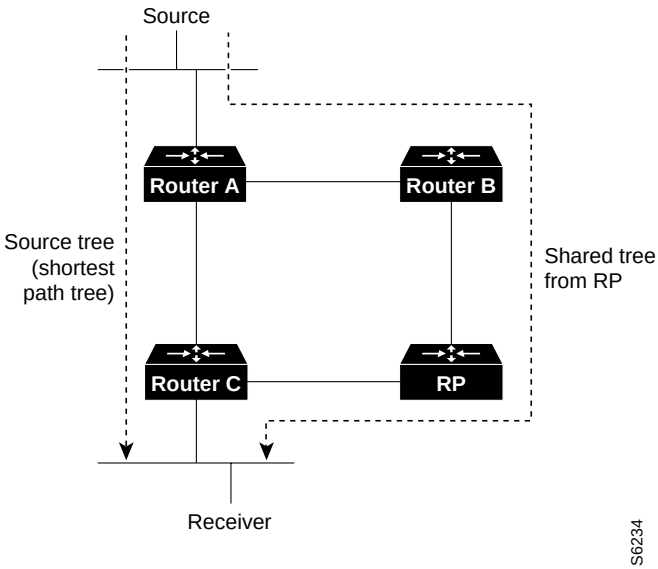
Perform the optional tasks in the following sections to configure PIM features:

- Understanding PIM Shared Tree and Source Tree (Shortest Path Tree), page 15-23
- Delaying the Use of PIM Shortest Path Tree, page 15-25
- Understanding Reverse Path Forwarding, page 15-25
- Assigning an RP to Multicast Groups, page 15-26
- Increasing Control over RPs, page 15-26
- Modifying the PIM Router-Query Message Interval, page 15-26

Understanding PIM Shared Tree and Source Tree (Shortest Path Tree)

By default, members of a group receive data from senders to the group across a single data distribution tree rooted at the RP. This type of distribution tree is called *shared tree*, and is shown in Figure 15-7. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 15-7 Shared Tree and Source Tree (Shortest Path Tree)



If the data rate warrants, leaf routers on the shared tree can initiate a switch to the data distribution tree rooted at the source. This type of distribution tree is called a *shortest path tree* or *source tree*. By default, the Cisco IOS software changes to a source tree configuration upon receiving the first data packet from a source.

The following process describes the move from shared tree to source tree:

1. Receiver joins a group; leaf Router C sends a join message toward the RP.
2. RP puts link to Router C in its outgoing interface list.
3. Source sends data; Router A encapsulates data in Register and sends it to the RP.
4. RP forwards data down the shared tree to Router C and sends a join message toward Source. At this point, data may arrive twice at Router C, once encapsulated and once natively.
5. When data arrives natively (unencapsulated) at RP, RP sends a Register-Stop message to Router A.
6. By default, reception of the first data packet prompts Router C to send a join message toward Source.
7. When Router C receives data on (S,G), it sends a prune message for Source up the shared tree.
8. RP deletes the link to Router C from outgoing interface of (S,G). RP triggers a prune message toward Source.

The join and prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM router along the path to the source or RP. Register and Register-Stop messages are not sent hop-by-hop. They are sent by the designated router that is directly connected to a source and are received by the RP for the group.

Multiple sources sending to groups use the shared tree.

The network manager can configure the router to stay on the shared tree, as described in the following section, “Delaying the Use of PIM Shortest Path Tree.”

Delaying the Use of PIM Shortest Path Tree

The change from shared to source tree happens upon the arrival of the first data packet at the last hop router (Router C in Figure 15-7). This change occurs because the **ip pim spt-threshold** command controls that timing, and its default setting is 0 Kbps.

The shortest path tree requires more memory than the shared tree, but reduces delay. You might want to postpone its use. Instead of allowing the leaf router to move to the shortest path tree immediately, you can specify that the traffic must first reach a threshold.

You can configure when a PIM leaf router should join the shortest path tree for a specified group. If a source sends at a rate greater than or equal to the specified kbps rate, the router triggers a PIM join message toward the source to construct a source tree (shortest path tree). If **infinity** is specified, all sources for the specified group use the shared tree, never switching to the source tree.

The group list is a standard access list that controls what groups the shortest path tree threshold applies to. If a value of 0 is specified or the group list is not used, the threshold applies to all groups.

To configure a traffic rate threshold that must be reached before multicast routing is switched from the source tree to the shortest path tree, enter the following command in interface configuration mode:

Command	Purpose
`Switch(config)# ip pim spt-threshold {kbps | infinity} [group-list access-list-number]`	Specifies the threshold that must be reached before moving to shortest path tree.

Understanding Reverse Path Forwarding

Reverse Path Forwarding (RPF) is an algorithm used for forwarding multicast datagrams. It functions as follows:

- If a router receives a datagram on an interface it uses to send unicast packets to the source, the packet has arrived on the RPF interface.
- If the packet arrives on the RPF interface, a router forwards the packet out the interfaces present in the outgoing interface list of a multicast routing table entry.
- If the packet does not arrive on the RPF interface, the packet is silently discarded to prevent loops.

PIM uses both source trees and RP-rooted shared trees to forward datagrams; the RPF check is performed differently for each, as follows:

- If a PIM router has source-tree state (that is, an (S,G) entry is present in the multicast routing table), the router performs the RPF check against the IP address of the source of the multicast packet.
- If a PIM router has shared-tree state (and no explicit source-tree state), it performs the RPF check on the RP's address (which is known when members join the group).

Sparse-mode PIM uses the RPF lookup function to determine where it needs to send joins and prunes. (S,G) joins (which are source-tree states) are sent toward the source. The (*,G) joins (which are shared-tree states) are sent toward the RP.

DVMRP and dense-mode PIM use only source trees and use RPF as previously described.

Assigning an RP to Multicast Groups

If you have configured PIM sparse mode, you must configure a PIM RP for a multicast group. An RP can either be configured statically in each box or learned through a dynamic mechanism. This section explains how to statically configure an RP. If the RP for a group is learned through a dynamic mechanism (such as Auto-RP), you need not perform this task for that RP. You should use Auto-RP, which is described in the section "Configuring Auto-RP" earlier in this chapter.

PIM-designated routers forward data from directly connected multicast sources to the RP for distribution down the shared tree.

Data is forwarded to the RP in one of two ways. It is encapsulated in Register packets and unicast directly to the RP, or, if the RP has itself joined the source tree, it is multicast forwarded per the RPF forwarding algorithm described in the preceding section, "Understanding Reverse Path Forwarding." Last-hop routers directly connected to receivers can join themselves to the source tree and prune themselves from the shared tree.

A single RP can be configured for multiple groups defined by an access list. If there is no RP configured for a group, the router treats the group as dense using the dense-mode PIM techniques.

If a conflict exists between the RP configured with this command and one learned by Auto-RP, the Auto-RP information is used, unless the **override** keyword is configured.

To assign an RP to one or more multicast groups, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip pim rp-address` *ip-address* [*group-access-list-number*] [**override**]	Assigns an RP to multicast groups.

Increasing Control over RPs

You can take measures to prevent a misconfigured leaf router from interrupting PIM service to the remainder of a network. To do so, configure the local router to accept join messages only if they contain the RP address specified by the access list. To configure this feature, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip pim accept-rp` {*address* | **auto-rp**} [*access-list-number*]	Controls which RPs the local router will accept join messages to.

Modifying the PIM Router-Query Message Interval

Router-query messages are used to elect a PIM-designated router. The designated router is responsible for sending IGMP host-query messages. By default, multicast routers send PIM router-query messages every 30 seconds. To modify this interval, enter the following command in interface configuration mode:

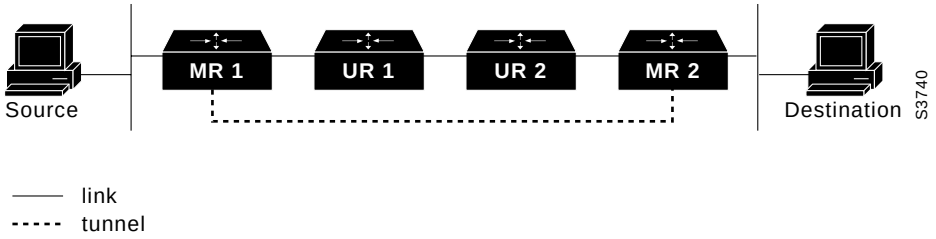
Command	Purpose
`Switch(interface)# ip pim query-interval` *seconds*	Configures the frequency at which multicast routers send PIM router-query messages.

Configuring an IP Multicast Static Route

IP multicast static routes (mroutes) allow multicast paths to diverge from the unicast paths. When using PIM, the router expects to receive packets on the same interface where it sends unicast packets back to the source. This expectation is beneficial if your multicast and unicast topologies are congruent. However, you might want unicast packets to take one path and multicast packets to take another.

The most common reason for using separate unicast and multicast paths is tunneling. When a path between a source and a destination does not support multicast routing, a solution is to configure two routers with a GRE tunnel between them. In Figure 15-8, each unicast router (UR) supports unicast packets only; each multicast router (MR) supports multicast packets.

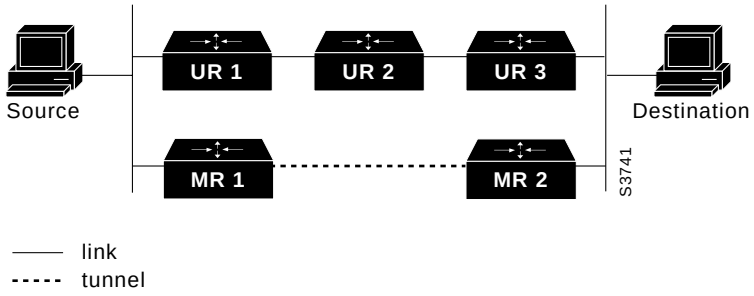
Figure 15-8 Tunnel for Multicast Packets



As shown in Figure 15-8, Source delivers multicast packets to Destination by using MR 1 and MR 2. MR 2 accepts the multicast packet only if it thinks it can reach Source over the tunnel. If so, when Destination sends unicast packets to Source, MR 2 sends them over the tunnel. This could be slower than natively sending the unicast packet through UR 2, UR 1, and MR 1.

Prior to the implementation of multicast static routes, the configuration shown in Figure 15-9 was used to overcome the problem of both unicasts and multicasts using the tunnel. In this figure, MR 1 and MR 2 are used as multicast routers only. When Destination sends unicast packets to Source, it uses the (UR 3, UR 2, UR 1) path. When Destination sends multicast packets, the UR routers do not understand or forward them. However, the MR routers forward the packets.

Figure 15-9 Separate Paths for Unicast and Multicast Packets



To make the configuration shown in Figure 15-9 work, MR 1 and MR 2 must run another routing protocol (typically, a different instantiation of the same protocol running in the UR routers), so that paths from sources are learned dynamically.

A multicast static route allows you to use the configuration shown in Figure 15-8 by configuring a static multicast source. The Cisco IOS software uses the configuration information instead of the unicast routing table. Therefore, multicast packets can use the tunnel without having unicast packets use the tunnel. Static mroutes are local to the router they are configured on and not advertised or redistributed in any way to any other router.

To configure a multicast static route, enter the following command in global configuration mode:

Command	Purpose
`Switch(config)# ip mroute source mask [protocol as-number] {rpf-address | type number} [distance]`	Configures an IP multicast static route.

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 15-28
- Displaying the Multicast Routing Table, page 15-29
- Displaying IP MFIB, page 15-31
- Displaying IP MFIB Fast Drop, page 15-32
- Displaying PIM Statistics, page 15-32
- Clearing Tables and Databases, page 15-33

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 enter any of the following commands in EXEC mode:

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 Output 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, you can enter the following commands in EXEC mode:

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

To display fast-drop entries, enter the following command in EXEC mode:

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, enter the following commands in EXEC mode:

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 15-34
- PIM Sparse Mode Example, page 15-34
- BSR Configuration Example, page 15-34

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 16

Configuring Network Security

This chapter contains network security information that is unique to the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- Hardware and Software ACL Support, page 16-1
- Guidelines and Restrictions for Using Layer 4 Operators in ACLs, page 16-2

For network security information and procedures, refer to these publications:

- *Cisco IOS Security Configuration Guide*, Release 12.1, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/index.htm
- *Cisco IOS Security Command Reference*, Release 12.1, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/index.htm

By default, the Catalyst 4006 switch with Supervisor Engine III 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, regardless of whether the **ip unreachables** command is enabled.

All packets denied by an output access list are always forwarded to the CPU.

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Hardware and Software ACL Support

This section describes how to determine whether access control lists (ACLs) are processed in hardware or in software:

- Flows that match a deny statement in standard and extended ACLs (input and output) are dropped in hardware if ICMP unreachable messages are disabled.
- Flows that match a permit statement in standard and extended ACLs (input and output) are processed in hardware.

- The following ACL types are not supported in software:
  - Standard XNS access list
  - Extended XNS access list
  - DECnet access list
  - Extended MAC address access list
  - Protocol type-code access list
  - Standard 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 nonlogged packet processing is not impacted.

Guidelines and Restrictions for Using Layer 4 Operators in ACLs

The following sections describe guidelines and restrictions for configuring ACLs that include Layer 4 port operations:

- How to Apply Layer 4 Operations, page 16-2
- How ACL Processing Impacts CPU, page 16-3

How to Apply 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)

We recommend that you not specify more than six different Layer 4 operations on the same ACL. If you exceed this number, each new operation might cause the affected ACE (access list entry) to be processed in software.

Use the following guidelines when applying Layer 4 operators:

1. Layer 4 operations are considered different if the operator or operand differ. For example, in the following ACL, three different Layer 4 operations exist (gt 10 and gt 11 are considered two different Layer 4 operations):

```
... gt 10 permit
... lt 9 deny
... gt 11 deny
```

Note The eq operators can be used an unlimited number of times as this operator does not use a Layer 4 operation in hardware.

2. 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)
  - neg6 permit is shared between the two ACLs because they are identical and both operate on the same destination port.
- An explanation 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:

1. For some packets, access control list matches must be performed by the software when the hardware runs out of resources.
   - TCP flag combinations other than rst ack and syn fin rst are processed in software. rst ack is equivalent to the keyword **established**.

– You can have up to six Layer 4 operations (lt, gt, neq, and range) in an ACL, in order for all operations to be processed in hardware. The eq operator does not require any Layer 4 operations and can be used any number of times. In addition, Layer 4 operations can be shared by source and destination operands as even-pairings only, if the total number of Layer 4 operations is six. You can set zero source and six destination operations or two source and four destination operations, but you cannot set three source and three destination operations if you want all six Layer 4 operations performed in hardware. If you use three source and three destination operations, the third access control entry will be handled in software.

– If the total number of Layer 4 operations in an ACL is less than six, the operations can be distributed in any way you choose.

– If the total number of Layer 4 operations in an ACL is greater than six, the additional Layer 4 operations are processed in software

Examples:

The following access lists will be 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
```

Note Access lists 104 and 105 are identical; established is shorthand for rst and ack.

Access list 101, below, will be processed completely in software:

```
access-list 101 permit tcp any any urg
```

Because four source and two destination operations exist, access list 106, below, will be 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 first two access lists in access list 102 will be processed in hardware. The third access list will be processed in software, because three source and three destination operations exist.

```
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 will be 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 Source port lt 80 and destination port lt 80 are considered different operations.

2. 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 when:
   - A log keyword is used
   - An output ACL denies a packet
   - 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)

CHAPTER 17

Understanding and Configuring IGMP Snooping

This chapter describes how to configure Internet Group Management Protocol (IGMP) snooping on the Catalyst 4006 switch with Supervisor Engine III. It provides guidelines, procedures, and configuration examples.

This chapter consists of the following major sections:

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*, Release 12.1at this URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt3/1cdmulti.htm

Note For complete syntax and usage information for commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the additional publications at this URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Overview of IGMP Snooping

Note Quality of service does not apply to IGMP packets when IGMP snooping is enabled.

IGMP snooping allows a switch to *snoop* or capture information from IGMP packets being sent back and forth between hosts and a router. Based on this information, a switch will add/delete multicast addresses from its address table, thereby enabling/disabling multicast traffic from flowing to the individual host ports.

In subnets where you have configured IGMP, IGMP snooping manages multicast traffic at Layer 2. You can configure interfaces using the **switchport** keyword to dynamically forward multicast traffic only to those interfaces that want to receive it.

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 and RFC 2236.

IGMP (on a router) periodically sends out general IGMP queries. When you enable IGMP snooping, the switch responds at Layer 2 to the IGMP queries with only one IGMP join request per Layer 2 multicast group. The switch creates one entry per subnet 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 join requests 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 for a multicast group address statically, your static setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can consist of both user-defined settings and IGMP snooping.

Groups with IP addresses in the 224.0.0—255 range are reserved for routing control packets and are flooded to all forwarding ports of the VLAN. These addresses map to the multicast MAC address range 0100.5E00.0001 to 01-00-5E-00-00-0xFF.

Note If a spanning-tree topology change occurs in a VLAN, IP multicast traffic floods in all ports in the VLAN where PortFast is not enabled for two general query intervals.

In order for a host connected to a Layer 2 interface to join an IP multicast group, the host sends an IGMP join request specifying the IP multicast group. For a host to leave a multicast group, it can either ignore the periodic general IGMP queries or it can send an IGMP leave message. When the switch receives an IGMP leave message from a host, it sends out a mac-based-general IGMP 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.

Fast-Leave Processing

IGMP snooping fast-leave processing allows the switch to remove an interface from the forwarding-table entry without first sending out mac-based-general queries to the interface. The VLAN interface is pruned from the multicast tree for the multicast group specified in the original leave message. Fast-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 IP mac-based-general IGMPv2 leave message, it sends a mac-based-general query out the interface where the leave message was received to determine when there are any other hosts attached to that interface that are interested in the MAC multicast group. The interface is removed from the port list of the (mac-group, VLAN) entry in the Layer 2 forwarding table, if the following conditions apply:

- If the switch does not receive an IGMP join message within the query-response-interval
- If none of the other 31 IP groups corresponding to the MAC group is interested in the multicast traffic for that MAC group
- If no multicast routers have been learned on the interface

With fast-leave 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 Use fast-leave processing only on VLANs where only one host is connected to each interface. If fast-leave is enabled in VLANs where more than one host is connected to an interface, some hosts might be dropped inadvertently. Immediate-leave processing is supported only with IGMP version 2 hosts.

Configuring IGMP Snooping

Note To use IGMP snooping, configure a Layer 3 interface in the subnet for multicast routing (see Chapter 15, "Understanding and Configuring IP Multicast").

Note When you are configuring IGMP, configure the VLAN in the VLAN database (see Chapter 7, "Understanding and Configuring VLANs").

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

Table 17-1 shows the IGMP snooping default configuration values.

Table 17-1 IGMP Snooping Default Configuration Values

Feature	Default Value
IGMP snooping	Enabled
Multicast routers	None configured
IGMP snooping learning method	PIM/DVMRP[1]

1. PIM/DVMRP = Protocol Independent Multicast/Distance Vector Multicast Routing Protocol

Enabling IGMP Snooping

To enable IGMP snooping globally, follow these steps:

	Task	Command
Step 1	Enable IGMP snooping. Use the **no** keyword to disable IGMP snooping.	`Switch(config)# [no] ip igmp snooping`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show ip igmp snooping | include globally`

This example shows how to enable IGMP snooping globally and verify the configuration:

```
Switch(config)# ip igmp snooping
Switch(config)# end
Switch# show ip igmp snooping | include globally
  IGMP snooping is globally enabled
Switch#
```

To enable IGMP snooping in a VLAN, follow these steps:

	Task	Command
Step 1	Enable IGMP snooping. Use the **no** keyword to disable IGMP snooping.	`Switch(config)# [no] ip igmp snooping vlan` *vlan_ID*
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show ip igmp snooping vlan` *vlan_ID*

This example shows how to enable IGMP snooping on VLAN 200 and verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 200
Switch(config)# end
Switch# show ip igmp snooping vlan 200
vlan 200
IGMP snooping is globally enabled
IGMP snooping is enabled on this VLAN
IGMP snooping intermediate-leave is disabled on this VLAN
IGMP snooping mrouter learn mode is pim-dvmrp on this VLAN
IGMP snooping is running in IGMP_ONLY mode on the VLAN
Switch#
```

Configuring Learning Methods

The following sections describe IGMP snooping learning methods:

- Configuring PIM/DVMRP Learning, page 17-5
- Configuring CGMP Learning, page 17-5

Configuring PIM/DVMRP Learning

To configure IGMP snooping to learn from PIM/DVMRP packets, use the following command:

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(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, use the following command:

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 use CGMP self-join packets as the learning method:

```
Switch(config)# ip igmp snooping vlan 1 mrouter learn cgmp
Switch(config)# end
Switch#
```

Configuring a Multicast Router Port Statically

To configure a static connection to a multicast router, enter the **ip igmp snooping mrouter** command on the switch instead of the **ip cgmp router-only** command.

To configure a static connection to a multicast router, follow these steps:

	Task	Command
Step 1	Specify 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 must be administratively up, and the line protocol must be up.	`Switch(config)# ip igmp snooping vlan vlan_ID mrouter interface interface_num`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show ip igmp snooping mrouter vlan vlan_ID`

This example shows how to configure a static connection to a multicast router:

```
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 Fast-Leave Processing

When you enable IGMP fast-leave processing in a VLAN, the switch will remove an interface from the multicast group as soon as it detects an IGMP version 2 leave message on that interface.

To enable IGMP fast-leave processing on an interface, use the following command:

Command	Purpose
`Switch(config)# ip igmp snooping vlan` *vlan_ID* `immediate-leave`	Enables IGMP fast-leave processing in the VLAN.

This example shows how to enable IGMP fast-leave processing on interface VLAN 200 and verify the configuration:

```
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
IGMP snooping fast-leave is enabled on this vlan
Switch(config)#
```

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, use the following command:

Command	Purpose
`Switch(config-if)# ip igmp snooping vlan` *vlan_ID* `static` *mac_address* `interface` *interface_num*	Configures a host statically in the VLAN.

This example shows how to configure a host statically in VLAN 200 on interface FastEthernet 2/11:

```
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)#
```

Suppressing Multicast Flooding

An IGMP snooping-enabled switch will flood 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, a new interface command and two new global commands are introduced in release 12.1(11b)EW.

The new interface command is as follows:

[no | default] ip igmp snooping tcn flood

These are the new global commands:

[no | default] ip igmp snooping tcn flood query count [1 - 10]

[no | default] ip igmp snooping tcn query solicit

Prior to 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 release 12.1(11b)EW, the default time period the switch will wait before multicast flooding will stop 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.

Note To immediately issue an IGMP query in the current Cisco router IGMP implementation, the router must be CGMP server-enabled. Otherwise, the router will issue an IGMP general query at the regular interval.

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. The Catalyst 4000 switch takes special actions when the topology changes in order 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 4000 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 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 will apply 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, use the following procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 2	Disable 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**	`Switch(config-if)# no ip igmp snooping tcn flood`
Step 3	Exit configuration mode.	`Switch(config)# end`
Step 4	Verify the configuration.	`Switch# show running interface {fastethernet | gigabitethernet} slot/port`

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, use the following procedure:

	Task	Command
Step 1	Modify the number of IGMP queries the switch will wait 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** .	`Switch(config)#` **`ip igmp snooping tcn flood query count <n>`**
Step 2	Exit configuration mode.	`Switch(config)#` **`end`**

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, use the following procedure:

	Task	Command
Step 1	Configure 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**	`Switch(config)#` **`ip igmp snooping tcn query solicit`**
Step 2	Exit configuration mode.	`Switch(config)#` **`end`**

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 tell you how to display IGMP snooping information:

- Displaying Multicast Router Interfaces, page 17-10
- Displaying MAC Address Multicast Entries, page 17-10
- Displaying IGMP Snooping Information for a VLAN Interface, page 17-11

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, use the following command:

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, use the following command:

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 a VLAN:

```
Switch# show mac-address-table multicast vlan 1 count
Multicast MAC Entries for vlan 1:    4
Switch#
```

Displaying IGMP Snooping Information for a VLAN Interface

To display IGMP snooping information for a VLAN interface, use the following command:

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 interface VLAN 200:

```
Switch#show ip igmp snooping vlan 1
vlan 1
----------
 IGMP snooping is globally enabled
 IGMP snooping TCN solicit query is globally disabled
 IGMP snooping global TCN flood query count is 2
 IGMP snooping is enabled on this Vlan
 IGMP snooping immediate-leave is disabled on this Vlan
 IGMP snooping mrouter learn mode is pim-dvmrp on this Vlan
 IGMP snooping is running in IGMP_ONLY mode on this Vlan
Switch#
```

Configuring IGMP Filtering

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 17-2 shows the default IGMP filtering configuration.

Table 17-2 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.

Beginning in privileged EXEC mode, follow these steps to create an IGMP profile:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip igmp profile** *profile number*	Enter IGMP profile configuration mode, and assign a number to the profile you are configuring. The range is from 1 to 4,294,967,295.
Step 3	**permit** | **deny**	(Optional) Set 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	**range** *ip multicast address*	Enter 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	**end**	Return to privileged EXEC mode.
Step 6	**show ip igmp profile** *profile number*	Verify the profile configuration.
Step 7	**copy running-config startup-config**	(Optional) Save 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 # config t
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.

Beginning in privileged EXEC mode, follow these steps to apply an IGMP profile to a switch port:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter 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	**ip igmp filter** *profile number*	Apply the specified IGMP profile to the interface. The profile number can be from 1 to 4,294,967,295.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running configuration interface** *interface-id*	Verify the configuration.
Step 6	**copy running-config startup-config**	(Optional) Save 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 verify the configuration.

```
Switch # config t
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.

Beginning in privileged EXEC mode, follow these steps to apply an IGMP profile to a switch port:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Enter 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	**ip igmp max-groups** *number*	Set 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.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-configuration interface** *interface-id*	Verify the configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the maximum group limitation and return to the default of no maximum, use the **no ip igmp max-groups** command.

This example shows how to limit the number of IGMP groups that an interface can join to 25.

```
Switch# config t
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.

Use the privileged EXEC commands in Table 17-3 to display IGMP filtering configuration:

Table 17-3 Commands to Display IGMP Filtering Configuration

Command	Purpose
show ip igmp profile [*profile number*]	Displays the specified IGMP profile or all IGMP profiles defined on the 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 18

Understanding and Configuring CDP

This chapter describes how to configure Cisco Discovery Protocol (CDP) on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.1; *Cisco IOS System Management*; *Configuring Cisco Discovery Protocol (CDP)* at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_c/fcprt3/fcd301c.htm and to the *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.1; *Cisco IOS System Management Commands*; and *CDP Commands* publication at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/frprt3/frd3001b.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

To enable CDP globally, enter the following command:

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, enter the following command:

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 18-3.

Enabling CDP on an Interface

To enable CDP on an interface, enter the following command:

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, enter the following command:

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:

Task	Command
Reset the traffic counters to zero.	`Switch# clear cdp counters`
Delete the CDP table of information about neighbors.	`Switch# clear cdp table`
Display global information such as frequency of transmissions and the holdtime for packets being transmitted.	`Switch# show cdp`
Display information about a specific neighbor. The display can be limited to protocol or version information.	`Switch# show cdp entry entry_name [protocol | version]`
Display information about interfaces on which CDP is enabled.	`Switch# show cdp interface [type/number]`
Display 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 neighbors [type/number] [detail]`
Display CDP counters, including the number of packets sent and received and checksum errors.	`Switch# show cdp traffic`
Display information about the types of debugging that are enabled for your switch.	`Switch# show debugging`

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 19

Understanding and Configuring UDLD

This chapter describes how to configure the UniDirectional Link Detection (UDLD) on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/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.

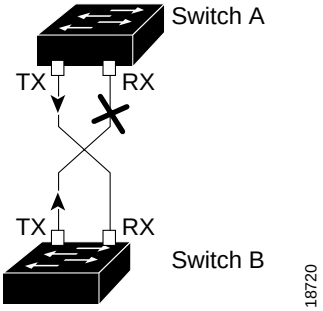
If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally from a Layer 1 perspective, then UDLD at Layer 2 determines whether those fibers are connected correctly and whether 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 19-1 shows an example of a unidirectional link condition. Switch B successfully receives traffic from Switch A on the interface. However, Switch A does not receive traffic from Switch B on the same interface. UDLD detects the problem and disables the interface.

Figure 19-1 Unidirectional Link



Default UDLD Configuration

Table 19-1 shows the UDLD default configuration.

Table 19-1 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

The following sections describe how to configure UDLD:

- Enabling UDLD Globally, page 19-3
- Enabling UDLD on Individual Interfaces, page 19-3
- Disabling UDLD on Nonfiber-Optic Interfaces, page 19-3
- Disabling UDLD on Fiber-Optic Interfaces, page 19-4
- Resetting Disabled Interfaces, page 19-4

Enabling UDLD Globally

To enable UDLD globally on all fiber-optic interfaces on the switch, enter the following command:

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 procedure:

	Task	Command
Step 1	Enable UDLD on a specific interface. On a fiber-optic interface, this command overrides the **udld enable** global configuration command setting.	`Switch(config-if)# udld enable`
Step 2	Verify the configuration.	`Switch# show udld interface`

Disabling UDLD on Nonfiber-Optic Interfaces

To disable UDLD on individual nonfiber-optic interfaces, perform this procedure:

	Task	Command
Step 1	Disable UDLD on a nonfiber-optic interface. **Note** On fiber-optic interfaces, the **no udld enable** command reverts the interface configuration to the **udld enable** global configuration command setting.	`Switch(config-if)# no udld enable`
Step 2	Verify the configuration.	`Switch# show udld interface`

Disabling UDLD on Fiber-Optic Interfaces

To disable UDLD on individual fiber-optic interfaces, perform this procedure:

	Task	Command
Step 1	Disable 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.	`Switch(config-if)# udld disable`
Step 2	Verify the configuration.	`Switch# show udld interface`

Resetting Disabled Interfaces

To reset all interfaces that have been shut down by UDLD, enter the following command:

Command	Purpose
`Switch# udld reset`	Resets all interfaces that have been shut down by UDLD.

CHAPTER 20

Understanding and Configuring QoS

This chapter describes how to configure quality of service (QoS) on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter consists of the following sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Overview of QoS

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.

The QoS implementation for this release is based on the DiffServ architecture, an emerging standard from the Internet Engineering Task Force (IETF). 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 20-1:

- 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.

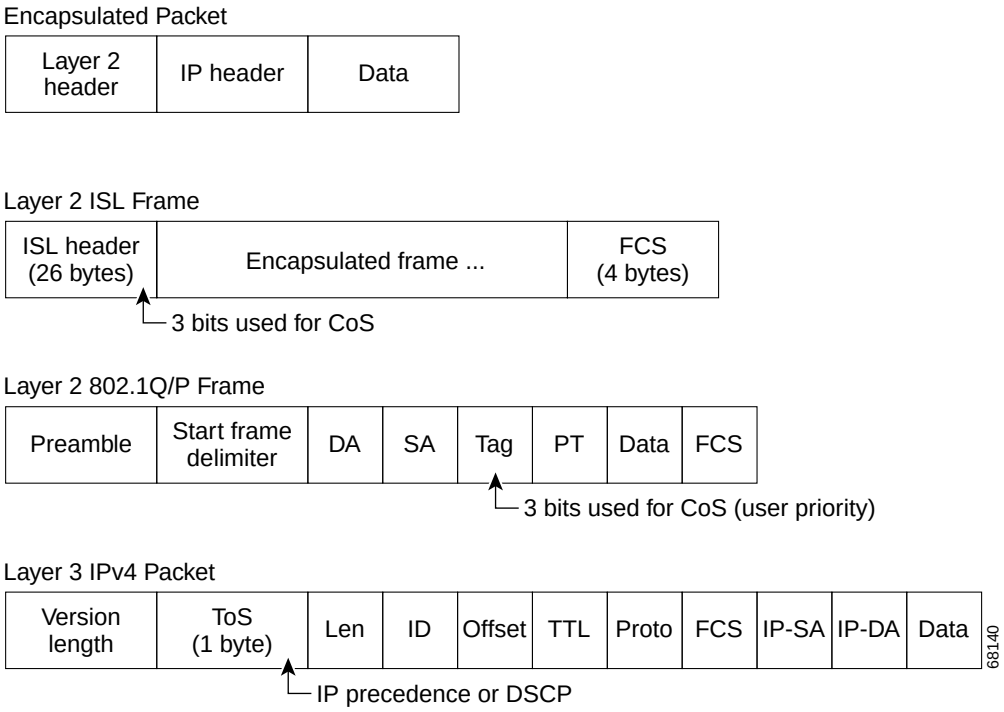
- Prioritization bits in Layer 3 packets:

  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 20-1 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 20-32).

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 20-1).

Table 20-1 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
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 20-12.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.

Basic QoS Model

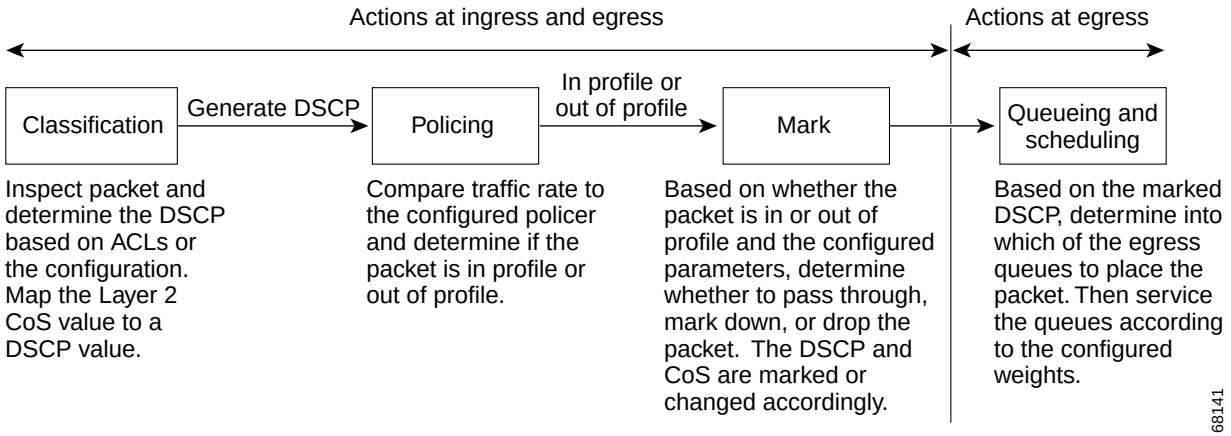
Figure 20-2 shows the basic QoS model. Actions at the ingress and egress interfaces include classifying traffic, policing, and marking:

- 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 20-6.
- 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 20-9.
- 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 20-9.

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 20-13.

Figure 20-2 Basic QoS Model



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 20-3.

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.

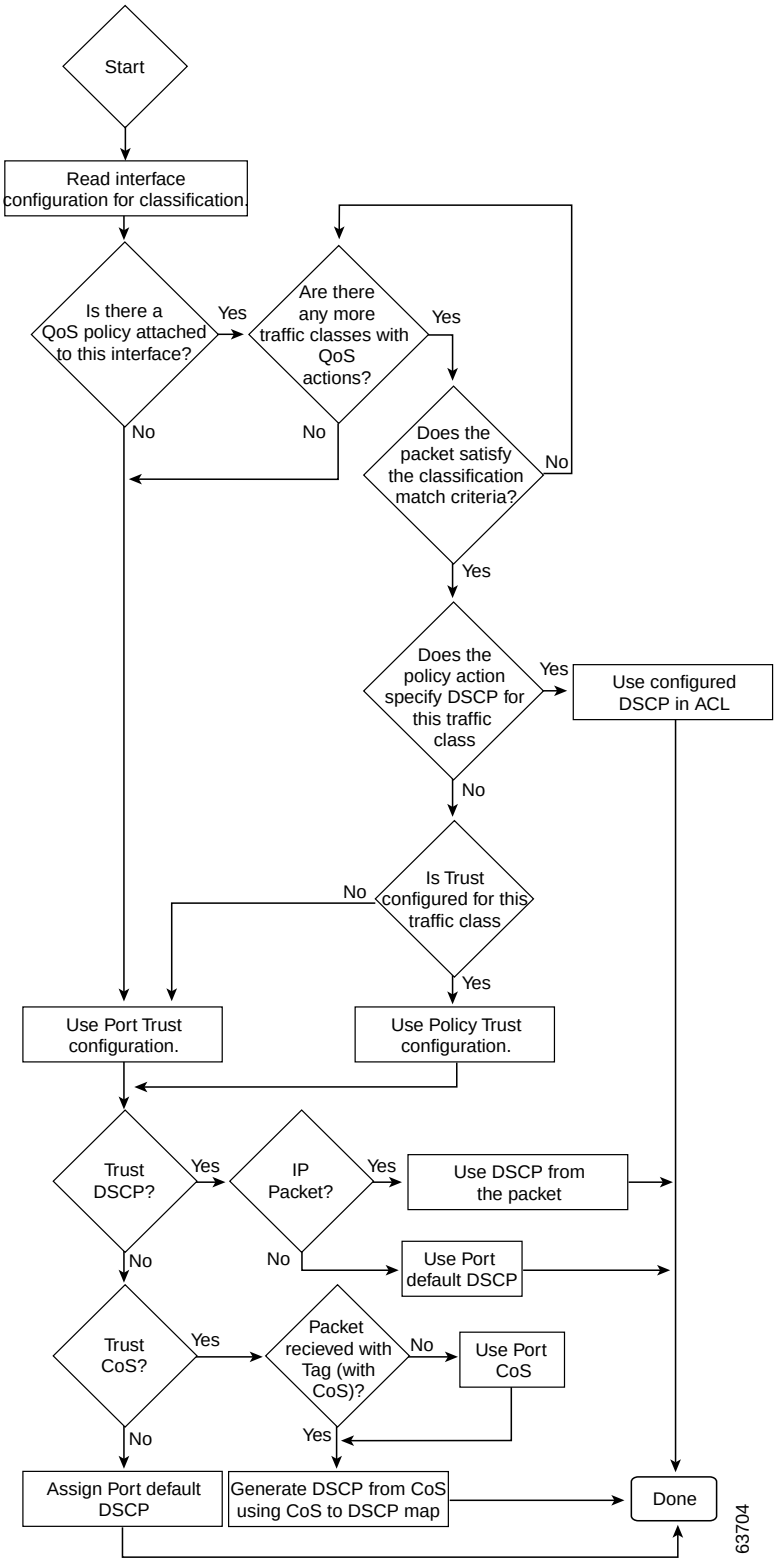
  The trust DSCP configuration is meaningless for non-IP traffic. If you configure a port with trust DSCP and non-IP traffic is received, the switch assigns the default port DSCP.

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.

For information on the maps described in this section, see the “Mapping Tables” section on page 20-12. For configuration information on port trust states, see the “Configuring the Trust State of Interfaces” section on page 20-27.

Figure 20-3 Classification Flowchart



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 Classification Based on Class Maps and Policy Maps, page 20-8.

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 20-19.

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 or IP precedence 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 20-9.

A policy map also has these characteristics:

- A policy map can contain up to eight 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 20-19.

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 20-4.

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 20-12.

You can create these types of policers:

- Individual

  QoS applies the bandwidth limits specified in the policer separately to each matched traffic class. You configure this type of policer within a policy map by using the **police** policy-map configuration command.

- 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.

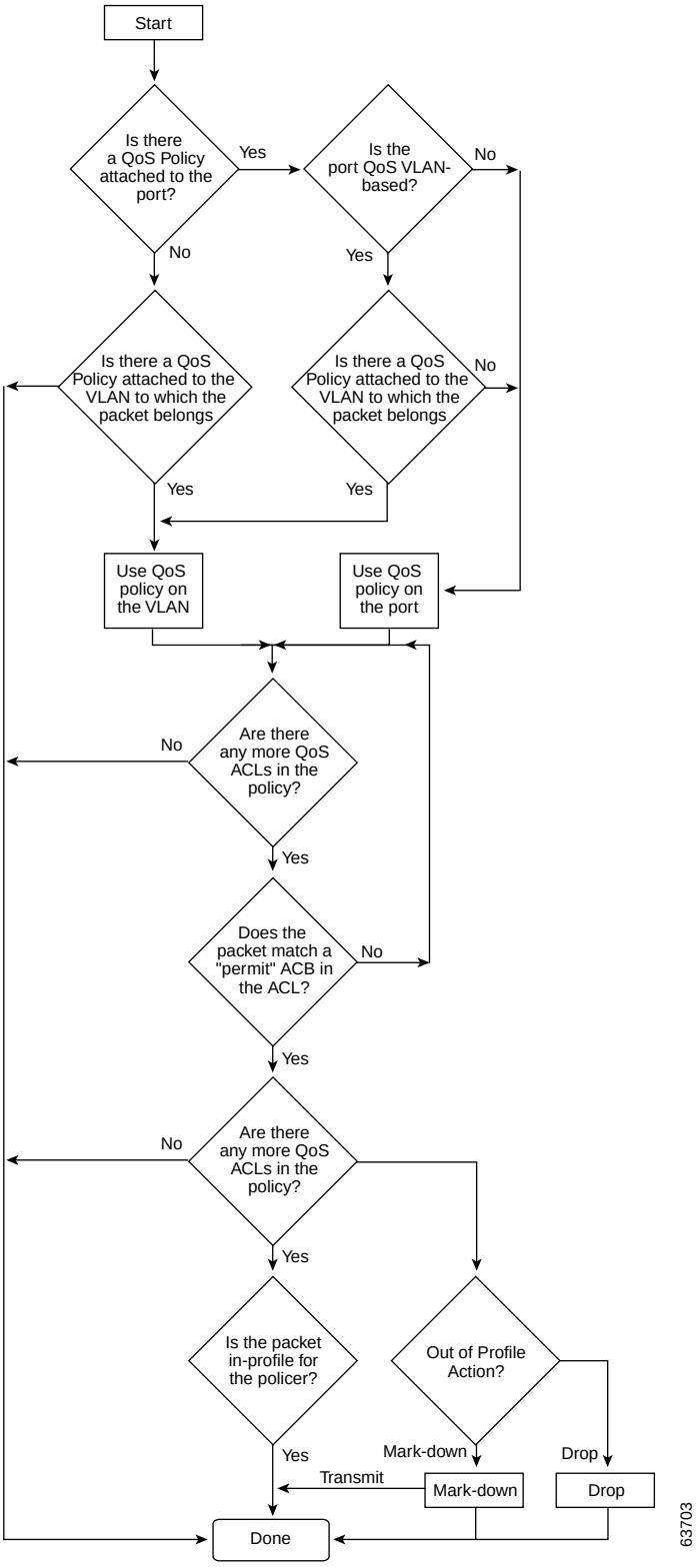
When configuring policing and policers, keep these items in mind:

- By default, no policers are configured.
- Only the average rate and committed burst parameters are configurable.
- Policing can occur on ingress and egress interfaces:
  - 1024 policers are supported on ingress ports.
  - 1024 policers are supported on egress ports.

  - All policers can be individual or aggregate.
  - Two input and two output policers are reserved and used for “no policing” policers.
- 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 20-19 and the “Creating Named Aggregate Policers” section on page 20-17.

Figure 20-4 Policing and Marking Flowchart



Internal DSCP Values

The following sections describe the internal DSCP values:

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 20-32).

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 20-1 on page 20-4).

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 20-34). 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 will 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 20-32.

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 20-29.

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.

Bandwidth can only be configured on:

- Uplink ports on Supervisor Engine III (WS-X4014)
- Ports on the WS-X4306-GB linecard.
- The 2 1000BASE-X ports on the WS-X4232-GB-RJ linecard
- The first 2 ports on the WS-X4418-GB linecard
- The two 1000BASE-X ports on the WS-X4412-2GB-TX linecard

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 will be queued and transmitted at the configured rate. If the burst of traffic, exceeds the size of the queue, packets will be 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 will be queued and transmitted at the configured rate. If the burst of traffic exceeds the size of the queue, packets will be 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

Configuring QoS

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 4006 switch with Supervisor Engine III:

- Default QoS Configuration, page 20-15
- Enabling QoS Globally, page 20-16
- Creating Named Aggregate Policers, page 20-17
- Configuring a QoS Policy, page 20-19
- Enabling or Disabling QoS on an Interface, page 20-25
- Enabling or Disabling QoS on an Interface, page 20-25

Default QoS Configuration

Table 20-2 shows the QoS default configuration.

Table 20-2 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
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

Table 20-2 QoS Default Configuration (continued)

Feature	Default Value
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.
- It is not possible to match IP fragments against configured IP extended ACLs to enforce QoS. IP fragments are transmitted as best effort. IP fragments are denoted by fields in the IP header.
- 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.
- If you want to use the set command in the policy map, you must enable IP routing (disabled by default) and configure an IP default route to send traffic to the next-hop device that is capable of forwarding.

Note QoS processes both unicast and multicast traffic.

Enabling QoS Globally

To enable QoS globally, perform this procedure:

	Task	Command
Step 1	Enable QoS on the switch. Use the **no qos** command to globally disable QoS.	`Switch(config)# qos`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show qos`

This example shows how to enable QoS globally:

```
Switch(config)# qos
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos
  QoS is enabled globally

Switch#
```

Creating Named Aggregate Policers

To create a named aggregate policer, enter the following command:

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 20-16.
- 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)# 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 sections describe QoS policy configuration:

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 16, "Configuring Network Security," for information about ACLs on the Catalyst 4000 family 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 20-19).
- **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 sections describe class map configuration:

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, enter the following command:

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, enter one of the following commands:

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 20-19.
`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 any`	(Optional) Matches any IP traffic or non-IP traffic.

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 the port on which the packet is received, be configured to **trust dscp**. If the incoming port trust state is not set to **trust dscp**, the IP packet DSCP/IP-precedence is not used for matching the traffic; instead the receiving port's default DSCP is used.

Note The interfaces on Catalyst 4006 switch with Supervisor Engine III 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.

Note The Catalyst 4006 switch with Supervisor Engine III supports up to eight match statements per class map. Even though you can configure more than eight match statements from the CLI, only the first eight supported match statements in the class map are used.

Verifying Class-Map Configuration

To verify class-map configuration, perform this procedure:

	Task	Command
Step 1	Exit configuration mode.	`Switch (config-cmap)# end`
Step 2	Verify the configuration.	`Switch# show class-map class_name`

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#
```

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 20-21
- Configuring Policy-Map Class Actions, page 20-22

Creating a Policy Map

To create a policy map, enter the following command:

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

Note The Catalyst 4006 switch with Supervisor Engine III supports up to eight class actions per policy map. Even though you can configure more than eight class actions from the CLI, only the first eight supported class actions in the policy map are used.

These sections describe policy-map class action configuration:

- Configuring the Policy-Map Class Trust State, page 20-22
- Configuring Policy-Map Class Policing, page 20-22
- Using a Named Aggregate Policer, page 20-22
- Configuring a Per-Interface Policer, page 20-23

Configuring the Policy-Map Class Trust State

To configure the policy-map class trust state, enter the following command:

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 20-12). 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 Policy-Map Class Policing

These sections describe configuration of policy-map class policing:

- Using a Named Aggregate Policer, page 20-22
- Configuring a Per-Interface Policer, page 20-23

Using a Named Aggregate Policer

To use a named aggregate policer (see the "Creating Named Aggregate Policers" section on page 20-17), enter the following command:

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 20-9), enter the following command:

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 32000000

Note See the "Configuration Guidelines" section on page 20-16.

- 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 the "Configuring the Policed-DSCP Map" section on page 20-33).
  - 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**, 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)# police 2000000000 2000000 conform-action transmit exceed-action
policed-dscp-transmit
Switch(config-pmap-c)# end
```

Verifying Policy-Map Configuration

To verify policy-map configuration, perform this procedure:

	Task	Command
Step 1	Exit policy-map class configuration mode. **Note** Enter additional **class** commands to create additional classes in the policy map.	`Switch(config-pmap-c)# end`
Step 2	Verify the configuration.	`Switch# show policy-map policy_name`

Note The Catalyst 4006 switch with Supervisor Engine III supports up to eight class actions per policy map. Even though you can configure more than eight class actions from the CLI, only the first eight supported class actions in the policy map are used.

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
   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 procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface | Port-channel number}`
Step 2	Attach a policy map to the input direction of the interface. Use the **no** keyword to detach a policy map from an interface.	`Switch(config-if)# [no] service-policy input policy_map_name`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show policy-map interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface}`

This example shows how to attach the policy map named **pmap1** to Fast Ethernet interface 5/36:

```
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
```

This example shows how to verify the configuration:

```
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#
```

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 procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface | Port-channel number}`
Step 2	Enable QoS on the interface. Use the **no** keyword to disable QoS on an interface.	`Switch(config-if)# [no] qos`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show qos interface`

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 20-25).

To configure VLAN-based QoS on a Layer 2 interface, perform this procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface | Port-channel number`
Step 2	Configure VLAN-based QoS on a Layer 2 interface. Use the **no** keyword to disable VLAN-based QoS on an interface.	`Switch(config-if)# [no] qos vlan-based`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show qos`

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 procedure:

	Task	Command
Step 1	Selects the interface to configure.	`Switch(config)# interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface | Port-channel number}`
Step 2	Configure the trust state of an interface. Use the **no** keyword to clear a configured value and return to the default.	`Switch(config-if)# [no] qos trust [dscp | cos]`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show qos`

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 will always use the interface DSCP and not the incoming packet DSCP.

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 procedure:

	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 cos5
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 procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface | Port-channel number`
Step 2	Configure the ingress interface DSCP value. Use the **no** keyword to clear a configured value and return to the default.	`Switch(config-if)# [no] qos dscp default_dscp`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Verify the configuration.	`Switch# show qos interface {fastethernet | gigabitethernet} slot/interface`

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 describes how to configure the transmit queues:

- Mapping DSCP Values to Specific Transmit Queues, page 20-30
- Allocating Bandwidth Among Transmit Queues, page 20-31
- Configuring Traffic Shaping of Transmit Queues, page 20-31
- Configuring a High Priority Transmit Queue, page 20-32

Depending on the complexity of your network and your QoS solution, you might need to perform all of the procedures in the next sections. Before You will need to make decisions about these characteristics:

- 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 procedure:

	Task	Command
Step 1	Map 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.	`Switch(config)# [no] qos map dscp dscp-values to tx-queue queue-id`
Step 2	Exit configuration mode.	`Switch(config)# end`
Step 3	Verify the configuration.	`Switch# show qos maps dscp tx-queues`

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 5 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 procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface gigabitethernet slot/interface`
Step 2	Select the transmit queue to configure.	`Switch(config-if)# tx-queue queue_id`
Step 3	Set the bandwidth rate for the transmit queue. Use the **no** keyword to reset the transmit queue bandwidth ratios to the default values.	`Switch(config-if-tx-queue)# [no] bandwidth rate`
Step 4	Exit configuration mode.	`Switch(config-if-tx-queue)# end`
Step 5	Verify the configuration.	`Switch# show qos interface`

The bandwidth rate varies with the interface.

Bandwidth can only be configured on:

- Uplink ports on Supervisor Engine III (WS-X4014)
- Ports on the WS-X4306-GB linecard.
- The 2 1000BASE-X ports on the WS-X4232-GB-RJ linecard
- The first 2 ports on the WS-X4418-GB linecard
- The two 1000BASE-X ports on the WS-X4412-2GB-TX linecard

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 procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface`
Step 2	Select the transmit queue to configure.	`Switch(config-if)# tx-queue queue_id`
Step 3	Set the transmit rate for the transmit queue. Use the **no** keyword to clear the transmit queue maximum rate.	`Switch(config-if-tx-queue)# [no] shape rate`
Step 4	Exit configuration mode.	`Switch(config-if-tx-queue)# end`
Step 5	Verify the configuration.	`Switch# show qos interface`

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 procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface`
Step 2	Select transmit queue 3 to configure.	`Switch(config-if)# tx-queue 3`
Step 3	Set the transmit queue to high priority. Use the **no** keyword to clear the transmit queue priority.	`Switch(config-if)# [no] priority high`
Step 4	Exit configuration mode.	`Switch(config-if)# end`
Step 5	Verify the configuration.	`Switch# show qos interface`

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 20-32
- Configuring the Policed-DSCP Map, page 20-33
- Configuring the DSCP-to-CoS Map, page 20-34

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 20-3 shows the default CoS-to-DSCP map.

Table 20-3 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.

Beginning in privileged EXEC mode, follow these steps to modify the CoS-to-DSCP map:

	Task	Command
Step 1	Enter global configuration mode.	`Switch# configure terminal`
Step 2	Modify 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.	`Switch(config)# qos map cos` *cos1 ... cos8* `to dscp` *dscp*
Step 3	Return to privileged EXEC mode.	`Switch(config)# end`
Step 4	Verify your entries.	`Switch# show qos maps cos-dscp`
Step 5	(Optional) Save your entries in the configuration file.	`Switch# copy running-config startup-config`

To return to the default map, use the **no qos cos to dscp** global configuration command.

This example shows how to modify and display the CoS-to-DSCP map:

```
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)#
```

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.

Beginning in privileged EXEC mode, follow these steps to modify the policed-DSCP map:

	Task	Command
Step 1	Enter global configuration mode.	`Switch# configure terminal`
Step 2	Modify 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.	`Switch(config)# qos map dscp policed` *dscp-list* `to dscp` *mark-down-dscp*
Step 3	Return to privileged EXEC mode.	`Switch(config)# end`
Step 4	Verify your entries.	`Switch# show qos maps dscp policed`
Step 5	(Optional) Save your entries in the configuration file.	`Switch# copy running-config startup-config`

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 above 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 20-4 shows the default DSCP-to-CoS map.

Table 20-4 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 these values are not appropriate for your network, you need to modify them.

Beginning in privileged EXEC mode, follow these steps to modify the DSCP-to-CoS map:

	Task	Command
Step 1	Enter global configuration mode.	`Switch# configure terminal`
Step 2	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.	`Switch(config)# qos map dscp` *dscp-list* `to cos` *cos*
Step 3	Return to privileged EXEC mode.	`Switch(config)# end`
Step 4	Verify your entries.	`Switch# show qos maps dscp to cos`
Step 5	(Optional) Save your entries in the configuration file.	`Switch# copy running-config startup-config`

To return to the default map, use the **no qos dscp to cos** global configuration command.

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 above 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.

CHAPTER 21

Configuring SPAN

This chapter describes how to configure the Switched Port Analyzer (SPAN) feature on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration guidelines.

This chapter consists of the following 5sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

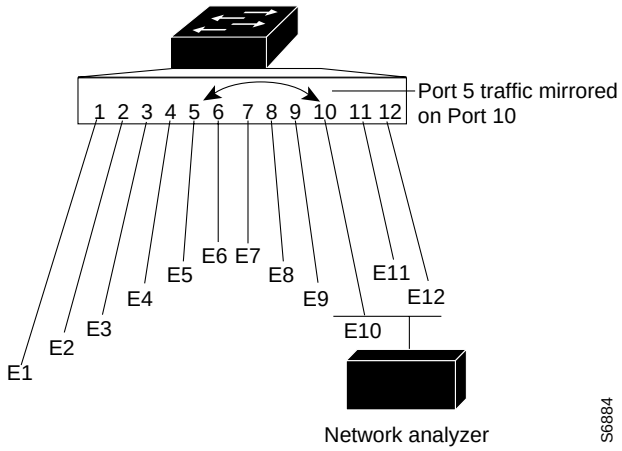
Overview of SPAN

SPAN selects network traffic for analysis by a network analyzer such as a SwitchProbe device or other Remote Monitoring (RMON) probe. 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 21-1, 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; a copy of the packets received or transmitted by the source interfaces are sent to the destination interface.

Figure 21-1 Example SPAN Configuration



The following sections describe how SPAN works:

SPAN Session

A SPAN session is an association of a destination interface with a set of source interfaces; you configure SPAN sessions using parameters that specify the type of network traffic to monitor. SPAN sessions allow you to monitor traffic on one or more interfaces, or one or more VLANs, and send either ingress traffic, egress traffic, or both to a destination interface.

You can configure up to six separate SPAN sessions (2 ingress, 4 egress) with separate or overlapping sets of SPAN source interfaces or VLANs. A bi-directional SPAN session counts as both 1 ingress and 1 egress session. Both switched and routed interfaces can be configured as SPAN sources.

SPAN sessions do not interfere with the normal operation of the switch. When enabled, a SPAN session might become active or inactive based on various events or actions; a syslog message indicates this. The **show monitor session** command displays the operational status of a SPAN session.

A SPAN session will remain inactive after system boot-up until the destination interface is operational.

Destination Interface

A destination interface (also called a *monitor interface*) is a switched or routed interface where SPAN sends packets for analysis. Once an interface becomes an active destination interface, incoming traffic is disabled. You cannot configure a SPAN destination interface to receive ingress traffic. The interface does not forward any traffic except that required for the SPAN session.

An interface specified as a destination interface in one SPAN session cannot be a destination interface for another SPAN session. An interface configured as a destination interface cannot be configured as a source interface. EtherChannel interfaces cannot be SPAN destination interfaces.

Specifying a trunk interface as a SPAN destination interface stops trunking on the interface.

Source Interface

A source interface is an interface monitored for network traffic analysis. One or more source interfaces can be monitored in a single SPAN session with user-specified traffic types (ingress, egress, or both) applicable for all the source interfaces. All sources for a particular SPAN session are spanned in the same direction.

You can configure source interfaces for any VLAN. You can configure VLANs as source interfaces, which means that all interfaces in the specified VLANs are source interfaces for the SPAN session.

Trunk interfaces can be configured as source interfaces and can be mixed with nontrunk source interfaces; however, the destination interface never encapsulates, so you do not see any encapsulation out of the SPAN destination interface.

Traffic Types

Ingress SPAN (Rx) copies network traffic received by the source interfaces for analysis at the destination interface. Egress SPAN (Tx) copies network traffic transmitted from the source interfaces. Specifying the configuration option “both” copies network traffic received and transmitted by the source interfaces to the destination interface.

VLAN-Based SPAN

VLAN-based SPAN analyzes the network traffic in one or more VLANs. You can configure VLAN based-SPAN as ingress SPAN, egress SPAN, or both. All of the interfaces in the source VLANs become source interfaces for the VLAN-based SPAN session.

Use the following guidelines for VLAN-based SPAN sessions:

- Trunk interfaces are included as source interfaces for VLAN-based SPAN sessions.
- For VLAN-based SPAN sessions with both ingress and egress SPAN configured, two packets are forwarded by the SPAN destination interface if the packets get switched on the same VLAN.
- When a VLAN is cleared, it is removed from the source list for VLAN-based SPAN sessions.
- Inactive VLANs are not allowed for VLAN-based SPAN configuration.
- If a VLAN is being ingress monitored and the switch routes traffic from another VLAN to the monitored VLAN, that traffic is not monitored—it is not seen on the SPAN destination interface. Additionally, traffic that gets routed from an egress-monitored VLAN to some other VLAN does not get monitored. VLAN-based SPAN only monitors traffic that leaves or enters the switch, not traffic that gets routed between VLANs.

SPAN Traffic

All network traffic, including multicast and bridge protocol data unit (BPDU) packets, can be monitored using SPAN.

In some SPAN configurations, multiple copies of the same source packet are sent to the SPAN destination interface. For example, a bidirectional (both ingress and egress) SPAN session is configured for sources a1 and a2 to a destination interface d1. If a packet enters the switch through a1 and gets switched to a2, both incoming and outgoing packets are sent to destination interface d1; both packets would be the same (unless a Layer-3 rewrite had occurred, in which case the packets would be different).

SPAN Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring SPAN:

- 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.
- All source interfaces in a given SPAN session must be spanned in the same direction.
- 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.
- 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.
- You cannot configure a SPAN destination interface to receive ingress traffic.
- 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.
- All CPU generated out-going traffic (i.e. CDP, BPDU, ARP reply, etc.) are not picked up by the SPAN destination interface. The incoming CPU traffic from its neighbor is picked up by the SPAN destination interface.

Configuring SPAN

The following sections describe how to configure SPAN:

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

Configuring SPAN Sources

To configure the source for a SPAN session, enter the following command:

Command	Purpose
`Switch(config)# [no] monitor session {session_number} {source {interface type/num} | {vlan vlan_ID}} [, | - | rx | tx | both]`	Specifies the SPAN session number (1 through 6), the source interfaces (FastEthernet or GigabitEthernet), or VLANs (1 through 1005), and the traffic direction to be monitored. 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
```

Configuring SPAN Destinations

To configure the destination for a SPAN session, enter the following command:

Command	Purpose
`Switch(config)# [no] monitor session {session_number} {destination {interface type/num} }`	Specifies the SPAN session number (1 through 6) and the destination interfaces or VLANs. Use the **no** keyword to restore the defaults.

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, enter the following command:

Command	Purpose
`Switch(config)# [no] monitor session {session_number} {filter vlan {vlan_ID} [, | - ]}`	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. 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 1005. 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#
```

CHAPTER 22

Power Management and Environmental Monitoring

This chapter describes power management and environmental monitoring features in the Catalyst 4006 switch with Supervisor Engine III. It provides guidelines, procedures, and configuration examples.

This chapter consists of the following major sections:

Note For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and the publications at http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm

Understanding How Environmental Monitoring Works

Environmental monitoring of chassis components provides early warning indications of possible component failure. This warning helps to ensure safe and reliable system operation and avoid network interruptions. This section describes the monitoring of these critical system components, which will help you to identify and rapidly correct hardware-related problems in your system.

Environmental Monitoring Using CLI Commands

This section describes the **show environment** CLI command.

Enter the **show environment** [**alarm** | **status** | **temperature**] command to display system status information. Keyword descriptions are listed in Table 22-1. For more information on these commands, refer to the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III*.

Table 22-1 show environment Keyword Descriptions

Keyword	Purpose
alarm	Displays environmental alarms

Table 22-1 show environment Keyword Descriptions (continued)

Keyword	Purpose
status	Displays field-replaceable unit (FRU) operational status and power and power supply fan sensor information
temperature	Displays temperature of the chassis

The following example shows how to display the environment conditions:

```
Switch#show env
no alarm

Chassis Temperature                    = 32 degrees Celsius
Chassis Over Temperature Threshold     = 75 degrees Celsius
Chassis Critical Temperature Threshold = 95 degrees Celsius

Power                                          Fan
Supply  Model No        Type       Status      Sensor
------  --------------  ---------  -------     ------
PS1     PWR-C4K-400AC   AC  400W   good        good
PS2     none            --         --          --

Chassis Type :WS-C4006

Supervisor Led Color :Green

Fantray :good

Switch#
```

LED Indications

There are two alarm types, major and minor. Major alarms indicate a critical problem that could lead to system shutdown. Minor alarms are just informational, alerting you to a problem that could turn critical if corrective action is not taken.

When the system has an alarm (major or minor) indicating an over-temperature condition, the alarm is not canceled or any action taken (such as module reset or shutdown) for 5 minutes. If the temperature falls 5°C below the alarm threshold during this period, the alarm is canceled.

Table 22-2 lists the LEDs that are used as environmental indicators for the supervisor engine and switching modules.

Note Refer to the *Catalyst 4000 Family Module Installation Guide* for additional information on LEDs, including the supervisor engine system LED.

Table 22-2 Alarms for Supervisor Engine and Switching Modules

Event	Alarm Type	LED Color	Action
Supervisor engine temperature sensor exceeds major threshold[1]	Major	Status[2] LED red	Syslog message. If the overtemperature condition is not corrected, the system shuts down after 5 minutes. Alarm thresholds: • Chassis temperature = 32°C • Chassis over temperature threshold = 75°C • Chassis critical temperature threshold = 95°C
Supervisor engine temperature sensor exceeds minor threshold	Minor	Status LED orange	Syslog message. Monitor the condition.

1. Temperature sensors monitor key supervisor engine components including daughter cards.
2. A Status LED is located on the supervisor engine front panel and all module front panels.

Power Management

This section describes power management in the Catalyst 4006 switch and includes the following major sections:

Power Redundancy

The power redundancy feature is designed to support an optimized Catalyst 4006 chassis consisting of a Supervisor Engine III and four WS-X4148-RJ or WS-X4148-RJ21 modules operating in 1+1 redundancy mode (one primary power supply plus one redundant power supply). Additional configurations are possible.

In systems with redundant power supplies, it is recommended that both power supplies be of the same wattage. The Catalyst 4000 family switches allow you to mix AC-input and DC-input power supplies in the same chassis. For detailed information on supported power supply configurations for each chassis, refer to the *Catalyst 4000 Family Installation Guide*.

Catalyst 4000 family modules have different power requirements; thus, some switch configurations require more power than 1+1 redundancy mode (a single power supply) can provide. The 2+1 redundancy mode requires three power supplies (2 primary power supplies plus one redundant power supply). The default is 2+1 redundancy mode.

The Catalyst 4006 switch contains holding bays for up to three power supplies. You need two primary power supplies to operate a fully loaded Catalyst 4006 chassis. You can set the power redundancy to 2+1 redundancy mode or 1+1 redundancy mode. The 1+1 redundancy mode may not support a fully loaded chassis.

If your switch has only two power supplies and is in 2+1 redundancy mode, there is no redundancy. You can create redundancy with only two power supplies by setting the power redundancy to operate in 1+1 redundancy mode (one primary plus one redundant power supply). However, 1+1 redundancy will not support all configurations.

The 1+1 redundancy mode is designed and optimized for a specific hardware configuration: a Catalyst 4006 chassis with a Supervisor Engine III and four WS-X4148-RJ or WS-X4148-RJ21 modules. Although other configurations are possible, we do not recommend that you use them without careful consideration of the power usage in the system. For example, other similar and possible configurations can consist of four modules, where each module consumes 65W or less (for a total of 265W) or, more generally, where the total module power usage does not exceed the absolute maximum module usage of 265W.

The Supervisor Engine III uses 110W, and the fan box uses 25W, for a system total of 400W (modules + supervisor + fan). If you opt to use the 1+1 redundancy mode, the type and number of modules supported are limited by the power available from a single power supply. To determine the power consumption for each module in your chassis, see the "Power Consumption of Modules" section on page 22-5.

To choose a 1+1 redundancy configuration, you must change the system configuration from the default 2+1 redundancy mode to 1+1 redundancy mode by using the **power supplies required 1** command. The **power supplies required 1** command sets the power redundancy to 1+1 redundancy mode. In the 1+1 redundancy mode, the nonredundant power available to the system is the power of the single weakest power supply. The second power supply installed in your switch provides full redundancy.

Limitations of the 1+1 Redundancy Mode

If you attempt to configure the system to operate in 1+1 redundancy mode, and you have more modules installed in the chassis than a single power supply can handle, the system displays the following error message: `Insufficient power supplies for the current chassis configuration`. This message will also appear in the **show power** command output.

If you are already operating in 1+1 redundancy mode with a valid module configuration and you attempt to insert additional modules that require more power than the single power supply provides, the system immediately places the newly inserted module into reset mode and issues these error messages: `Module has been inserted` and `Insufficient power supplies operating`. Additionally, if a chassis that has been operating in 1+1 redundancy mode with a valid module configuration is powered down, and you insert a module or change the module configuration inappropriately and power on the switch again, the module(s) in the chassis (at boot up) that require more power than is available, are placed into reset mode.

Both of these scenarios initiate the five-minute evaluation countdown timer. When this timer runs out, the system attempts to resolve this power usage limitation by evaluating the type and number of modules installed. The evaluation process may require several cycles to stabilize the chassis' power usage.

During the evaluation cycle, the modules are in effect removed and re-inserted, thus causing disruption of network connectivity; the switch reactivates only the modules it is able to support with the limited power available and leaves the remaining modules in reset mode. The supervisor engine always remains enabled. Modules placed in reset mode still consume some power. If the chassis module combination, including the modules in reset, still require more power than is available, the five-minute evaluation countdown timer starts for another evaluation cycle, and additional modules are placed in reset mode until the power usage is stable.

If the power requirement of the active modules and with the modules in reset does not exceed the available power, the system is stable and no more evaluation cycles are run, until something again causes insufficient power usage. One or possibly two cycles are required to stabilize the system. If you configure the chassis correctly, the system will not enter the evaluation cycle.

Note If you have all three power supplies installed and you still choose to operate in 1+1 redundancy mode but later add additional modules which exceed the power available, the five-minute evaluation timer starts again. The switch may require several evaluation cycles to stabilize the system. To correct this power situation caused by the additional modules, you can either remove the extra modules or change the power redundancy to 2+1 redundancy mode. If you choose to return to the 2+1 redundancy mode, each module held in reset mode is brought up one-by-one to an operational state.

A module in reset continues to draw power as long as it is installed in the chassis; however, the **show module** command output indicates that there is not enough power for the module.

To compute the power requirements for your system and verify that your system has enough power, add up the power consumed by the supervisor engine module, the fan box, and the modules you wish to use. A single power supply provides 400W, and two power supplies provide 750W. For 1+1 redundancy mode, verify that the total is less than 400W.

For example, the following configuration requires a minimum of 395W:

- WS-X4014 Supervisor Engine III—110W
- Four WS-X4148-RJ modules—65W each (260W total—the optimized module configuration)
- Fan box—25W

This is 5W less than the maximum that a single power supply can provide in 1+1 redundancy mode.

The following configuration, however, requires more power than a single power supply can supply:

- WS-X4013 supervisor engine—110W
- Two WS-X4148-RJ modules, in slots 2 and 3—65W each (130W total)
- Two WS-X4448-GB-LX modules in slots 4 and 5—90W each (180W total)
- Fan box—25W

This configuration requires 445W and cannot be used in 1+1 redundancy mode.

Remember, when considering the 1+1 redundancy mode, you must carefully plan the configuration of the module power usage of your chassis. An incorrect configuration will momentary disrupt your system during the evaluation cycle. To avoid this disruption, carefully plan your configuration to ensure it is within the power limits, or return to the default 2+1 redundancy configuration by installing a third power supply in your switch and set the power redundancy to 2+1 redundancy mode.

Use the **power supplies required 2** command to set the power redundancy to the 2+1 redundancy mode.

Power Consumption of Modules

A single power supply provides 400W; two power supplies provide 750W. When operational, the supervisor engines consume no more than 110W and the fan box consumes 25W. For power consumption of common Catalyst 4006 modules, see Table 22-3.

Enter the **show power** command to display the current power redundancy and the current system power usage.

Table 22-3 Power Consumption Rates for Catalyst 4006 Modules

Module	Power Consumed During Operation (W)	Power Consumed in Reset Mode (W)
6-port 1000BASE-X (GBIC) Gigabit Ethernet WS-X4306-GB	35	30
32-port 10/100 Fast Ethernet RJ-45 WS-X4232-RJ-XX	55	35
Catalyst 4000 Access Gateway Module with IP/FW IOS WS-X4604-GWY	120	60
24-port 100BASE-FX Fast Ethernet switching module WS-X4124-FX-MT	90	75
32-port 10/100 Fast Ethernet RJ-45, plus 2-port 1000BASE-X (GBIC) Gigabit Ethernet WS-X4232-GB-RJ	50	35
48-port 100BASE-FX Fast Ethernet switching module WS-X4148-FX-MT	120	10
18-port server switching 1000BASE-X (GBIC) Gigabit Ethernet WS-X4418-GB	80	50
Catalyst 4006 Backplane Channel Module WS-X4019	10	10
48-port 10/100 Fast Ethernet RJ-45 WS-X4148-RJ	65	40
Catalyst 4003 and 4006 Layer 3 Services Module WS-X4232-L3	120	70
12-port 1000BASE-T Gigabit Ethernet, plus 2-port 1000BASE-X (GBIC) Gigabit Ethernet WS-X4416	110	70
24-port 1000BASE-X Gigabit Ethernet WS-X4424-GB-RJ45	90	50
48-port 1000BASE-X Gigabit Ethernet WS-X4448-GB-RJ45	120	72
48-port 1000BASE-X Gigabit Ethernet WS-X4448-GB-LX	90	50
48-port Telco 10/100BASE-TX switching module WS-X4148-RJ21	65	40
48-port inline power 10/100BASE-TX switching module WS-X4148-RJ45V	60	50
4-port MT-RJ uplink module WS-U4504-FX-MT	10	10

Setting the Power Redundancy Mode

To configure the power redundancy mode, perform the following procedure:

	Task	Command
Step 1	Enter configuration mode	`Switch# configure terminal`
Step 2	Set the power redundancy mode.	`Switch(config)# power supplies required {1 | 2}`
Step 3	Verify the power redundancy mode and the current power usage for the switch.	`Switch(config)# show power`

The default power redundancy mode is 2 (2+1) redundancy mode.

The following example shows how to set the power redundancy mode to 1 (1+1 redundancy mode).

```
Switch (config)# power supplies required 1
Switch (config)#
```

The following example shows how to display the current power status of system components and the power redundancy mode:

```
Switch# show power
Power                                       Fan
Supply  Model No        Type       Status   Sensor
------  --------------  ---------  -------  ------
PS1     WS-X4008        AC 400W    good     good
PS2     WS-X4008        AC 400W    good     good
PS3     none            --         --       --
PEM     none            --         --       --


Power Summary
 (in Watts)    Available   Used   Remaining
-------------  ---------  ------  ---------
System Power    750        255     495
Inline Power    0          0       0

Power supplies needed by system = 2
Switch#
```

The following example shows the **show module** command output for a system with inadequate power for all installed modules.

```
Switch# show module

Mod  Ports Card Type                                Model             Serial No.
----+-----+---------------------------------------+-----------------+--------
---
 1      2  1000BaseX (GBIC) Supervisor Module     WS-X4014
JAB06070600
 2     48  10/100BaseTX (RJ45)                    WS-X4148
JAB040409GK
 3     48  10/100BaseTX (RJ45)                    WS-X4148
JAB023402RM
```

```
 M MAC addresses                   Hw  Fw                Sw              Stat
--+--------------------------------+---+-----------------+---------------+-----
 1 0004.27b6.5a00 to 0004.27b6.5dff 1.2 12.1(11r)EW       12.1(11)EK2(0.2 Ok
 2 00b0.6465.2cf0 to 00b0.6465.2d1f 2.3                                   Ok
 3 0050.0f10.2850 to 0050.0f10.287f 1.0                                   Ok
 4 0001.6445.c3e0 to 0001.6445.c40f 0.0                                   Other
 5 0030.850e.2dc8 to 0030.850e.2ddf 0.7                                   Other
 6 0002.b9f1.e340 to 0002.b9f1.e36f 1.0                                   Other
Switch#
```

Configuring Inline Power

If your switch has a module capable of providing inline power to end stations, you can set each interface on the module to automatically detect and apply inline power if the end station requires power.

To set an interface to automatically detect an end station requiring power, and apply the inline power, perform this procedure:

	Task	Command
Step 1	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 2	Set the interface to automatically detect if the end device requires power and apply inline power to the end device if necessary. Use the **never** keyword to disable detection and power for the inline-power capable interface.	`Switch(config-if)# power inline {auto | never}`
Step 3	Exit configuration mode.	`Switch(config-if)# end`
Step 4	Display the inline power state for the switch.	`Switch# show power inline {fastethernet | gigabitethernet} slot/port`

If you set a non-inline-power-capable interface to automatically detect and apply power, an error message indicates that the configuration is not valid.

This example shows how to set the Fast Ethernet interface 5/8 to automatically detect inline power and send power through that interface.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/8
Switch(config-if)# power inline auto
Switch(config-if)# end
```

This example shows how to verify the inline power configuration for the Fast Ethernet interface 5/8.

```
Switch# show power inline fastethernet 5/8

Interface            Admin  Oper      Power ( mWatt )  Device
-------------------- ----- ---------- --------------- -----------
FastEthernet5/8      auto  on         400             cisco phone device
Switch#
```

If a Cisco IP phone is connected to an interface with external power, the switch does not recognize the phone. The **Device** column in the **show power inline command** displays as unknown.

The following example shows 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
```

CHAPTER 23

Configuring Voice Interfaces

The Catalyst 4006 switch with Supervisor Engine III 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.

The Catalyst 4006 switch with Supervisor Engine III has extensive QoS capabilities so that voice traffic can be prioritized and serviced differently from the data traffic. QoS uses classification and scheduling to transmit network traffic from the switch in a predictable manner. The Cisco 7960 IP phone itself is also a configurable device, and you can configure it to forward traffic with an 802.1p priority. You can use the CLI to configure the Catalyst 4006 switch with Supervisor Engine III 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 to the following devices:

- Port 1 connects to the Catalyst 4006 switch with Supervisor Engine III or other voice-over-IP device.
- Port 2 is an internal 10/100 interface that carries the phone traffic.
- Port 3 connects to a PC or other device.

Figure 23-1 shows one way to configure a Cisco 7960 IP phone.

Figure 23-1 Cisco 7960 IP phone Connected to a Catalyst 4006 Switch with Supervisor Engine III

Configuring Voice Ports to Carry Voice and Data Traffic on Different VLANs

You can configure a switch port to forward voice and data traffic on different virtual LANs (VLANs).

Beginning in privileged EXEC mode, follow these steps to configure a port to receive voice and data from a Cisco IP Phone in different VLANs:

	Purpose	Command
Step 1	Enter configuration mode.	`Switch# configure terminal`
Step 2	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 3	Instruct the Cisco IP Phone to forward all voice traffic through the *vlan_num* VLAN. The Cisco IP Phone forwards the traffic with an 802.1p priority of 5.	`Switch(config-if)# switchport voice vlan vlan_num`
Step 4	Return to privileged EXEC mode.	`Switch(config-if)# end`
Step 5	Verify the configuration.	`Switch# show interface {fastethernet | gigabitethernet} slot/port switchport`

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
Switch(config)# interface fastethernet 2/5
SSwitch(config-if)# switchport voice vlan 2
switchport voice vlan 2
Switch(config-if)# end
Switch# show interface fastethernet 2/5 switchport
show interface fastethernet 2/5 switchport
Name:Fa2/5
Switchport:Enabled
Administrative Mode:dynamic auto
Operational Mode:down
Administrative Trunking Encapsulation:negotiate
Negotiation of Trunking:On
Access Mode VLAN:1 (default)
Trunking Native Mode VLAN:1 (default)
Voice VLAN:2 ((Inactive))
Appliance trust:none
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#
```

Configuring a Port to Connect to a Cisco 7690 IP Phone

Because a Cisco 7960 IP phone also supports connection to a PC or other device, an interface connecting a Catalyst 4006 switch with Supervisor Engine III switch to a Cisco 7960 IP phone can carry mixed 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, and all traffic is in the same VLAN.
- Voice and data traffic are carried on separate VLANs, and voice traffic always has a CoS priority of 5.

Beginning in privileged EXEC mode, follow these steps 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.

	Purpose	Command
Step 1	Enter configuration mode.	`Switch# configure terminal`
Step 2	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 3	Instruct the switch to use 802.1p priority tagging for voice traffic and to use VLAN 1 (default native VLAN) to carry all traffic.	`Switch(config-if)# switchport voice vlan dot1p`
Step 4	Return to privileged EXEC mode.	`Switch(config-if)# end`
Step 5	Verify the port configuration.	`Switch# show interface {fastethernet | gigabitethernet} slot/port switchport`

Overriding the CoS Priority of Incoming Frames

A PC or other data device can connect to a Cisco 7960 IP phone port. The PC can generate packets with an assigned CoS value. If you want, you can use the switch CLI to override the priority of frames arriving on the phone port from connected devices. You can also set the phone port to accept (trust) the priority of frames arriving on the port.

Beginning in privileged EXEC mode, follow these steps to override the CoS priority setting received from the non-voice port on the Cisco 7960 IP phone:

	Purpose	Command
Step 1	Enter configuration mode.	`Switch# configure terminal`
Step 2	Select the interface to configure.	`Switch(config)# interface {fastethernet | gigabitethernet} slot/port`
Step 3	Set 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.	`Switch(config-if)# [no] qos trust extend cos 3`
Step 4	Return to privileged EXEC mode.	`Switch(config-if)# end`
Step 5	Verify the change.	`Switch# show interface {fastethernet | gigabitethernet} slot/port switchport`

Configuring Inline Power

The Catalyst 4006 switch with Supervisor Engine III can supply inline power to the Cisco 7960 IP phone, if necessary. The Cisco 7960 IP phone can also be connected to an AC power source and supply its own power to the voice circuit.

The Catalyst 4006 switch with Supervisor Engine III senses if it is connected to a Cisco 7960 IP phone. If there is *no* power on the circuit, the switch supplies the power. If there is power on the circuit, the switch does not supply it.

You can configure the switch to never supply power to the Cisco 7960 IP phone and to disable the detection mechanism. See the "Configuring Inline Power" section on page 22-8 for the CLI commands that you use to supply inline power to a Cisco 7960 IP phone.

Acronyms

Table A-1 defines the acronyms 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
CHAP	Challenge Handshake Authentication Protocol

Table A-1 Acronyms (continued)

Acronym	Expansion
CIR	committed information rate
CLI	command-line interface
CLNS	Connection-Less Network Service
CMNS	Connection-Mode Network Service
COPS	Common Open Policy Server
COPS-DS	Common Open Policy Server Differentiated Services
CoS	class of service
CPLD	Complex Programmable Logic Device
CRC	cyclic redundancy check
CRF	concentrator relay function
CST	Common Spanning Tree
CUDD	University of Colorado Decision Diagram
DCC	Data Country Code
dCEF	distributed Cisco Express Forwarding
DDR	dial-on-demand routing
DE	discard eligibility
DEC	Digital Equipment Corporation
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
EAP	Extensible Authentication Protocol
EARL	Enhanced Address Recognition Logic

Table A-1 Acronyms (continued)

Acronym	Expansion
EEPROM	electrically erasable programmable read-only memory
EHSA	enhanced high system availability
EIA	Electronic Industries Association
ELAN	Emulated Local Area Network
EOBC	Ethernet out-of-band channel
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
LDA	Local Director Acceleration
LCP	Link Control Protocol
LEC	LAN Emulation Client

Table A-1 Acronyms (continued)

Acronym	Expansion
LECS	LAN Emulation Configuration Server
LEM	link error monitor
LER	link error rate
LES	LAN Emulation Server
LLC	Logical Link Control
LTL	Local Target Logic
MAC	Media Access Control
MD5	Message Digest 5
MFD	multicast fast drop
MIB	Management Information Base
MII	media-independent interface
MLS	Multilayer Switching
MLSE	maintenance loop signaling entity
MOP	Maintenance Operation Protocol
MOTD	message-of-the-day
MLSE	maintenance loops signaling entity
MRM	multicast routing monitor
MSDP	Multicast Source Discovery Protocol
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
OAM	Operation, Administration, and Maintenance
ODM	order dependent merge
OSI	Open System Interconnection
OSPF	open shortest path first
PAE	port access entity
PAgP	Port Aggregation Protocol

Table A-1 Acronyms (continued)

Acronym	Expansion
PBD	packet buffer daughterboard
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
PPP	Point-to-Point Protocol
PRID	Policy Rule Identifiers
PVST+	Per VLAN Spanning Tree+
QM	QoS manager
QoS	quality of service
RACL	router interface access control list
RADIUS	Remote Access Dial-In User Service
RAM	random-access memory
RCP	Remote Copy Protocol
RGMP	Router-Ports Group Management Protocol
RIB	routing information base
RIF	Routing Information Field
RMON	remote network monitor
ROM	read-only memory
ROMMON	ROM monitor
RP	route processor or rendezvous point
RPC	remote procedure call
RPF	reverse path forwarding
RSPAN	remote SPAN
RST	reset
RSVP	ReSerVation Protocol
SAID	Security Association Identifier
SAP	service access point
SCM	service connection manager
SCP	Switch-Module Configuration Protocol
SDLC	Synchronous Data Link Control

Table A-1 Acronyms (continued)

Acronym	Expansion
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
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

Table A-1 Acronyms (continued)

Acronym	Expansion
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

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V