16- and 36-Port Ethernet Switch Module for Cisco 2600 Series, Cisco 3600 Series, and Cisco 3700 Series

Feature History

Release	Modification
12.2(2)XT	This feature was introduced on the Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.
12.2(8)T	This feature was integrated into Cisco IOS Release 12.2(8)T.
12.2(15)ZJ	Added switching software enhancements: IEEE 802.1x, QoS (including Layer 2/Layer 3 CoS/DSCP mapping and rate limiting), security ACL, IGMP snooping, per-port storm control, and fallback bridging support for switch virtual interfaces (SVIs).

This feature module describes the 16- and 36-Port Ethernet Switch Module (NM-16ESW and NM-36ESW) for Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers in Cisco IOS Release 12.2(2)XT and Cisco IOS Release 12.2(8)T and above. Enhancements were added in Cisco IOS Release 12.2(15)ZJ.

This document includes the following sections:

Feature Overview

This document explains how to configure the 16- and 36-port Ethernet switch network modules. This network module is supported on Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers. The Ethernet switch network module is a modular, high-density voice network module that provides Layer 2 switching across Ethernet ports. The 16-port Ethernet switch network module has 16 10/100BASE-TX ports and an optional 10/100/1000BASE-T Gigabit Ethernet port. The 36-port Ethernet switch network module has 36 10/100BASE-TX ports and two optional 10/100/1000BASE-T Gigabit Ethernet ports. The gigabit Ethernet can be used as an uplink port to a server or as a stacking link to another 16- or 36-port Ethernet switch network modules in the same system. The 36-port Ethernet switch network module requires a double-wide slot. An optional power module can also be added to provide inline power for IP telephones.

The 16- and 36-port Ethernet switch network modules support the following:

- Layer 2 Ethernet Interfaces, page 2
- Switch Virtual Interfaces, page 5
- Routed Ports, page 5
- VLAN Trunk Protocol, page 5
- EtherChannel, page 7
- 802.1x Port-Based Authentication, page 8
- Spanning Tree Protocol, page 12
- Cisco Discovery Protocol, page 24
- Switched Port Analyzer, page 24
- Network Security with ACLs, page 25
- Quality of Service, page 29
- Maximum Number of VLAN and Multicast Groups, page 35
- IP Multicast Support, page 35
- IGMP Snooping, page 35
- Global Storm-Control, page 38
- Per-Port Storm-Control, page 40
- Port Security, page 40
- Ethernet Switching in Cisco AVVID Architecture, page 40
- Stacking, page 41
- Flow Control, page 41
- Fallback Bridging, page 42

Layer 2 Ethernet Interfaces

Layer 2 Ethernet Switching

Ethernet switch network modules support simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

The Ethernet switch network module 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 stations 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.

Switching Frames Between Segments

Each Ethernet interface on an Ethernet switch network module can connect to a single workstation or server, or to a hub through which workstations or servers connect to the network.

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

To reduce degradation, the switch 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 Address Table

The Ethernet switch network module builds the address table by using the source address of the frames received. When the switch receives a frame for a destination address not listed in its address table, it floods the frame to all interfaces of the same virtual local area network (VLAN) except the interface that received the frame. When the destination station 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 8,191 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.

Note Default parameters on the aging timer are recommended.

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 and supports only one encapsulation on all Ethernet interfaces: 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 the "Configuring Layer 2 EtherChannels (Port-Channel Logical Interfaces)" section on page 56.

Layer 2 Interface Modes

Switchport mode access puts the interface into nontrunking mode. The interface will stay in access mode regardless of what the connected port mode is. Only access VLAN traffic will travel on the access port and untagged (802.3).

Switchport mode trunk puts the interface into permanent trunking mode.

Table 1 Default Layer 2 Ethernet Interface Configuration

Feature	Default Value
Interface mode	switchport mode access / trunk
Trunk encapsulation	switchport trunk encapsulation dot1q
Allowed VLAN range	VLANs 1-1005
Default VLAN (for access ports)	VLAN 1
Native VLAN (for 802.1Q trunks)	VLAN 1
Spanning Tree Protocol (STP)	Enabled for all VLANs
STP port priority	128
STP port cost	100 for 10-Mbps Ethernet interfaces 19 for 10/100-Mbps Fast Ethernet interfaces 19 for Gigabit Ethernet interfaces operated in 100-Mb mode 4 for Gigabit Ethernet interfaces operated in 1000-Mb mode

When you connect a Cisco switch to a device other than a Cisco device through an 802.1Q trunk, the Cisco switch combines the spanning tree instance of the VLAN trunk with the spanning tree instance of the other 802.1Q switch. However, spanning tree information for each VLAN is maintained by Cisco switches separated by a cloud of 802.1Q switches that are not Cisco switches. The 802.1Q cloud separating the Cisco switches that is not Cisco devised, is treated as a single trunk link between the switches.

Make sure that 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. Inconsistencies detected by a Cisco switch mark the line as broken and block traffic for the specific VLAN.

Disabling spanning tree on the VLAN of an 802.1Q trunk without disabling spanning tree on every VLAN in the network can potentially cause spanning tree loops. Cisco recommends that you leave spanning tree enabled on the VLAN of an 802.1Q trunk or that you disable spanning tree on every VLAN in the network. Make sure that your network is loop-free before disabling spanning tree.

Layer 2 Interface Configuration Guidelines and Restrictions

Follow these guidelines and restrictions 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. 802.1Q switches that are not Cisco switches, maintain only one instance of spanning tree for all VLANs allowed on the trunks.

Switch Virtual Interfaces

A switch virtual interface (SVI) represents a VLAN of switch ports as one interface to the routing or bridging function in the system. Only one SVI can be associated with a VLAN, but it is necessary to configure an SVI for a VLAN only when you wish to route between VLANs, fallback-bridge nonroutable protocols between VLANs, or to provide IP host connectivity to the switch. By default, an SVI is created for the default VLAN (VLAN 1) to permit remote switch administration. Additional SVIs must be explicitly configured. You can configure routing across SVIs.

SVIs are created the first time that you enter the **vlan** interface configuration command for a VLAN interface. The VLAN corresponds to the VLAN tag associated with data frames on an ISL or 802.1Q encapsulated trunk or the VLAN ID configured for an access port. Configure a VLAN interface for each VLAN for which you want to route traffic, and assign it an IP address.

SVIs support routing protocol and bridging configurations. For more information about configuring IP routing, see the "Configuring IP Multicast Layer 3 Switching" section on page 98.

Routed Ports

A routed port is a physical port that acts like a port on a router; it does not have to be connected to a router. A routed port is not associated with a particular VLAN, as is an access port. A routed port behaves like a regular router interface, except that it does not support subinterfaces. Routed ports can be configured with a Layer 3 routing protocol.

Configure routed ports by putting the interface into Layer 3 mode with the **no switchport** interface configuration command. Then assign an IP address to the port, enable routing, and assign routing protocol characteristics by using the **ip routing** and **router** *protocol* global configuration commands.

Caution Entering a **no switchport** interface configuration command shuts the interface down and then reenables it, which might generate messages on the device to which the interface is connected. Furthermore, when you use this command to put the interface into Layer 3 mode, you are deleting any Layer 2 characteristics configured on the interface. (Also, when you return the interface to Layer 2 mode, you are deleting any Layer 3 characteristics configured on the interface.)

The number of routed ports and SVIs that you can configure is not limited by software; however, the interrelationship between this number and the number of other features being configured might have an impact on CPU utilization because of hardware limitations.

Routed ports support only CEF switching (IP fast switching is not supported).

VLAN Trunk Protocol

VLAN Trunk Protocol (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 switches that share the same VTP domain name and that are interconnected with trunks. VTP minimizes misconfigurations and configuration inconsistencies that can result in a number of problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations. Before you create VLANs, you must decide whether to use VTP in your network. With VTP, you can make configuration changes centrally on one or more switches and have those changes automatically communicated to all the other switches in the network.

VTP Domain

A VTP domain (also called a VLAN management domain) is made up of one or more interconnected switches that share the same VTP domain name. A switch can be configured to be in one and only one VTP domain. You make global VLAN configuration changes for the domain using either the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

By default, the switch is in VTP server mode and is in an un-named domain state until the switch receives an advertisement for a domain over a trunk link or until 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 switches in the VTP domain. VTP advertisements are transmitted out all trunk connections using IEEE 802.1Q encapsulation.

VTP maps VLANs dynamically across multiple LAN types with unique names and internal index associations. Mapping eliminates excessive device administration required from network administrators.

VTP Modes

You can configure a 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) for the entire VTP domain. VTP servers advertise their VLAN configuration to other switches in the same VTP domain and synchronize their VLAN configuration with other switches 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 switches do not participate in VTP. A VTP transparent switch does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements. However, in VTP version 2, transparent switches do forward VTP advertisements that they receive out their trunk interfaces.

VTP Advertisements

Each switch in the VTP domain sends periodic advertisements out each trunk interface to a reserved multicast address. VTP advertisements are received by neighboring switches, which update their VTP and VLAN configurations as necessary.

The following global configuration information is distributed in VTP advertisements:

- VLAN IDs (801.Q)
- VTP domain name
- VTP configuration revision number
- VLAN configuration, including maximum transmission unit (MTU) size for each VLAN
- Frame format

VTP Version 2

If you use VTP in your network, you must decide whether to use VTP version 1 or version 2. VTP version 2 supports the following features not supported in version 1:

Unrecognized Type-Length-Value (TLV) Support—A VTP server or client propagates configuration changes to its other trunks, even for TLVs it is not able to parse. The unrecognized TLV is saved in NVRAM.

Version-Dependent Transparent Mode—In VTP version 1, a VTP transparent switch inspects VTP messages for the domain name and version, and forwards a message only if the version and domain name match. Since only one domain is supported in the NM-16ESW software, VTP version 2 forwards VTP messages in transparent mode, without checking the version.

Consistency Checks—In VTP version 2, VLAN consistency checks (such as VLAN names and values) are performed only when you enter new information through the CLI or SNMP. Consistency checks are not performed when new information is obtained from a VTP message, or when information is read from NVRAM. If the digest on a received VTP message is correct, its information is accepted without consistency checks.

VTP Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when implementing VTP in your network:

- All switches in a VTP domain must run the same VTP version.
- You must configure a password on each switch in the management domain when in secure mode.
- A VTP version 2-capable switch can operate in the same VTP domain as a switch running VTP version 1, provided that VTP version 2 is disabled on the VTP version 2-capable switch. (VTP version 2 is disabled by default).
- Do not enable VTP version 2 on a switch unless all switches in the same VTP domain are version 2-capable. When you enable VTP version 2 on a switch, all version 2-capable switches in the domain enable VTP version 2
- The Cisco IOS **end** and **Ctrl**-**Z** commands are not supported in VLAN database mode.
- The VLAN database stored on internal flash is supported.
- Use the **squeeze flash** command to remove old copies of overwritten VLAN databases.

EtherChannel

EtherChannel bundles up to eight individual Ethernet links into a single logical link that provides bandwidth of up to 1600 Mbps (Fast EtherChannel full duplex) between the network module and another switch or host.

A Ethernet switch network module system supports a maximum of six EtherChannels. All interfaces in each EtherChannel must have the same speed duplex and mode.

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, or IP addresses; either source or destination or both source and destination. The selected mode applies to all EtherChannels configured on the switch.

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 may result in better load balancing.

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 2 EtherChannels:

- Assign all interfaces in the EtherChannel to the same VLAN, or configure them as trunks.

An EtherChannel supports the same allowed range of VLANs on all interfaces in a trunking Layer 2 EtherChannel. If the allowed range of VLANs is not the same, the interfaces do not form an EtherChannel.

Interfaces with different Spanning Tree Protocol (STP) port path costs can form an EtherChannel as long they are otherwise compatibly configured. Setting different STP port path costs does not, by itself, make interfaces incompatible for the formation of an EtherChannel.

After you configure an EtherChannel, configuration that you apply to the port-channel interface affects the EtherChannel.

802.1x Port-Based Authentication

This section describes how to configure IEEE 802.1x port-based authentication to prevent unauthorized devices (clients) from gaining access to the network. As LANs extend to hotels, airports, and corporate lobbies, insecure environments could be created.

Understanding 802.1x Port-Based Authentication

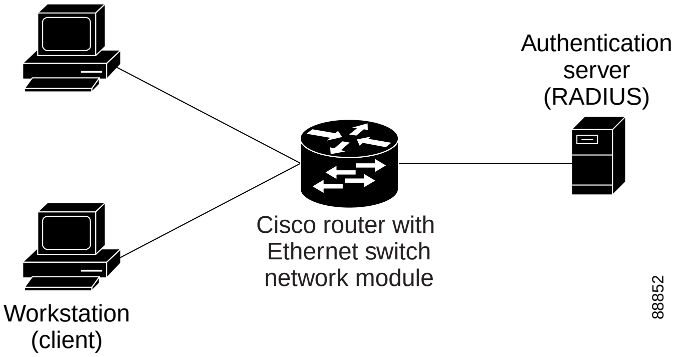
The IEEE 802.1x standard defines a client/server-based access control and authentication protocol that restricts unauthorized devices from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a switch port before making available any services offered by the switch or the LAN.

Until the client is authenticated, 802.1x access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

Device Roles

With 802.1x port-based authentication, the devices in the network have specific roles as shown in Figure 1.

Figure 1 802.1x Device Roles



- *Client*—the device (workstation) that requests access to the LAN and switch services and responds to the requests from the switch. The workstation must be running 802.1x-compliant client software such as that offered in the Microsoft Windows XP operating system. (The client is the *supplicant* in the IEEE 802.1x specification.)

Note To resolve Windows XP network connectivity and 802.1x authentication issues, read the Microsoft Knowledge Base article at this URL: http://support.microsoft.com/support/kb/articles/Q303/5/97.ASP

- *Authentication server*—performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch whether or not the client is authorized to access the LAN and switch services. Because the switch acts as the proxy, the authentication service is transparent to the client. In this release, the Remote Authentication Dial-In User Service (RADIUS) security system with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server version 3.0. RADIUS operates in a client/server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.
- *Switch* (edge switch or wireless access point)—controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client. The switch includes the RADIUS client, which is responsible for encapsulating and decapsulating the Extensible Authentication Protocol (EAP) frames and interacting with the authentication server.

  When the switch receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is reencapsulated in the RADIUS format. The EAP frames are not modified or examined during encapsulation, and the authentication server must support EAP within the native frame format. When the switch receives frames from the authentication server, the server's frame header is removed, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client.

  The devices that can act as intermediaries include the Catalyst 3550 multilayer switch, Catalyst 2950 switch, or a wireless access point. These devices must be running software that supports the RADIUS client and 802.1x.

Authentication Initiation and Message Exchange

The switch or the client can initiate authentication. If you enable authentication on a port by using the **dot1x port-control auto** interface configuration command, the switch must initiate authentication when it determines that the port link state changes from down to up. It then sends an EAP-request/identity frame to the client to request its identity (typically, the switch sends an initial identity/request frame followed by one or more requests for authentication information). Upon receipt of the frame, the client responds with an EAP-response/identity frame.

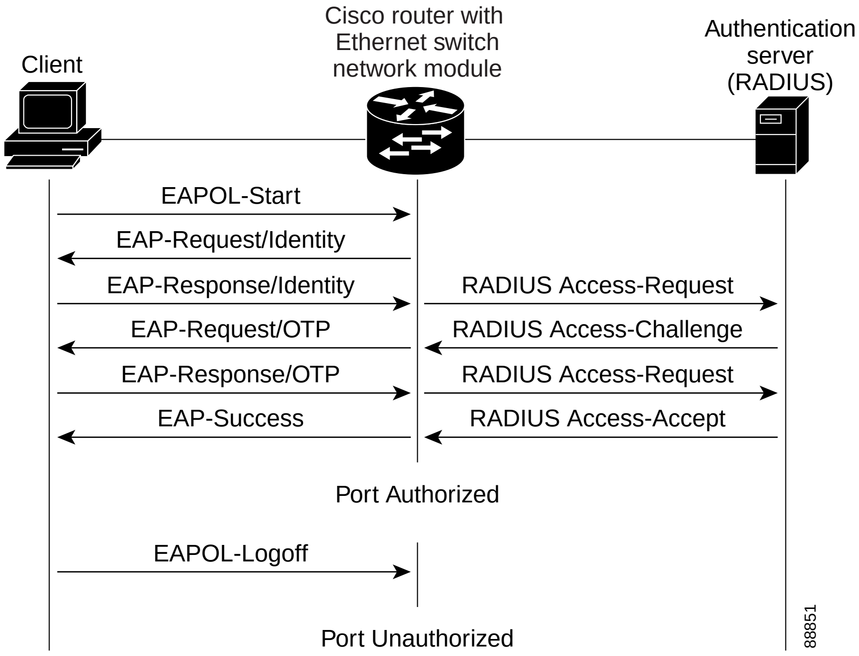
However, if during bootup, the client does not receive an EAP-request/identity frame from the switch, the client can initiate authentication by sending an EAPOL-start frame, which prompts the switch to request the client's identity.

Note If 802.1x is not enabled or supported on the network access device, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client transmits frames as if the port is in the authorized state. A port in the authorized state effectively means that the client has been successfully authenticated. For more information, see the "Ports in Authorized and Unauthorized States" section on page 11.

When the client supplies its identity, the switch begins its role as the intermediary, passing EAP frames between the client and the authentication server until authentication succeeds or fails. If the authentication succeeds, the switch port becomes authorized. For more information, see the "Ports in Authorized and Unauthorized States" section on page 11.

The specific exchange of EAP frames depends on the authentication method being used. Figure 2 shows a message exchange initiated by the client using the One-Time-Password (OTP) authentication method with a RADIUS server.

Figure 2 Message Exchange



Ports in Authorized and Unauthorized States

The switch port state determines whether or not the client is granted access to the network. The port starts in the *unauthorized* state. While in this state, the port disallows all ingress and egress traffic except for 802.1x packets. When a client is successfully authenticated, the port changes to the *authorized* state, allowing all traffic for the client to flow normally.

If a client that does not support 802.1x is connected to an unauthorized 802.1x port, the switch requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1x-enabled client connects to a port that is not running 802.1x, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

You control the port authorization state by using the **dot1x port-control** interface configuration command and these keywords:

- **force-authorized**—disables 802.1x and causes the port to change to the authorized state without any authentication exchange required. The port transmits and receives normal traffic without 802.1x-based authentication of the client. This is the default setting.
- **force-unauthorized**—causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The switch cannot provide authentication services to the client through the interface.
- **auto**—enables 802.1x and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port changes from down to up, or when an EAPOL-start frame is received. The switch requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each client attempting to access the network is uniquely identified by the switch by using the client's MAC address.

If the client is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated client are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the switch can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and network access is not granted.

When a client logs off, it sends an EAPOL-logoff message, causing the switch port to change to the unauthorized state.

If the link state of a port changes from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

Supported Topologies

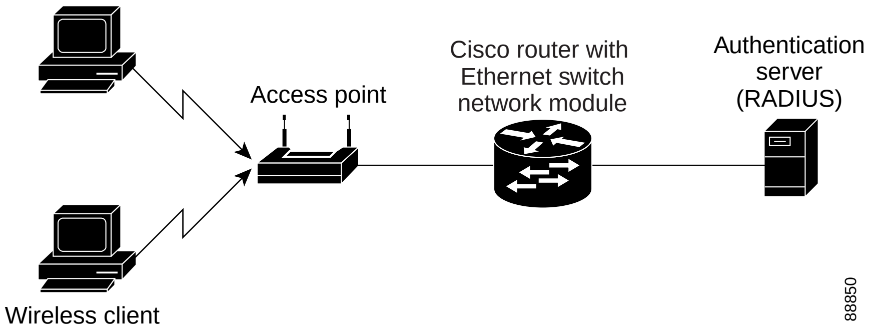
The 802.1x port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration (see Figure 1 on page 9), only one client can be connected to the 802.1x-enabled switch port. The switch detects the client when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

Figure 3 shows 802.1x-port-based authentication in a wireless LAN. The 802.1x port is configured as a multiple-host port that becomes authorized as soon as one client is authenticated. When the port is authorized, all other hosts indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the switch denies access to the network to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and the wireless access point acts as a client to the switch.

Figure 3 Wireless LAN Example



Spanning Tree Protocol

This section describes how to configure the Spanning Tree Protocol (STP) on Ethernet switch network module systems.

Spanning tree 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 to a switched LAN of multiple segments.

The Ethernet switch network module uses STP (the IEEE 802.1D bridge protocol) on all VLANs. By default, a single instance of STP runs on each configured VLAN (provided that you do not manually disable STP). You can enable and disable STP on a per-VLAN basis.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The 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 endstation MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network.

Spanning Tree Protocol 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.

Bridge Protocol Data Units

The stable active spanning tree topology of a switched network is determined by the following:

- 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

The Bridge Protocol Data Units **(**BPDU) are transmitted in one direction from the root switch, and each switch sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the 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 bridge packet data unit (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 is 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 (32768), 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 needed to reach the root switch from anywhere in the switched network are placed in spanning tree blocking mode.

BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. Spanning tree uses this information 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 2 describes the STP timers that affect the entire spanning tree performance:

Table 2 STP Timers

Timer	Purpose
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 protocol information received on a port is stored by the switch.

Spanning Tree 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 changes 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:

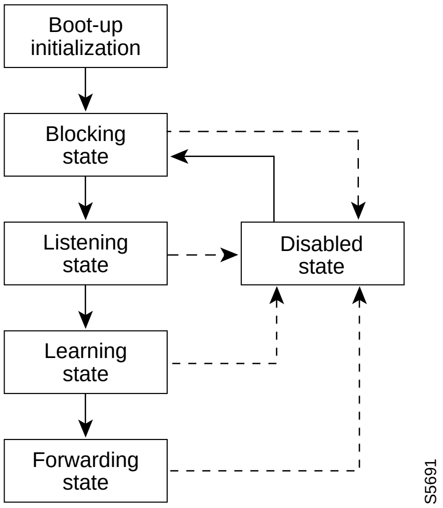
- Blocking—The Layer 2 interface does not participate in frame forwarding.
- Listening—First transitional state after the blocking state when spanning tree determines that the Layer 2 interface should participate in frame forwarding.
- Learning—The Layer 2 interface prepares to participate in frame forwarding.
- Forwarding—The Layer 2 interface forwards frames.
- Disabled—The Layer 2 interface does not participate in spanning tree and is not forwarding 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 4 illustrates how a port moves through the five stages.

Figure 4 STP Port 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.

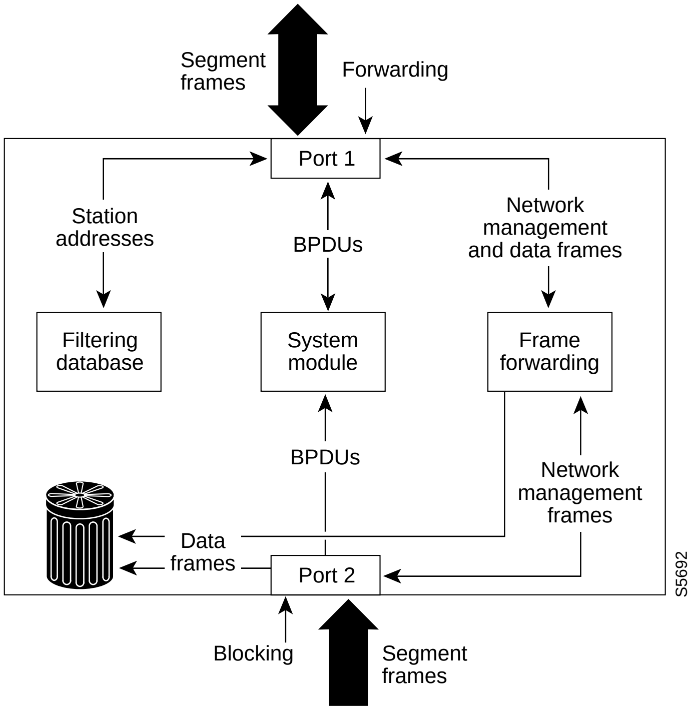
When the spanning tree algorithm places a Layer 2 interface in the forwarding state, the following process occurs:

1. The Layer 2 interface is put into the listening state while it waits for protocol information that suggests that 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 5. 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 5 Interface 2 in Blocking State



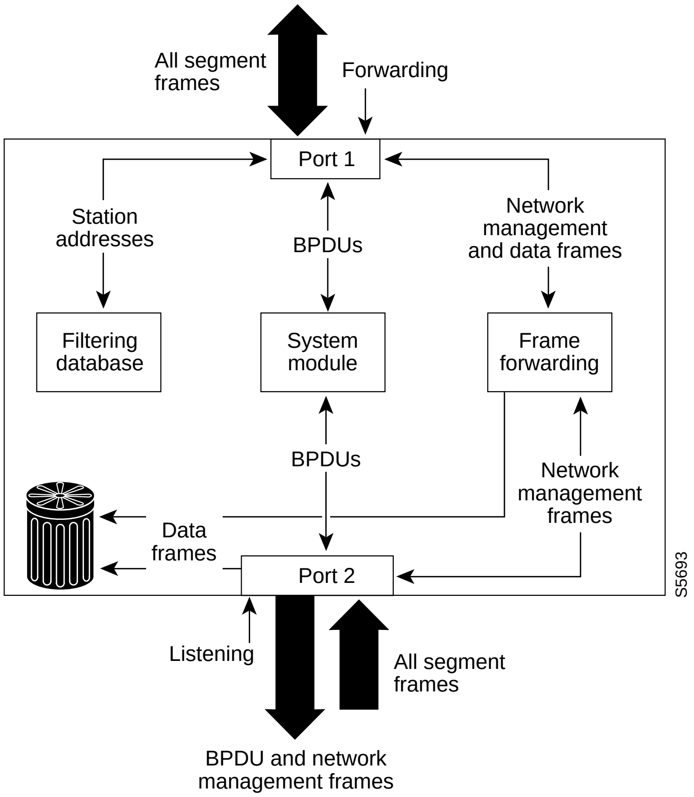
A Layer 2 interface in the blocking state performs as follows:

- 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 a Layer 2 interface enters after the blocking state. The Layer 2 interface enters this state when STP determines that the Layer 2 interface should participate in frame forwarding. Figure 6 shows a Layer 2 interface in the listening state.

Figure 6 Interface 2 in Listening State



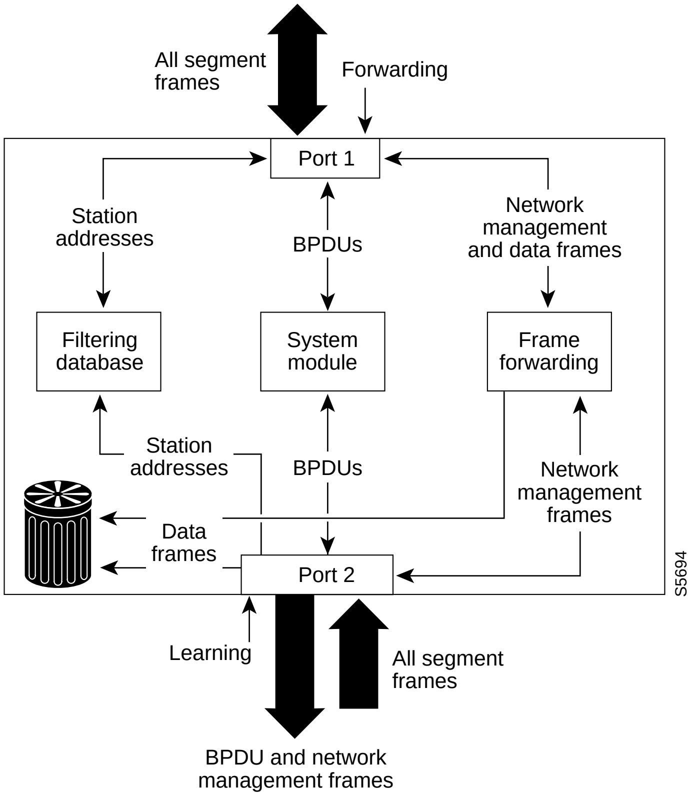
A Layer 2 interface in the listening state performs as follows:

- 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 7 shows a Layer 2 interface in the learning state.

Figure 7 Interface 2 in Learning State



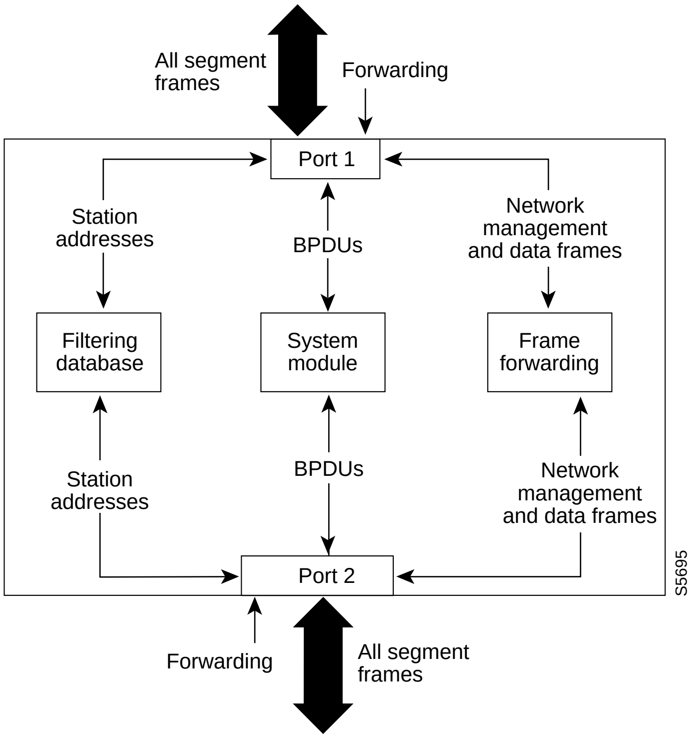
A Layer 2 interface in the learning state performs as follows:

- 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 8. The Layer 2 interface enters the forwarding state from the learning state.

Figure 8 Interface 2 in Forwarding State



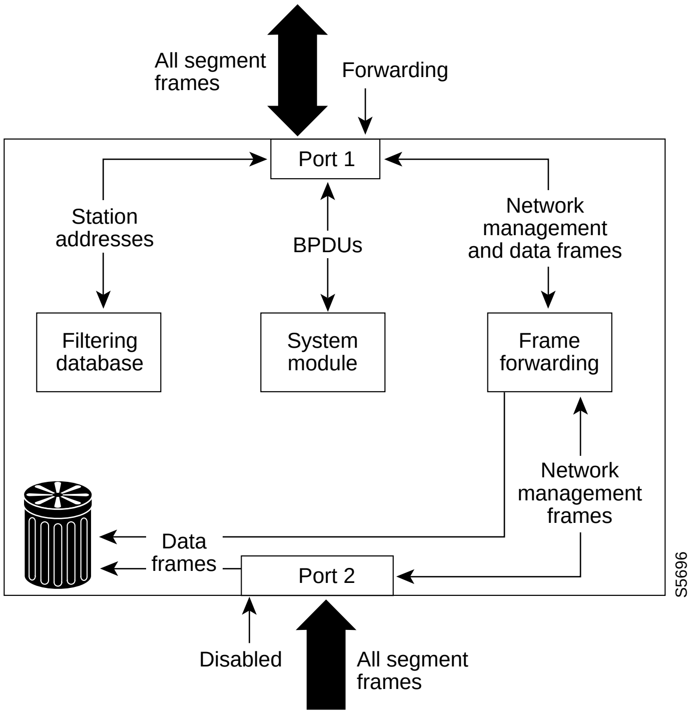
A Layer 2 interface in the forwarding state performs as follows:

- 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 9. A Layer 2 interface in the disabled state is virtually nonoperational.

Figure 9 Interface 2 in Disabled State



A disabled Layer 2 interface performs 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 MAC address allocation manager has a pool of MAC addresses that are used as the bridge IDs for the VLAN spanning trees. In Table 3 you can view the number of VLANs allowed for each platform.

Table 3 Number of VLANs Allowed by Platform

Platform	Maximum number of VLANs allowed
Cisco 3640 or higher	64 VLANS
Cisco 3620	32 VLANs
Cisco 2600	32 VLANs

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 forth.

Default Spanning Tree Configuration

In Table 4 you can view the default Spanning Tree configuration values.

Table 4 Spanning Tree Default Configuration

Feature	Default Value
Enable state	Spanning tree enabled for all VLANs
Bridge priority	32768
Spanning tree port priority (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)	Fast Ethernet: 19 Ethernet: 100 Gigabit Ethernet: 19 when operated in 100-Mb mode, and 4 when operated in 1000-Mb mode
Spanning tree VLAN port priority (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)	Fast Ethernet: 10 Ethernet: 10
Hello time	2 seconds
Forward delay time	15 seconds
Maximum aging time	20 seconds

Spanning Tree 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 to 255, configurable in increments of 4 (the default is 128).

Cisco IOS software uses the port priority value when the interface is configured as an access port and uses VLAN port priority values when the interface is configured as a trunk port.

Spanning Tree Port Cost

The spanning tree port path cost default value is derived from the media speed of an interface. 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 0 to 65535 (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.

BackboneFast

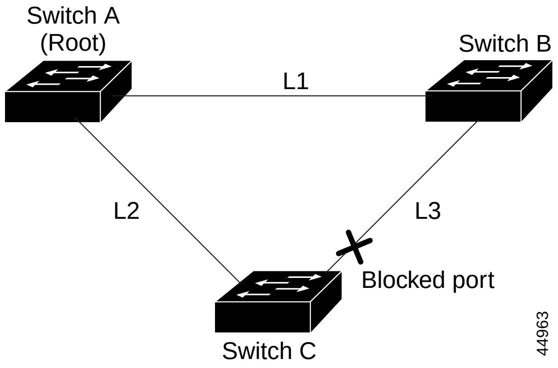
BackboneFast is initiated when a root port or blocked port on a switch receives inferior BPDUs from its designated bridge. An inferior BPDU identifies one switch as both the root bridge and the designated bridge. When a switch receives an inferior BPDU, it means that a link to which the switch is not directly connected (an *indirect* link) has failed (that is, the designated bridge has lost its connection to the root switch). Under STP rules, the switch ignores inferior BPDUs for the configured maximum aging time specified by the **spanning-tree max-age** global configuration command.

The switch tries to determine if it has an alternate path to the root switch. If the inferior BPDU arrives on a blocked port, the root port and other blocked ports on the switch become alternate paths to the root switch. (Self-looped ports are not considered alternate paths to the root switch.) If the inferior BPDU arrives on the root port, all blocked ports become alternate paths to the root switch. If the inferior BPDU arrives on the root port and there are no blocked ports, the switch assumes that it has lost connectivity to the root switch, causes the maximum aging time on the root to expire, and becomes the root switch according to normal STP rules.

If the switch has alternate paths to the root switch, it uses these alternate paths to transmit a new kind of Protocol Data Unit (PDU) called the Root Link Query PDU. The switch sends the Root Link Query PDU on all alternate paths to the root switch. If the switch determines that it still has an alternate path to the root, it causes the maximum aging time on the ports on which it received the inferior BPDU to expire. If all the alternate paths to the root switch indicate that the switch has lost connectivity to the root switch, the switch causes the maximum aging times on the ports on which it received an inferior BPDU to expire. If one or more alternate paths can still connect to the root switch, the switch makes all ports on which it received an inferior BPDU its designated ports and moves them out of the blocking state (if they were in the blocking state), through the listening and learning states, and into the forwarding state.

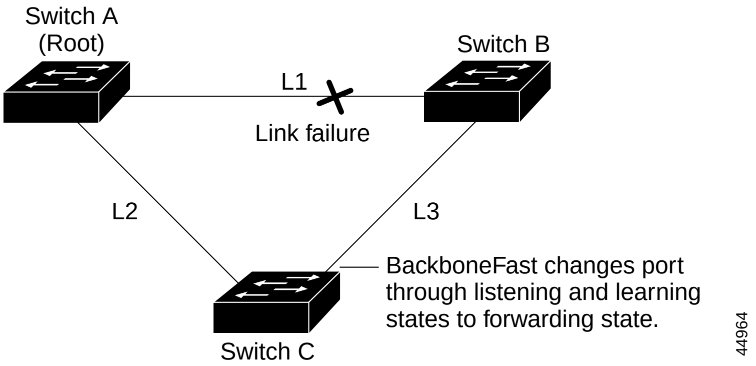
Figure 10 shows an example topology with no link failures. Switch A, the root switch, connects directly to Switch B over link L1 and to Switch C over link L2. The interface on Switch C that connects directly to Switch B is in the blocking state.

Figure 10 BackboneFast Example Before Indirect Link Failure



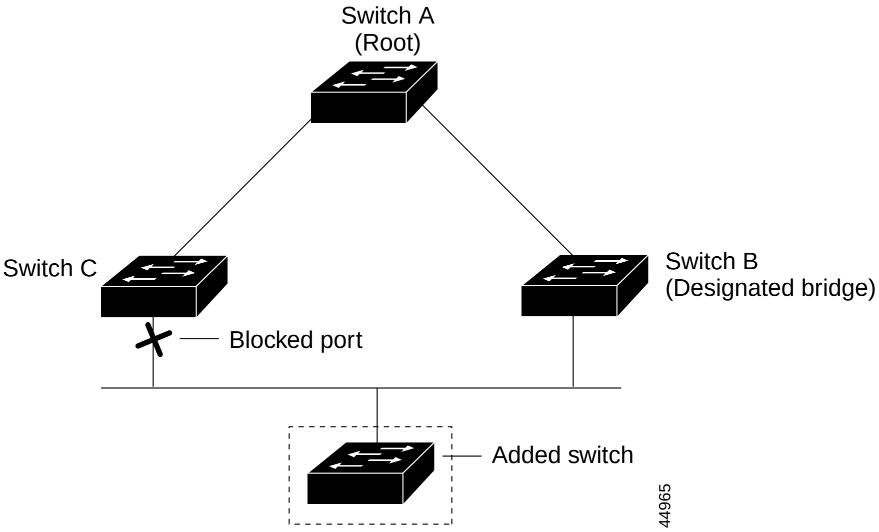
If link L1 fails, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root switch over L1, it detects the failure, elects itself the root, and begins sending BPDUs to Switch C, identifying itself as the root. When Switch C receives the inferior BPDUs from Switch B, Switch C assumes that an indirect failure has occurred. At that point, BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the maximum aging time for the port to expire. BackboneFast then changes the 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 11 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 11 BackboneFast Example After Indirect Link Failure


Switch A
(Root)
Switch B
L1
Link failure
L2
L3
BackboneFast changes port through listening and learning states to forwarding state.
Switch C
44964

If a new switch is introduced into a shared-medium topology as shown in Figure 12, 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 12 Adding a Switch in a Shared-Medium Topology



Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is a protocol that runs over Layer 2 (the data link layer) on all Cisco routers, bridges, access servers, and switches. CDP allows network management applications to discover Cisco devices that are neighbors of already known devices, in particular, neighbors running lower-layer, transparent protocols. With CDP, network management applications can learn the device type and the SNMP agent address of neighboring devices. This feature enables applications to send SNMP queries to neighboring devices.

CDP runs on all LAN and WAN media that support Subnetwork Access Protocol (SNAP). Each CDP-configured device sends periodic messages to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain the time-to-live, or hold-time information, which indicates the length of time a receiving device should hold CDP information before discarding it.

Switched Port Analyzer

Switched Port Analyzer Session

A Switched Port Analyzer (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 and to send either ingress traffic, egress traffic, or both to one destination interface. You can configure one SPAN session with separate or overlapping sets of SPAN source interfaces or VLANs. Only switched interfaces can be configured as SPAN sources or destinations on the same network module.

SPAN sessions do not interfere with the normal operation of the switch. You can enable or disable SPAN sessions with command-line interface (CLI) or SNMP commands. When enabled, a SPAN session might become active or inactive based on various events or actions, and this would be indicated by a syslog message. The **show monitor session SPAN session number** command displays the operational status of a SPAN session.

A SPAN session remains inactive after system power-up until the destination interface is operational.

Destination Interface

A destination interface (also called a monitor interface) is a switched interface to which SPAN sends packets for analysis. You can have one SPAN destination interface. 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 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.

You can configure source interfaces in any VLAN. You can configure EtherChannel 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 mixed with nontrunk source interfaces; however, the destination interface never encapsulates.

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.

SPAN Traffic

Network traffic, including multicast, can be monitored using SPAN. Multicast packet monitoring is enabled by default. 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).

Note Monitoring of VLANs is not supported.

SPAN Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring SPAN:

- Enter the **no monitor session** *session number* command with no other parameters to clear the SPAN session number.
- EtherChannel interfaces can be SPAN source interfaces; they cannot be SPAN destination interfaces.
- If you specify multiple SPAN source interfaces, the interfaces can belong to different VLANs.
- Monitoring of VLANs is not supported
- Only one SPAN session may be run at any given time.
- Outgoing CDP and BPDU packets will not be replicated.
- 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.
- Use a network analyzer to monitor interfaces.
- You can have one SPAN destination interface.
- You can mix individual source interfaces within a single SPAN session.
- You cannot configure a SPAN destination interface to receive ingress traffic.
- When enabled, SPAN uses any previously entered configuration.
- When you specify source interfaces and do not specify a traffic type (**Tx**, **Rx**, or **both**), **both** is used by default.

Network Security with ACLs

Network security on your Ethernet switch network module can be implemented using access control lists (ACLs), which are also referred to in commands and tables as access lists.

Understanding ACLs

Packet filtering can limit network traffic and restrict network use by certain users or devices. ACLs can filter traffic as it passes through a switch and permit or deny packets from crossing specified interfaces. An ACL is a sequential collection of permit and deny conditions that apply to packets. When a packet is received on an interface, the switch compares the fields in the packet against any applied ACLs to verify that the packet has the required permissions to be forwarded, based on the criteria specified in the access lists. The switch tests the packet against the conditions in an access list one by one. The first match determines whether the switch accepts or rejects the packet. Because the switch stops testing conditions after the first match, the order of conditions in the list is critical. If no conditions match, the switch rejects the packet. If there are no restrictions, the switch forwards the packet; otherwise, the switch drops the packet.

You configure access lists on a Layer 2 switch to provide basic security for your network. If you do not configure ACLs, all packets passing through the switch could be allowed onto all parts of the network. You can use ACLs to control which hosts can access different parts of a network or to decide which types of traffic are forwarded or blocked at switch interfaces. For example, you can allow e-mail traffic to be forwarded but not Telnet traffic. ACLs can be configured to block inbound traffic.

An ACL contains an ordered list of access control entries (ACEs). Each ACE specifies *permit* or *deny* and a set of conditions the packet must satisfy in order to match the ACE. The meaning of *permit* or *deny* depends on the context in which the ACL is used.

The Ethernet switch network module supports IP ACLs to filter IP traffic, including TCP or User Datagram Protocol (UDP) traffic (but not both traffic types in the same ACL).

ACLs

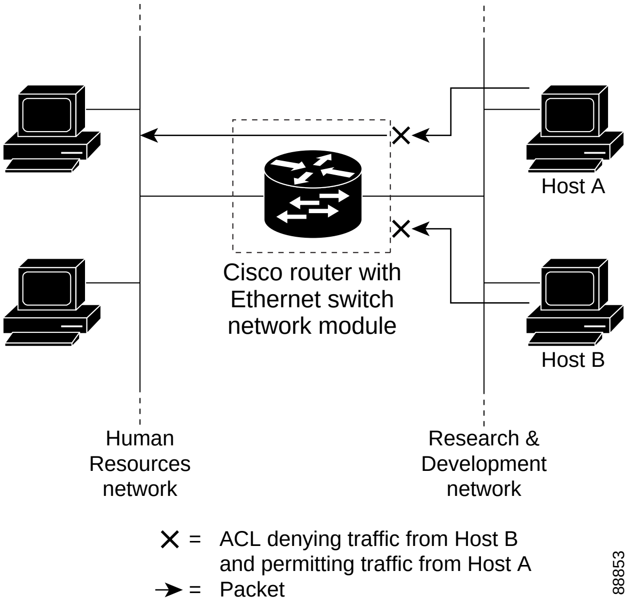
You can apply ACLs on physical Layer 2 interfaces. ACLs are applied on interfaces only on the inbound direction.

- Standard IP access lists use source addresses for matching operations.
- Extended IP access lists use source and destination addresses and optional protocol type information for matching operations.

The switch examines access lists associated with features configured on a given interface and a direction. As packets enter the switch on an interface, ACLs associated with all inbound features configured on that interface are examined.

ACLs permit or deny packet forwarding based on how the packet matches the entries in the ACL. For example, you can use ACLs to allow one host to access a part of a network, but to prevent another host from accessing the same part. In Figure 13, ACLs applied at the switch input allow Host A to access the Human Resources network, but prevent Host B from accessing the same network.

Figure 13 Using ACLs to Control Traffic to a Network


Host A
Cisco router with Ethernet switch network module
Host B
Human Resources network
Research & Development network
✕ = ACL denying traffic from Host B and permitting traffic from Host A
➔ = Packet
88853

Handling Fragmented and Unfragmented Traffic

IP packets can be fragmented as they cross the network. When this happens, only the fragment containing the beginning of the packet contains the Layer 4 information, such as TCP or UDP port numbers, ICMP type and code, and so on. All other fragments are missing this information.

Some ACEs do not check Layer 4 information and therefore can be applied to all packet fragments. ACEs that do test Layer 4 information cannot be applied in the standard manner to most of the fragments in a fragmented IP packet. When the fragment contains no Layer 4 information and the ACE tests some Layer 4 information, the matching rules are modified:

- Permit ACEs that check the Layer 3 information in the fragment (including protocol type, such as TCP, UDP, and so on) are considered to match the fragment regardless of what the missing Layer 4 information might have been.
- Deny ACEs that check Layer 4 information never match a fragment unless the fragment contains Layer 4 information.

Consider access list 102, configured with these commands, applied to three fragmented packets:

```
Switch (config)# access-list 102 permit tcp any host 10.1.1.1 eq smtp
Switch (config)# access-list 102 deny tcp any host 10.1.1.2 eq telnet
Switch (config)# access-list 102 deny tcp any any
```

Note In the first and second ACEs in the examples, the **eq** keyword after the destination address means to test for the TCP-destination-port well-known numbers equaling Simple Mail Transfer Protocol (SMTP) and Telnet, respectively.

- Packet A is a TCP packet from host 10.2.2.2, port 65000, going to host 10.1.1.1 on the SMTP port. If this packet is fragmented, the first fragment matches the first ACE (a permit), as if it were a complete packet because all Layer 4 information is present. The remaining fragments also match the first ACE, even though they do not contain the SMTP port information because the first ACE only checks Layer 3 information when applied to fragments. (The information in this example is that the packet is TCP and that the destination is 10.1.1.1.)

- Packet B is from host 10.2.2.2, port 65001, going to host 10.1.1.2 on the Telnet port. If this packet is fragmented, the first fragment matches the second ACE (a deny) because all Layer 3 and Layer 4 information is present. The remaining fragments in the packet do not match the second ACE because they are missing Layer 4 information.
- Because the first fragment was denied, host 10.1.1.2 cannot reassemble a complete packet, so packet B is effectively denied. However, the later fragments that are permitted will consume bandwidth on the network and resources of host 10.1.1.2 as it tries to reassemble the packet.
- Fragmented packet C is from host 10.2.2.2, port 65001, going to host 10.1.1.3, port FTP. If this packet is fragmented, the first fragment matches the third ACE (a deny). All other fragments also match the third ACE because that ACE does not check any Layer 4 information and because Layer 3 information in all fragments shows that they are being sent to host 10.1.1.3, and the earlier permit ACEs were checking different hosts.

Understanding Access Control Parameters

Before configuring ACLs on the Ethernet switch network module, you must have a thorough understanding of the Access Control Parameters (ACPs). ACPs are referred to as masks in the switch CLI commands, and output.

Each ACE has a mask and a rule. The Classification Field or mask is the field of interest on which you want to perform an action. The specific values associated with a given mask are called *rules*.

Packets can be classified on these Layer 3 and Layer 4 fields.

- Layer 3 fields:
  - IP source address (Specify all 32 IP source address bits to define the flow, or specify a user-defined subnet. There are no restrictions on the IP subnet to be specified.)
  - IP destination address (Specify all 32 IP destination address bits to define the flow, or specify a user-defined subnet. There are no restrictions on the IP subnet to be specified.)

    You can use any combination or all of these fields simultaneously to define a flow.
- Layer 4 fields:
  - TCP (You can specify a TCP source, destination port number, or both at the same time.)
  - UDP (You can specify a UDP source, destination port number, or both at the same time.)

Note A mask can be a combination of multiple Layer 3 and Layer 4 fields.

There are two types of masks:

- User-defined mask—masks that are defined by the user.
- System-defined mask—these masks can be configured on any interface:

```
Switch (config-ext-nacl)# permit tcp any any
Switch (config-ext-nacl)# deny tcp any any
Switch (config-ext-nacl)# permit udp any any
Switch (config-ext-nacl)# deny udp any any
Switch (config-ext-nacl)# permit ip any any
Switch (config-ext-nacl)# deny ip any any
Switch (config-ext-nacl)# deny any any
Switch (config-ext-nacl)# permit any any
```

Note In an IP extended ACL (both named and numbered), a Layer 4 system-defined mask cannot precede a Layer 3 user-defined mask. For example, a Layer 4 system-defined mask such as **permit tcp any any** or **deny udp any any** cannot precede a Layer 3 user-defined mask such as **permit ip 10.1.1.1 any**. If you configure this combination, the ACL is not configured. All other combinations of system-defined and user-defined masks are allowed in security ACLs.

The Ethernet switch network module ACL configuration is consistent with Cisco Catalyst switches. However, there are significant restrictions as well as differences for ACL configurations on the Ethernet switch network module.

Guidelines for Configuring ACLs on the Ethernet Switch Network Module

These configuration guidelines apply to ACL filters:

- Only one ACL can be attached to an interface. For more information, refer to the **ip access-group** interface command.
- All ACEs in an ACL must have the same user-defined mask. However, ACEs can have different rules that use the same mask. On a given interface, only one type of user-defined mask is allowed, but you can apply any number of system-defined masks. For more information on system-defined masks, see the "Understanding Access Control Parameters" section on page 28.

  The following example shows the same mask in an ACL:

  ```
  Switch (config)#ip access-list extended acl2
  Switch (config-ext-nacl)# permit tcp 10.1.1.1 0.0.0.0 any eq 80
  Switch (config-ext-nacl)# permit tcp 20.1.1.1 0.0.0.0 any eq 23
  ```

  In this example, the first ACE permits all the TCP packets coming from the host 10.1.1.1 with a destination TCP port number of 80. The second ACE permits all TCP packets coming from the host 20.1.1.1 with a destination TCP port number of 23. Both the ACEs use the same mask; therefore, a Ethernet switch network module supports this ACL.
- Only four user-defined masks can be defined for the entire system. These can be used for either security or quality of service (QoS) but cannot be shared by QoS and security. You can configure as many ACLs as you require. However, a system error message appears if ACLs with more than four different masks are applied to interfaces.

Table 5 lists a summary of the ACL restrictions on Ethernet switch network modules.

Table 5 Summary of ACL Restrictions

Restriction	Number Permitted
Number of user-defined masks allowed in an ACL	1
Number of ACLs allowed on an interface	1
Total number of user-defined masks for security and QoS allowed on a switch	4

Quality of Service

Quality of service (QoS) can be implemented on your Ethernet switch network module. With this feature, you can provide preferential treatment to certain types of traffic. Without QoS, the switch offers best-effort service to each packet, regardless of the packet contents or size. It transmits the packets without any assurance of reliability, delay bounds, or throughput.

Understanding Quality of Service (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.

With the QoS feature configured on your switch, you can select specific network traffic, prioritize it according to its relative importance, and use congestion-management and congestion-avoidance techniques to provide preferential treatment. Implementing QoS in your network makes 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 14:

- Prioritization values in Layer 2 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.

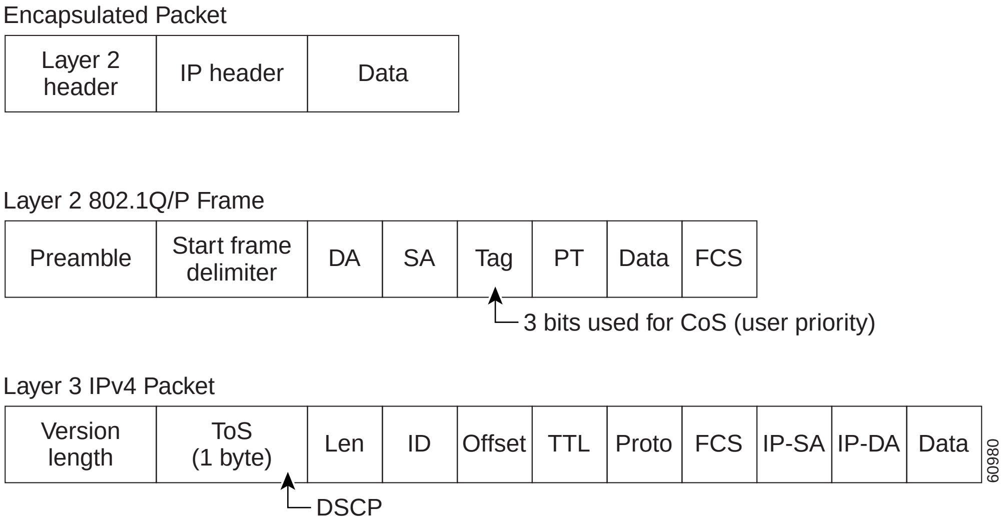
  Other frame types cannot carry Layer 2 CoS values.

  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 a Differentiated Services Code Point (DSCP) value. The supported DSCP values are 0, 8, 10, 16, 18, 24, 26, 32, 34, 40, 46, 48, and 56.

Figure 14 QoS Classification Layers in Frames and Packets



Note Layer 2 ISL Frame is not supported in this release.

Note Layer 3 IPv6 packets are dropped when received by the switch.

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.

The Ethernet switch network module can function as a Layer 2 switch connected to a Layer 3 router. When a packet enters the Layer 2 engine directly from a switch port, it is placed into one of four queues in the dynamic, 32-MB shared memory buffer. The queue assignment is based on the dot1p value in the packet. Any voice bearer packets that come in from the Cisco IP phones on the voice VLAN are automatically placed in the highest priority (Queue 3) based on the 802.1p value generated by the IP phone. The queues are then serviced on a weighted round robin (WRR) basis. The control traffic, which uses a CoS or ToS of 3, is placed in Queue 2.

Table 6 summarizes the queues, CoS values, and weights for Layer 2 QoS on the Ethernet switch network module.

Table 6 Queues, CoS values, and Weights for Layer 2 QoS

Queue Number	CoS Value	Weight
3	5,6,7	255
2	3,4	64
1	2	16
0	0,1	1

The weights specify the number of packets that are serviced in the queue before moving on to the next queue. Voice Realtime Transport Protocol (RTP) bearer traffic marked with a CoS or ToS of 5 and Voice Control plane traffic marked with a CoS/ToS of 3 are placed into the highest priority queues. If the queue has no packets to be serviced, it is skipped. Weighted Random Early Detection (WRED) is not supported on the Fast Ethernet ports.

You cannot configure port-based QoS on the Layer 2 switch ports.

Basic QoS Model

Figure 15 shows the basic QoS model. Actions at the ingress interface include classifying traffic, policing, and marking:

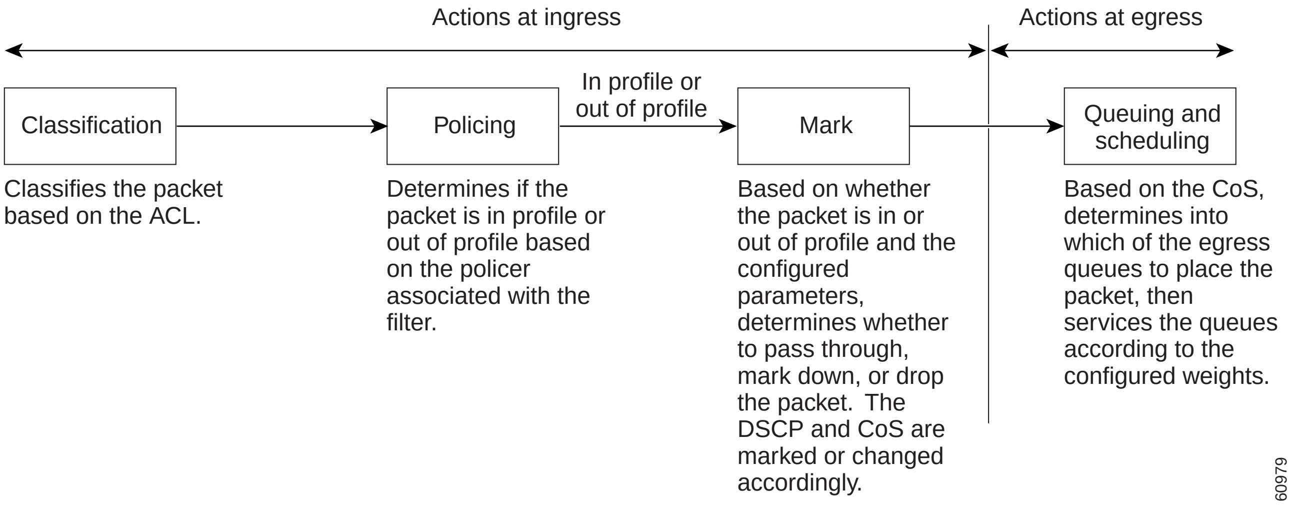
- Classifying distinguishes one kind of traffic from another. For more information, see the "Classification" section on page 32.

- Policing determines whether a packet is in or out of profile according to the configured policer, and the policer 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 34.
- Marking evaluates the policer and configuration information for 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 34.

Actions at the egress interface include queueing and scheduling:

- Queueing evaluates the CoS value and determines which of the four egress queues in which to place the packet.
- Scheduling services the four egress queues based on their configured WRR weights.

Figure 15 Basic QoS Model


Actions at ingress
Actions at egress
Classification
Policing
In profile or out of profile
Mark
Queuing and scheduling
Classifies the packet based on the ACL.
Determines if the packet is in profile or out of profile based on the policer associated with the filter.
Based on whether the packet is in or out of profile and the configured parameters, determines whether to pass through, mark down, or drop the packet. The DSCP and CoS are marked or changed accordingly.
Based on the CoS, determines into which of the egress queues to place the packet, then services the queues according to the configured weights.
60979

Classification

Classification is the process of distinguishing one kind of traffic from another by examining the fields in the packet.

Classification occurs only on a physical interface basis. No support exists for classifying packets at the VLAN or the switched virtual interface level.

You specify which fields in the frame or packet that you want to use to classify incoming traffic.

Classification Based on QoS ACLs

You can use IP standard or IP extended ACLs to define a group of packets with the same characteristics (*class*). In the QoS context, the permit and deny actions in the access control entries (ACEs) have different meanings than with security ACLs:

- If a match with a permit action is encountered (first-match principle), the specified QoS-related action is taken.
- If no match with a permit action is encountered and all the ACEs have been examined, no QoS processing occurs on the packet.
- If multiple ACLs are configured on an interface, the packet matches the first ACL with a permit action, and QoS processing begins.

- Configuration of a deny action is not supported in QoS ACLs on the 16- and 36-port Ethernet switch network modules.
- System-defined masks are allowed in class maps with these restrictions:
  - A combination of system-defined and user-defined masks cannot be used in the multiple class maps that are a part of a policy map.
  - System-defined masks that are a part of a policy map must all use the same type of system mask. For example, a policy map cannot have a class map that uses the **permit tcp any any** ACE and another that uses the **permit ip any any** ACE.
  - A policy map can contain multiple class maps that all use the same user-defined mask or the same system-defined mask.

Note For more information on the system-defined mask, see the "Understanding Access Control Parameters" section on page 28.

- For more information on ACL restrictions, see the "Guidelines for Configuring ACLs on the Ethernet Switch Network Module" section on page 29.

After a traffic class has been defined with the ACL, you can attach a policy to it. A policy might contain multiple classes with actions specified for each one of them. A policy might include commands to 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.

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 criteria used to match against a specific traffic flow to further classify it; the criteria can include matching the access group defined by the ACL. 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 further classify it through the use of a policy map.

A policy map specifies which traffic class to act on. Actions can include setting a specific DSCP 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 or the **class** policy-map configuration command. You should use the **class-map** global configuration command when the map is shared among many ports. When you enter the **class-map** global configuration command, the switch enters the class-map configuration mode. In this mode, you define the match criterion 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 **class** policy-map configuration command and the **police** policy-map class configuration command. 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 34.

A policy map also has these characteristics:

- A policy map can contain multiple class statements.
- A separate policy-map class can exist for each type of traffic received through an interface.
- A policy-map configuration state supersedes any actions due to an interface trust state.

For configuration information, see the "Configuring a QoS Policy" section on page 90.

Policing and Marking

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 dropping the packet, or marking down the packet with a new value that is user-defined.

You can create this type of policer:

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 policy-map configuration command.

For non-IP traffic, you have these marking options:

- Use the port default. If the frame does not contain a CoS value, assign the default port CoS value to the incoming frame.
- Trust the CoS value in the incoming frame (configure the port to trust CoS). 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.

  The trust DSCP configuration is meaningless for non-IP traffic. If you configure a port with this option and non-IP traffic is received, the switch assigns the default port CoS value and classifies traffic based on the CoS value.

For IP traffic, you have these 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. The supported DSCP values are 0, 8, 10, 16, 18, 24, 26, 32, 34, 40, 46, 48, and 56.
- Trust the CoS value (if present) in the incoming packet, and generate the DSCP by using the CoS-to-DSCP map.

When configuring policing and policers, keep these items in mind:

- By default, no policers are configured.
- Policers can only be configured on a physical port. There is no support for policing at a VLAN or switched virtual interface (SVI) level.
- Only one policer can be applied to a packet in the input direction.
- Only the average rate and committed burst parameters are configurable.
- Policing occurs on the ingress interfaces:
  - 60 policers are supported on ingress Gigabit-capable Ethernet ports.
  - 6 policers are supported on ingress 10/100 Ethernet ports.
  - Granularity for the average burst rate is 1 Mbps for 10/100 ports and 8 Mbps for Gigabit Ethernet ports.

- On an interface configured for QoS, all traffic received through the interface is classified, policed, and marked according to the policy map attached to the interface. On a trunk interface configured for QoS, traffic in *all* VLANs received through the interface is classified, policed, and marked according to the policy map attached to the interface.
- VLAN-based egress DSCP-to-COS mapping is supported. DSCP-to-COS mapping occurs for all packets with a specific VLAN ID egressing from the CPU to the physical port. The packets can be placed in the physical port egress queue depending on the COS value. Packets are handled according to type of service.

Note No policers can be configured on the egress interface on Ethernet switch network modules.

Mapping Tables

The Ethernet switch network modules support these types of marking to apply to the switch:

- CoS value to the DSCP value
- DSCP value to CoS value

Note An interface can be configured to trust either CoS or DSCP, but not both at the same time.

Before the traffic reaches the scheduling stage, QoS uses the configurable DSCP-to-CoS map to derive a CoS value from the internal DSCP value.

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 CoS Maps" section on page 96.

Maximum Number of VLAN and Multicast Groups

The maximum number is less than or equal to 242. The number of VLANs is determined by multiplying the number of VLANs by the number of multicast groups. For example, the maximum number for 10 VLANs and 20 groups would be 200, under the 242 limit.

IP Multicast Support

The maximum number of multicast groups is related to the maximum number of VLANs. The product of the number of multicast groups and the number of VLANs cannot exceed 242. This feature also provides support for Protocol Independent Multicast (PIM) sparse mode/dense mode sparse-dense mode.

IGMP Snooping

Understanding IGMP Snooping

Internet Group Management Protocol (IGMP) snooping constrains the flooding of multicast traffic by dynamically configuring the interfaces so that multicast traffic is forwarded only to those interfaces associated with IP multicast devices. The LAN switch snoops on the IGMP traffic between the host and the router and keeps track of multicast groups and member ports. When the switch receives an IGMP join report from a host for a particular multicast group, the switch adds the host port number to the associated multicast forwarding table entry. When it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. After it relays the IGMP queries from the multicast router, it deletes entries periodically if it does not receive any IGMP membership reports from the multicast clients.

When IGMP snooping is enabled, the multicast router sends out periodic IGMP general queries to all VLANs. The switch responds to the router queries with only one join request per MAC multicast group, and the switch creates one entry per VLAN in the Layer 2 forwarding table for each MAC group from which it receives an IGMP join request. All hosts interested in this multicast traffic send 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 MAC multicast groups by using the **ip igmp snooping vlan static** command. If you specify group membership for a multicast group address statically, your setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can consist of both user-defined and IGMP snooping-learned settings.

Ethernet switch network modules support a maximum of 255 IP multicast groups and support both IGMP version 1 and IGMP version 2.

If a port spanning-tree, a port group, or a VLAN ID change occurs, the IGMP snooping-learned multicast groups from this port on the VLAN are deleted.

In the IP multicast-source-only environment, the switch learns the IP multicast group from the IP multicast data stream and only forwards traffic to the multicast router ports.

Immediate-Leave Processing

IGMP snooping Immediate-Leave processing allows the switch to remove an interface that sends a leave message from the forwarding table 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. Immediate-Leave processing ensures optimal bandwidth management for all hosts on a switched network, even when multiple multicast groups are in use simultaneously.

Note You should use the Immediate-Leave processing feature only on VLANs where only one host is connected to each port. If Immediate-Leave is enabled on VLANs where more than one host is connected to a port, some hosts might be inadvertently dropped. Immediate Leave is supported only with IGMP version 2 hosts.

Setting the Snooping Method

Multicast-capable router ports are added to the forwarding table for every IP multicast entry. The switch learns of such ports through one of these methods:

- Snooping on PIM and DVMRP packets
- Statically connecting to a multicast router port with the **ip igmp snooping mrouter** global configuration command

You can configure the switch to snoop on PIM/Distance Vector Multicast Routing Protocol (PIM/DVMRP) packets. By default, the switch snoops on PIM/DVMRP packets on all VLANs. To learn of multicast router ports through PIM-DVMRP packets, use the **ip igmp snooping vlan** *vlan-id* **mrouter learn pim-dvmrp** interface command.

Joining a Multicast Group

When a host connected to the switch wants to join an IP multicast group, it sends an IGMP join message, specifying the IP multicast group it wants to join. When the switch receives this message, it adds the port to the IP multicast group port address entry in the forwarding table.

Refer to Figure 16. Host 1 wants to join multicast group 224.1.2.3 and multicasts an unsolicited IGMP membership report (IGMP join message) to the group with the equivalent MAC destination address of 0100.5E01.0203. The switch recognizes IGMP packets and forwards them to the CPU. When the CPU receives the IGMP report multicast by Host 1, the CPU uses the information to set up a multicast forwarding table entry as shown in Table 7 that includes the port numbers of Host 1 and the router.

Figure 16 Initial IGMP Join Message

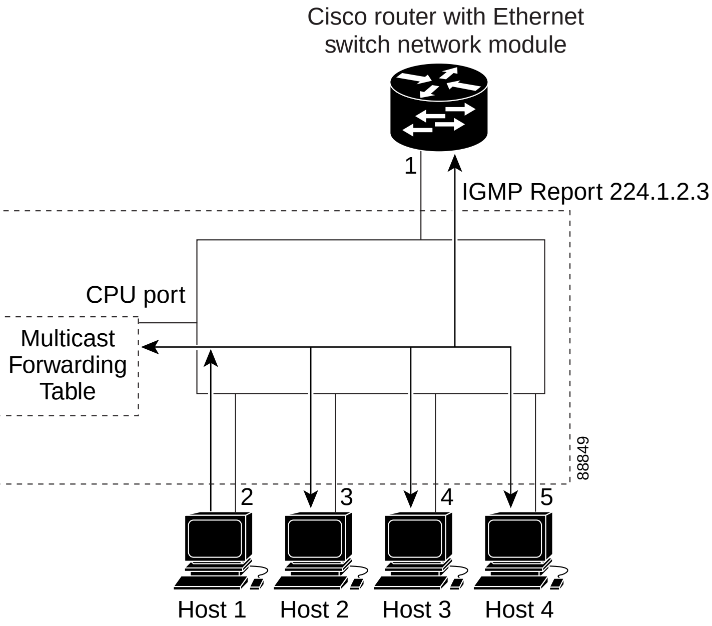


Table 7 IP Multicast Forwarding Table

Destination Address	Type of Packet	Ports
0100.5e01.0203	!IGMP	1, 2

Note that the switch architecture allows the CPU to distinguish IGMP information packets from other packets for the multicast group. The switch recognizes the IGMP packets through its filter engine. This prevents the CPU from becoming overloaded with multicast frames.

The entry in the multicast forwarding table tells the switching engine to send frames addressed to the 0100.5E01.0203 multicast MAC address that are not IGMP packets (!IGMP) to the router and to the host that has joined the group.

If another host (for example, Host 4) sends an IGMP join message for the same group (Figure 17), the CPU receives that message and adds the port number of Host 4 to the multicast forwarding table as shown in Table 8.

Figure 17 Second Host Joining a Multicast Group

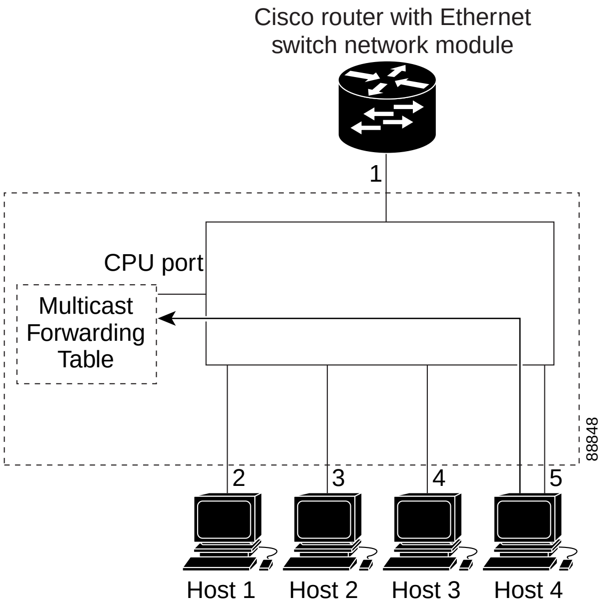

Cisco router with Ethernet switch network module
1
CPU port
Multicast Forwarding Table
88848
2
3
4
5
Host 1
Host 2
Host 3
Host 4

Table 8 Updated Multicast Forwarding Table

Destination Address	Type of Packet	Ports
0100.5e01.0203	!IGMP	1, 2, 5

Leaving a Multicast Group

The router sends periodic IP multicast general queries, and the switch responds to these queries with one join response per MAC multicast group. As long as at least one host in the VLAN needs multicast traffic, the switch responds to the router queries, and the router continues forwarding the multicast traffic to the VLAN. The switch only forwards IP multicast group traffic to those hosts listed in the forwarding table for that IP multicast group.

When hosts need to leave a multicast group, they can either ignore the periodic general-query requests sent by the router, or they can send a leave message. When the switch receives a leave message from a host, it sends out a group-specific query to determine if any devices behind that interface are interested in traffic for the specific multicast group. If, after a number of queries, the router processor receives no reports from a VLAN, it removes the group for the VLAN from its multicast forwarding table.

Global Storm-Control

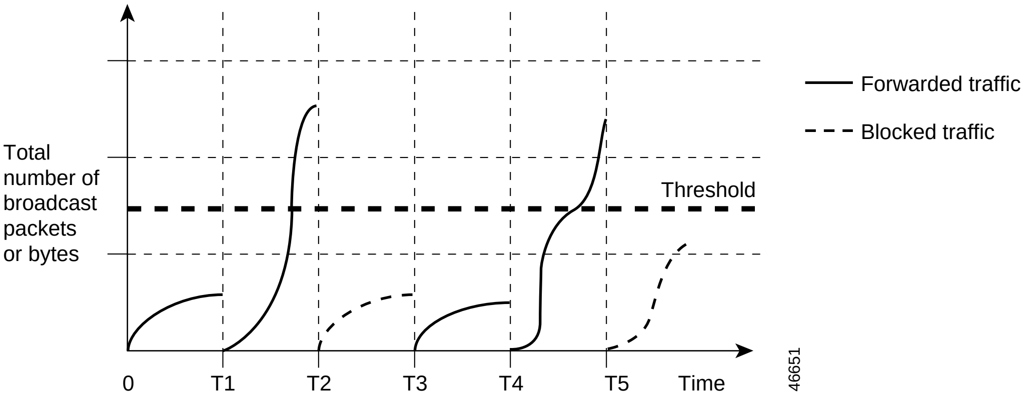
Global storm-control prevents switchports on a LAN from being disrupted by a broadcast, multicast, or unicast storm on one of the interfaces. A LAN storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation or in the network configuration can cause a storm.

Global storm-control monitors incoming traffic statistics over a time period and compares the measurement with a predefined suppression level threshold. The threshold represents the percentage of the total available bandwidth of the port. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level. Global storm-control is disabled by default.

The switch supports global storm-control for broadcast, multicast, and unicast traffic. This example of broadcast suppression can also be applied to multicast and unicast traffic.

The graph in Figure 18 shows broadcast traffic patterns on an interface over a given period of time. In this example, the broadcast traffic exceeded the configured threshold between time intervals T1 and T2 and between T4 and T5. When the amount of specified traffic exceeds the threshold, all traffic of that kind is dropped. Therefore, broadcast traffic is blocked during those intervals. At the next time interval, if broadcast traffic does not exceed the threshold, it is again forwarded.

Figure 18 Broadcast Suppression Example



When global storm-control is enabled, the switch monitors packets passing from an interface to the switching bus and determines if the packet is unicast, multicast, or broadcast. The switch monitors the number of broadcast, multicast, or unicast packets received within the 1-second time interval, and when a threshold for one type of traffic is reached, that type of traffic is dropped. This threshold is specified as a percentage of total available bandwidth that can be used by broadcast (multicast or unicast) traffic.

The combination of broadcast suppression threshold numbers and the 1-second time interval control the way the suppression algorithm works. A higher threshold allows more packets to pass through. A threshold value of 100 percent means that no limit is placed on the traffic.

Note Because packets do not arrive at uniform intervals, the 1-second time interval during which traffic activity is measured can affect the behavior of global storm-control.

The switch continues to monitor traffic on the port, and when the utilization level is below the threshold level, the type of traffic that was dropped is forwarded again.

You use the **storm-control broadcast**, **storm-control multicast**, and **storm-control unicast** interface configuration commands to set up the global storm-control threshold value.

Global storm-control and per-port storm-control cannot be enabled at the same time.

Per-Port Storm-Control

A packet storm occurs when a large number of broadcast, unicast, or multicast packets are received on a port. Forwarding these packets can cause the network to slow down or to time out. By default, per-port storm-control is disabled.

Per-port storm-control uses rising and falling thresholds to block and then restore the forwarding of broadcast, unicast, or multicast packets. You can also set the switch to shut down the port when the rising threshold is reached.

Per-port storm-control uses a bandwidth-based method to measure traffic activity. The thresholds are expressed as a percentage of the total available bandwidth that can be used by the broadcast, multicast, or unicast traffic.

The rising threshold is the percentage of total available bandwidth associated with multicast, broadcast, or unicast traffic before forwarding is blocked. The falling threshold is the percentage of total available bandwidth below which the switch resumes normal forwarding. In general, the higher the level, the less effective the protection against broadcast storms.

Per-port storm control and global storm-control cannot be enabled at the same time.

Port Security

You can use port security to block input to an Ethernet, Fast Ethernet, or Gigabit Ethernet port when the MAC address of the station attempting to access the port is different from any of the MAC addresses specified for that port. Alternatively, you can use port security to filter traffic destined to or received from a specific host based on the host MAC address.

Ethernet Switching in Cisco AVVID Architecture

This section describes the Ethernet switching capabilities of the Ethernet switch network module, which is designed to work as part of the Cisco Architecture for Voice, Video, and Integrated Data (AVVID) solution.

The section outlines some of the concepts involved in configuring Ethernet ports on the Ethernet switch network module to support Cisco IP phones in a branch office on your network. Also included is a section describing the default settings on the Ethernet switch network module.

The following topics are included:

- Configuring the Ethernet Switch Network Module for Cisco AVVID/IP Telephony, page 40
- Default Switch Configuration, page 41

Configuring the Ethernet Switch Network Module for Cisco AVVID/IP Telephony

The Ethernet switch network module has sixteen 10/100 switched Ethernet ports with integrated inline power and QoS features that make it an ideal choice for extending Cisco AVVID (Architecture for Voice, Video and Integrated Data) based voice-over-IP (VoIP) networks to small branch offices.

As an access gateway switch, the Ethernet switch network module can be deployed as a component of a centralized call-processing network using a centrally deployed Cisco CallManager (CCM). Instead of deploying and managing key systems or PBXs in small branch offices, applications are centrally located at the corporate headquarters or data center and are accessed via the IP WAN.

Default Switch Configuration

By default, the Ethernet switch network module provides the following settings with respect to Cisco AVVID:

- All switch ports are in access VLAN 1.
- All switch ports are static access ports, not 802.1Q trunk ports.
- Default voice VLAN is not configured on the switch.
- Inline power is automatically supplied on the 10/100 ports.

Stacking

Layer 2 switching may be extended in the router by connecting the Gigabit Ethernet (GE) ports of the Ethernet switch network module. This connection sustains a line-rate traffic similar to the switch fabric found in Cisco Catalyst switches and forms a single VLAN consisting of all ports in multiple Ethernet switch network modules.

- MAC address entries learned via intrachassis stacking are not displayed.
- Link status of intrachassis stacked ports are filtered.

Flow Control

Flow-control is a feature that Gigabit Ethernet ports use to inhibit the transmission of incoming packets. If a buffer on a Gigabit Ethernet port runs out of space, the port transmits a special packet that requests remote ports to delay sending packets for a period of time. This special packet is called a *pause frame*.

Using Flow-Control Keywords

Table 9 describes guidelines for using different configurations of the **send** and **receive** keywords with the **set port flowcontrol** command.

Table 9 Gigabit Ethernet Flow-Control Keyword Functions

Configuration	Description
send on	Enables a local port to send pause frames to a remote port. Use **send on** when a remote port is set to **receive on** or **receive desired**.
send off	Prevents a local port from sending pause frames to a remote port. Use **send off** when a remote port is set to **receive off** or **receive desired**.
send desired	Indicates preference to send pause frames, but autonegotiates flow control. You can use **send desired** when a remote port is set to **receive on**, **receive off**, or **receive desired**.
receive on	Enables a local port to process pause frames that a remote port sends. Use **receive on** when a remote port is set to **send on** or **send desired**.
receive off	Prevents a local port from processing pause frames. Use **receive off** when a remote port is set to **send off** or **send desired**.
receive desired	Indicates preference to process pause frames, but autonegotiates flow control. You can use **receive desired** when a remote port is set to **send on**, **send off**, or **send desired**.

Fallback Bridging

With fallback bridging, the switch bridges together two or more VLANs or routed ports, essentially connecting multiple VLANs within one bridge domain. Fallback bridging forwards traffic that the multilayer switch does not route and forwards traffic belonging to a nonroutable protocol such as DECnet.

Fallback bridging does not allow the spanning trees from the VLANs being bridged to collapse; each VLAN has its own Spanning Tree Protocol (STP) instance and a separate spanning tree, called the VLAN-bridge spanning tree, which runs on top of the bridge group to prevent loops.

A VLAN bridge domain is represented using the switch virtual interface (SVI). A set of SVIs and routed ports (which do not have any VLANs associated with them) can be configured to form a bridge group.

Recall that an SVI represents a VLAN of switch ports as one interface to the routing or bridging function in the system. Only one SVI can be associated with a VLAN, and it is only necessary to configure an SVI for a VLAN when you want to route between VLANs, to fallback-bridge nonroutable protocols between VLANs, or to provide IP host connectivity to the switch. A routed port is a physical port that acts like a port on a router, but it is not connected to a router. A routed port is not associated with a particular VLAN, does not support subinterfaces, but behaves like a normal routed interface.

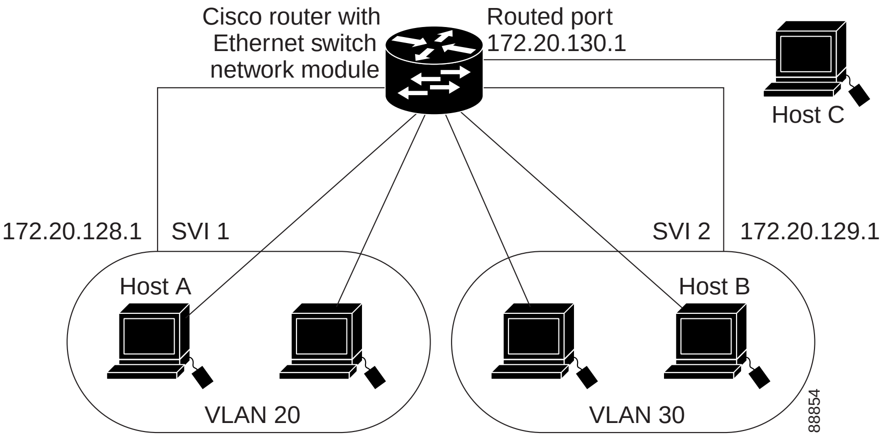
A bridge group is an internal organization of network interfaces on a switch. Bridge groups cannot be used to identify traffic switched within the bridge group outside the switch on which they are defined. Bridge groups on the same switch function as distinct bridges; that is, bridged traffic and bridge protocol data units (BPDUs) cannot be exchanged between different bridge groups on a switch. An interface can be a member of only one bridge group. Use a bridge group for each separately bridged (topologically distinct) network connected to the switch.

The purpose of placing network interfaces into a bridge group is twofold:

- To bridge all nonrouted traffic among the network interfaces making up the bridge group. If the packet destination address is in the bridge table, it is forwarded on a single interface in the bridge group. If the packet destination address is not in the bridge table, it is flooded on all forwarding interfaces in the bridge group. The bridge places source addresses in the bridge table as it learns them during the bridging process.
- To participate in the spanning-tree algorithm by receiving, and in some cases sending, BPDUs on the LANs to which they are attached. A separate spanning process runs for each configured bridge group. Each bridge group participates in a separate spanning-tree instance. A bridge group establishes a spanning-tree instance based on the BPDUs it receives on only its member interfaces.

Figure 19 shows a fallback bridging network example. The multilayer switch has two interfaces configured as SVIs with different assigned IP addresses and attached to two different VLANs. Another interface is configured as a routed port with its own IP address. If all three of these ports are assigned to the same bridge group, non-IP protocol frames can be forwarded among the end stations connected to the switch.

Figure 19 Fallback Bridging Network Example



Benefits

- Statistical gains by combining multiple traffic types over a common IP infrastructure.
- Long distance savings
- Support for Intra-chassis stacking
- Voice connectivity over data applications
- IPSEC, ACL, VPN and Firewall options
- New broadband WAN options

The Interface Range Specification feature makes configuration easier for these reasons:

- Identical commands can be entered once for a range of interfaces, rather than being entered separately for each interface.
- Interface ranges can be saved as macros.

Restrictions

The following functions are not supported in this release:

- CGMP client, CGMP fast-leave
- Dynamic ports
- Dynamic access ports
- Secure ports
- Dynamic trunk protocol
- Dynamic VLANs
- GARP, GMRP, and GVRP
- ISL tagging (The chip does not support ISL.)
- Layer 3 switching onboard
- Monitoring of VLANs

- Multi-VLAN ports Network Port
- Shared STP instances
- STP uplink fast for clusters
- VLAN-based SPAN
- VLAN Query Protocol
- VTP Pruning Protocol
- Web-based management interface

Related Features and Technologies

- IP Phone Telephony
- Voice over IP (VoIP)
- Wireless LAN

Related Documents

For information about installing voice network modules and voice interface cards in Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers refer to these documents:

- *Cisco 2600 Series Modular Routers Quick Start Guide*
- *Cisco 2600 Series Hardware Installation Guide*
- Quick Start Guides for Cisco 3600 series routers
- *Cisco 3600 Series Hardware Installation Guide*
- Quick start guides for Cisco 3700 series routers
- Hardware installation documents for Cisco 3700 series
- *WAN Interface Card Hardware Installation Guide*

For information about configuring Voice over IP features, refer to these documents:

- *Cisco 2600 Series Software Configuration Guide*
- *Cisco IOS Voice, Video, and Fax Configuration Guide,* Release 12.2
- *Cisco IOS Voice, Video, and Fax Command Reference,* Release 12.2

For more information on Flow control, refer to the following document:

- *Configuring Gigabit Ethernet Switching*

Supported Platforms

- Cisco 2600 series
- Cisco 3600 series
- Cisco 3700 series

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Supported Standards, MIBs, and RFCs

Standards

- 802.1d
- 802.1p
- 802.1q
- 802.1x

MIBs

- RFC 1213
- IF MIB
- RFC 2037 ENTITY MIB
- CISCO-CDP-MIB
- CISCO-IMAGE-MIB
- CISCO-FLASH-MIB
- OLD-CISCO-CHASSIS-MIB
- CISCO-VTP-MIB
- CISCO-HSRP-MIB
- OLD-CISCO-TS-MIB
- CISCO-ENTITY-ASSET-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- BRIDGE MIB (RFC 1493)
- CISCO-VLAN-MEMBERSHIP-MIB
- CISCO-VLAN-IFINDEX-RELATIONSHIP-MIB
- RMON1-MIB
- PIM-MIB
- CISCO-STP-EXTENSIONS-MIB
- OSPF MIB (RFC 1253)

- IPMROUTE-MIB
- CISCO-MEMORY-POOL-MIB
- ETHER-LIKE-MIB (RFC 1643)
- CISCO-ENTITY-FRU-CONTROL-MIB.my
- CISCO-RTTMON-MIB
- CISCO-PROCESS-MIB
- CISCO-COPS-CLIENT-MIB

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

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

RFCs

- RFC 2284, *PPP Extensible Authentication Protocol (EAP)*

Prerequisites

- Cisco IOS Release 12.2 or later release
- Basic configuration of the Cisco 2600 series, Cisco 3600 series, or Cisco 3700 series router

In addition, complete the following tasks before configuring this feature:

- Configure IP routing

  For more information on IP routing, refer to the *Cisco IOS IP Configuration Guide,* Release 12.2.

- Set up the call agents

  For more information on setting up call agents, refer to the documentation that accompanies the call agents used in your network configuration.

Configuration Tasks

See the following sections for configuration tasks for the Ethernet switch network module.

- Configuring Layer 2 Interfaces, page 47
- Configuring VLANs, page 52
- Configuring VLAN Trunking Protocol, page 54
- Configuring Layer 2 EtherChannels (Port-Channel Logical Interfaces), page 56
- Configuring 802.1x Authentication, page 59
- Configuring Spanning Tree, page 67
- Configuring MAC Table Manipulation — Port Security, page 72
- Configuring Cisco Discovery Protocol, page 74
- Configuring Switched Port Analyzer, page 76
- Configuring Network Security with ACLs, page 78
- Configuring Quality of Service (QoS), page 86

- Configuring Power Management on the Interface, page 98
- Configuring IP Multicast Layer 3 Switching, page 98
- Configuring IGMP Snooping, page 102
- Configuring Global Storm-Control, page 104
- Configuring Per-Port Storm-Control, page 106
- Configuring Separate Voice and Data Subnets, page 107
- Configuring Intrachassis Stacking, page 119
- Configuring Flow Control on Gigabit Ethernet Ports, page 119
- Configuring Layer 3 Interfaces, page 120
- Configuring Fallback Bridging, page 121

Configuring Layer 2 Interfaces

- Configuring a Range of Interfaces (required)
- Defining a Range Macro (optional)
- Configuring Layer 2 Optional Interface Features (optional)
- Configuring an Ethernet Interface as a Layer 2 Trunk (optional)
- Configuring an Ethernet Interface as a Layer 2 Access (optional)

Configuring a Range of Interfaces

To configure a range of interfaces, use the **interface range** command in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface range** {**vlan** *vlan-id* - *vlan-id*} | {{**ethernet** | **fastethernet** | **macro** *macro-name*} *slot/interface* - *interface*}[, {{**ethernet** | **fastethernet** | **macro** *macro-name*} *slot/interface* - *interface*}]	Selects the range of interfaces to be configured. • The space before the dash is required. For example, the command **interface range fastethernet 1 - 5** is valid; the command **interface range fastethernet 1-5** is not valid. • You can enter one macro or up to five comma-separated ranges. • Comma-separated ranges can include both VLANs and physical interfaces. • You are not required to enter spaces before or after the comma. • The **interface range** command only supports VLAN interfaces that are configured with the **interface vlan** command.

Defining a Range Macro

To define an interface range macro, use the **define interface-range** command in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`define interface-range`** *`macro-name`* `{`**`vlan`** *`vlan-id`* `-` *`vlan-id`*`} | {{`**`ethernet`** `|` **`fastethernet`**`}` *`slot/interface`* `-` *`interface`*`} [,` `{{`**`ethernet`** `|` **`fastethernet`**`}` *`slot/interface`* `-` *`interface`*`}]`	Defines the interface-range macro and save it in NVRAM.

Verifying Configuration of a Range of Interfaces

Step 1 Use the **show running-configuration** command to show the defined interface-range macro configuration:

```
Router# show running-configuration | include define define interface-range enet_list
FastEthernet5/1 - 4
```

Configuring Layer 2 Optional Interface Features

- Interface Speed and Duplex Configuration Guidelines, page 48
- Configuring the Interface Speed, page 49
- Configuring the Interface Duplex Mode, page 49
- Configuring a Description for an Interface, page 50
- Configuring an Ethernet Interface as a Layer 2 Trunk, page 50
- Configuring an Ethernet Interface as a Layer 2 Access, page 52

Interface Speed and Duplex Configuration Guidelines

When configuring an interface speed and duplex mode, note these guidelines:

- If both ends of the line support autonegotiation, Cisco highly recommends the default autonegotiation settings.
- If one interface supports autonegotiation and the other end does not, configure duplex and speed on both interfaces; do not use the **auto** setting on the supported side.
- Both ends of the line need to be configured to the same setting. For example, both hard-set or both auto-negotiate. Mismatched settings are not supported.

⚠ **Caution** Changing the interface speed and duplex mode configuration might shut down and reenable the interface during the reconfiguration.

Configuring the Interface Speed

To set the interface speed, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface fastethernet** *slot*/*interface*	Specifies the interface to be configured.
Step 2	Router(config-if)# **speed** [**10** | **100** | **auto**]	Sets the interface speed of the interface.

Note If you set the interface speed to auto on a 10/100-Mbps Ethernet interface, both speed and duplex are autonegotiated.

Configuring the Interface Duplex Mode

To set the duplex mode of an Ethernet or Fast Ethernet interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **interface fastethernet** *slot*/*interface*	Selects the interface to be configured.
Step 2	Router(config-if)# **duplex** [**auto** | **full** | **half**]	Sets the duplex mode of the interface.

Note If you set the port speed to auto on a 10/100-Mbps Ethernet interface, both speed and duplex are autonegotiated. You cannot change the duplex mode of autonegotiation interfaces.

The following example shows how to set the interface duplex mode to full on Fast Ethernet interface 5/4:

```
Router(config)# interface fastethernet 5/4
Router(config-if)# duplex full
```

Verifying Interface Speed and Duplex Mode Configuration

Step 1 Use the **show interfaces** command to verify the interface speed and duplex mode configuration for an interface:

```
Router# show interfaces fastethernet 1/4

FastEthernet1/4 is up, line protocol is down
  Hardware is Fast Ethernet, address is 0000.0000.0c89 (bia 0000.0000.0c89)
  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)
  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
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     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
     3 packets output, 1074 bytes, 0 underruns(0/0/0)
     0 output errors, 0 collisions, 5 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Router#
```

Configuring 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, use the **description** command in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)#` **`description`** *`string`*	Adds a description for an interface.

Configuring an Ethernet Interface as a Layer 2 Trunk

To configure an Ethernet interface as a Layer 2 trunk, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{`**`ethernet`** `|` **`fastethernet`**`}` *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`shutdown`**	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete. **Note** Encapsulation is always dot1q.
Step 3	`Router(config-if)#` **`switch port mode trunk`**	Configures the interface as a Layer 2 trunk.
Step 4	`Router(config-if)#` **`switch port trunk native vlan`** *`vlan-num`*	For 802.1Q trunks, specifies the native VLAN.
Step 5	`Router(config-if)#` **`switch port trunk allowed vlan`** `{`**`add`** `|` **`except`** `|` **`none`** `|` **`remove`**`}` *`vlan1`*`[,`*`vlan`*`[,`*`vlan`*`[,...]]`	(Optional) Configures the list of VLANs allowed on the trunk. All VLANs are allowed by default. You cannot remove any of the default VLANs from a trunk.
Step 6	`Router(config-if)#` **`no shutdown`**	Activates the interface. (Required only if you shut down the interface.)
Step 7	`Router(config-if)#` **`end`**	Exits configuration mode.

Note Ports do not support Dynamic Trunk Protocol (DTP). Ensure that the neighboring switch is set to a mode that will not send DTP.

Verifying an Ethernet Interface as a Layer 2 Trunk

Step 1 Use the following **show** commands to verify the configuration of an Ethernet interface as a Layer 2 trunk:

```
Router# show running-config interface fastethernet 5/8

Building configuration...
Current configuration:
!
interface FastEthernet5/8
 no ip address
 switchport
 switchport trunk encapsulation dot1q
end
```

Step 2

```
Router# show interfaces fastethernet 5/8 switchport

Name: Fa5/8
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Disabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: false
Unknown unicast blocked: false
Unknown multicast blocked: false
Broadcast Suppression Level: 100
Multicast Suppression Level: 100
Unicast Suppression Level: 100
Voice VLAN: none
Appliance trust: none
```

Step 3

```
Router# show interfaces fastethernet 5/8 trunk

Port      Mode         Encapsulation  Status        Native vlan
Fa1/15    off          802.1q         not-trunking  1
Port      Vlans allowed on trunk
Fa1/15    1
Port      Vlans allowed and active in management domain
Fa1/15    1
Port      Vlans in spanning tree forwarding state and not pruned
Fa1/15    1
```

Configuring an Ethernet Interface as a Layer 2 Access

To configure an Ethernet Interface as a Layer 2 access use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{`**`ethernet`** `|` **`fastethernet`**`}` *`slot/port`*	Selects the interface to configure.
Step 2	`Router(config-if)#` **`shutdown`**	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete. Encapsulation is always dot1q.
Step 3	`Router(config-if)#` **`switchport mode access`**	Configures the interface as a Layer 2 access.
Step 4	`Router(config-if)#` **`switchport access vlan`** *`vlan-num`*	For access ports, specifies the access VLAN.
Step 5	`Router(config-if)#` **`no shutdown`**	Activates the interface. (Required only if you shut down the interface.)
Step 6	`Router(config-if)#` **`end`**	Exits configuration mode.

Verifying an Ethernet Interface as a Layer 2 Access

Step 1 Use the **show running-config interface** command to verify the running configuration of the interface:

```
Router# show running-config interface {ethernet | fastethernet} slot/port
```

Step 1 Use the **show interfaces** command to verify the switch port configuration of the interface:

```
Router# show interfaces [ethernet | fastethernet] slot/port switchport
```

Configuring VLANs

This section describes how to configure the VLANs on the Ethernet switch network modules, and it contains the following sections:

- Configuring VLANs (optional)
- Deleting a VLAN from the Database (optional)

Configuring VLANs

To configure an Ethernet Interface as a Layer 2 access, use the following commands beginning in EXEC mode:

	Command	Purpose
Step 1	`Router#` **`vlan database`**	Enters VLAN configuration mode.
Step 2	`Router(vlan)#` **`vlan`** *`vlan-id`*	Adds an Ethernet VLAN.
Step 3	`Router(vlan)#` **`exit`**	Updates the VLAN database, propagates it throughout the administrative domain, and returns to privileged EXEC mode.

Verifying the VLAN Configuration.

Step 1 Use the **show vlan name** command to verify the VLAN configuration:

```
Router# show vlan name VLAN0003

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa1/0, Fa1/1, Fa1/2, Fa1/3
                                                Fa1/4, Fa1/5, Fa1/6, Fa1/7
                                                Fa1/8, Fa1/9, Fa1/10, Fa1/11
                                                Fa1/12, Fa1/13, Fa1/14, Fa1/15
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        1002   1003
1002 fddi  101002     1500  -      -      -        -    -        1      1003
1003 tr    101003     1500  1005   0      -        -    srb      1      1002
1004 fdnet 101004     1500  -      -      1        ibm  -        0      0
1005 trnet 101005     1500  -      -      1        ibm  -        0      0
Router#
```

Deleting a VLAN from the Database

When you delete a VLAN from a switch that is in VTP server mode, the VLAN is removed from all switches in the VTP domain. When you delete a VLAN from a switch that is in VTP transparent mode, the VLAN is deleted only on that specific switch.

You cannot delete the default VLANs for the different media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005.

To delete a VLAN from the database, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# vlan database`	Enters VLAN configuration mode.
Step 2	`Router(vlan)# no vlan vlan-id`	Deletes the VLAN.
Step 3	`Router(vlan)# exit`	Updates the VLAN database, propagate it throughout the administrative domain, and return to privileged EXEC mode.

Verifying VLAN Deletion

Step 1 Use the **show vlan-switch brief** command to verify that a VLAN has been deleted from a switch:

```
Router# show vlan-switch brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/9, Fa0/14, Gi0/0
2    VLAN0002                         active
3    VLAN0003                         active    Fa0/4, Fa0/5, Fa0/10, Fa0/11
4    VLAN0004                         active    Fa0/6, Fa0/7, Fa0/12, Fa0/13
5    VLAN0005                         active
40   VLAN0040                         active    Fa0/15
50   VLAN0050                         active
1000 VLAN1000                         active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Router#
```

Configuring VLAN Trunking Protocol

This section describes how to configure the VLAN Trunking Protocol (VTP) on the Ethernet switch network module, and contains the following sections:

- Configuring the VTP Server, page 54
- Configuring a VTP Client, page 55
- Disabling VTP (VTP Transparent Mode), page 55
- Configuring VTP version 2, page 55

Configuring the VTP Server

When a switch is in VTP server mode, you can change the VLAN configuration and have it propagate throughout the network.

To configure the switch as a VTP server, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# vlan database`	Enters VLAN configuration mode.
Step 2	`Router(vlan)# vtp server`	Configures the switch as a VTP server.
Step 3	`Router(vlan)# vtp domain domain-name`	Defines the VTP domain name, which can be up to 32 characters long.
Step 4	`Router(vlan)# vtp password password-value`	(Optional) Sets a password, which can be from 8 to 64 characters long, for the VTP domain.
Step 5	`Router(vlan)# exit`	Exits VLAN configuration mode.

Configuring a VTP Client

When a switch is in VTP client mode, you cannot change the VLAN configuration on the switch. The client switch receives VTP updates from a VTP server in the management domain and modifies its configuration accordingly.

To configure the switch as a VTP client, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# vlan database`	Enters VLAN configuration mode.
Step 2	`Router(vlan)# vtp client`	Configures the switch as a VTP client.
Step 3	`Router(vlan)# exit`	Exits VLAN configuration mode.

Disabling VTP (VTP Transparent Mode)

When you configure the switch as VTP transparent, you disable VTP on the switch. A VTP transparent switch does not send VTP updates and does not act on VTP updates received from other switches. However, a VTP transparent switch running VTP version 2 does forward received VTP advertisements out all of its trunk links.

To disable VTP on the switch, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# vlan database`	Enters VLAN configuration mode.
Step 2	`Router(vlan)# vtp transparent`	Configures VTP transparent mode.
Step 3	`Router(vlan)# exit`	Exits VLAN configuration mode.

Configuring VTP version 2

To enable VTP version 2, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# vlan database`	Enters VLAN configuration mode.
Step 2	`Router(vlan)# [no] vtp v2-mode`	Enables VTP version 2. Use the **no** keyword to disable VTP version 2.
Step 3	`Router(vlan)# exit`	Exits VLAN configuration mode.

Verifying VTP

Step 1 Use the **show vtp status** to verify VTP status:

```
Router# show vtp status

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

Configuring Layer 2 EtherChannels (Port-Channel Logical Interfaces)

Configuring Layer 2 EtherChannels (Port-Channel Logical Interfaces)

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

Note Cisco IOS software creates port-channel interfaces for Layer 2 EtherChannels when you configure Layer 2 Ethernet interfaces with the **channel-group** command. You cannot put Layer 2 Ethernet interfaces into a manually created port-channel interface.

Note Layer 2 interfaces must be connected and functioning for Cisco IOS software to create port-channel interfaces for Layer 2 EtherChannels.

To configure Layer 2 Ethernet interfaces as a Layer 2 EtherChannel, use the following commands beginning in global configuration mode for each interface:

	Command	Purpose
Step 1	`Router(config)#` **`interface fastethernet`** *`slot/port`*	Selects a physical interface to configure.
Step 2	`Router(config-if)#` **`channel-group`** *`port-channel-number`* **`mode`** `{`**`on`**`}`	Configures the interface in a port-channel.
Step 3	`Router(config-if)#` **`end`**	Exits configuration mode.

Verifying Layer 2 EtherChannels

Use the following **show** commands to verify Layer 2 EtherChannels:

Step 1

```
Router# show running-config interface fastethernet 5/6
```

```
Building configuration...

Current configuration:
!
interface FastEthernet5/6
 no ip address
 switchport
 switchport access vlan 10
 switchport mode access
 channel-group 2 mode on
end
```

Step 2

```
Router# show interfaces fastethernet 5/6 etherchannel
```

```
Port state    = EC-Enbld Up In-Bndl Usr-Config
Channel group = 2           Mode = Desirable     Gcchange = 0
Port-channel  = Po2         GC   = 0x00020001
Port indx     = 1           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
```

Step 3

```
Router# show running-config interface port-channel 2
```

```
Building configuration...

Current configuration:
!
interface Port-channel2
 no ip address
 switchport
 switchport access vlan 10
 switchport mode access
end

Router#
```

Step 4

```
Router# 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     55      Fa5/6
  0     AA      Fa5/7

Time since last port bundled:    00h:23m:33s    Fa5/6
```

Configuring EtherChannel Load Balancing

To configure EtherChannel load balancing, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# port-channel load-balance {src-mac | dst-mac | src-dst-mac | src-ip | dst-ip | src-dst-ip}`	Configures EtherChannel load balancing, use the **no** form of this command to return EtherChannel load balancing to the default configuration.
Step 2	`Router(config)# end`	Exits configuration mode.

Note For new load balancing to take affect, the EtherChannel must be first configured to the default configuration.

Verifying EtherChannel Load Balancing

Step 1 Use the **show etherchannel load-balance** command to verify Layer 2 EtherChannel load balancing:

```
Router# show etherchannel load-balance

Source XOR Destination IP address
Router#
```

Removing an Interface from an EtherChannel

To remove an Ethernet interface from an EtherChannel, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# [no] port-channel load-balance {src-mac | dst-mac | src-dst-mac | src-ip | dst-ip | src-dst-ip}`	Configures EtherChannel load balancing. Use the **no** keyword to return EtherChannel load balancing to the default configuration.
Step 2	`Router(config)# end`	Exits configuration mode.

Configuring Removing an EtherChannel

To remove an EtherChannel, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# no interface port-channel` *port-channel-number*	Removes the port-channel interface.
Step 2	`Router(config)# end`	Exits configuration mode.

Verify Removing an EtherChannel

Step 1 Use the **show etherchannel summary** command to verify that the Etherchannel is removed:

```
Router# show etherchannel summary

Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        R - Layer3      S - Layer2
        U - in use
Group Port-channel  Ports
-----+------------+-----------------------------------------------------------

Router#
```

Configuring 802.1x Authentication

This section describes how to configure 802.1x port-based authentication on the Ethernet switch network module:

- Understanding the Default 802.1x Configuration, page 60
- Enabling 802.1x Authentication, page 61
- Configuring the Switch-to-RADIUS-Server Communication, page 62
- Enabling Periodic Reauthentication, page 63
- Changing the Quiet Period, page 64
- Changing the Switch-to-Client Retransmission Time, page 64

- Setting the Switch-to-Client Frame-Retransmission Number, page 65
- Enabling Multiple Hosts, page 66

Understanding the Default 802.1x Configuration

Table 10 shows the default 802.1x configuration.

Table 10 Default 802.1x Configuration

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled.
RADIUS server • IP address • UDP authentication port • Key	 • None specified. • 1645. • None specified.
Per-interface 802.1x enable state	Disabled (force-authorized). The port transmits and receives normal traffic without 802.1x-based authentication of the client.
Periodic reauthentication	Disabled.
Number of seconds between reauthentication attempts	3600 seconds.
Quiet period	60 seconds (number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client).
Retransmission time	30 seconds (number of seconds that the switch should wait for a response to an EAP request/identity frame from the client before retransmitting the request).
Maximum retransmission number	2 times (number of times that the switch will send an EAP-request/identity frame before restarting the authentication process).
Multiple host support	Disabled.
Client timeout period	30 seconds (when relaying a request from the authentication server to the client, the amount of time the switch waits for a response before retransmitting the request to the client). This setting is not configurable.
Authentication server timeout period	30 seconds (when relaying a response from the client to the authentication server, the amount of time the switch waits for a reply before retransmitting the response to the server). This setting is not configurable.

802.1x Configuration Guidelines

These are the 802.1x authentication configuration guidelines:

- When the 802.1x protocol is enabled, ports are authenticated before any other Layer 2 feature is enabled.
- The 802.1x protocol is supported on Layer 2 static-access ports, but it is not supported on these port types:
  - Trunk port—If you try to enable 802.1x on a trunk port, an error message appears, and 802.1x is not enabled. If you try to change the mode of an 802.1x-enabled port to trunk, the port mode is not changed.
  - EtherChannel port—Before enabling 802.1x on the port, you must first remove the port from the EtherChannel before enabling 802.1x on it. If you try to enable 802.1x on an EtherChannel or on an active port in an EtherChannel, an error message appears, and 802.1x is not enabled. If you enable 802.1x on a not-yet active port of an EtherChannel, the port does not join the EtherChannel.
  - Switch Port Analyzer (SPAN) destination port—You can enable 802.1x on a port that is a SPAN destination port; however, 802.1x is disabled until the port is removed as a SPAN destination. You can enable 802.1x on a SPAN source port.

Enabling 802.1x Authentication

To enable 802.1x port-based authentication, you must enable AAA and specify the authentication method list. A method list describes the sequence and authentication methods to be queried to authenticate a user.

The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle, the authentication process stops, and no other authentication methods are attempted.

Beginning in privileged EXEC mode, follow these steps to configure 802.1x port-based authentication. This procedure is required.

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa new-model**	Enables AAA.
Step 3	**aaa authentication dot1x {default |** *listname***}** *method1* [*method2...*]	Creates an 802.1x authentication method list. To create a default list that is used when a named list is *not* specified in the **authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces. Enter at least one of these keywords: • **group radius**—Use the list of all RADIUS servers for authentication. • **none**—Use no authentication. The client is automatically authenticated without the switch using the information supplied by the client.

	Command	Purpose
Step 4	**interface** *interface-id*	Enters interface configuration mode, and specify the interface to be enabled for 802.1x authentication.
Step 5	**dot1x port-control auto**	Enables 802.1x on the interface. For feature interaction information with trunk, dynamic, dynamic-access, EtherChannel, secure, and SPAN ports see the "802.1x Configuration Guidelines" section on page 61.
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show dot1x**	Verifies your entries. Check the Status column in the 802.1x Port Summary section of the display. An *enabled* status means the port-control value is set either to **auto** or to **force-unauthorized**.
Step 8	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable 802.1x AAA authentication, use the **no aaa authentication dot1x** {**default** | *list-name*} *method1* [*method2*...] global configuration command. To disable 802.1x, use the **dot1x port-control force-authorized** or the **no dot1x port-control** interface configuration command.

Configuring the Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by their host name or IP address, host name and specific UDP port numbers, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service—for example, authentication—the second host entry configured acts as the fail-over backup to the first one. The RADIUS host entries are tried in the order that they were configured.

Beginning in privileged EXEC mode, follow these steps to configure the RADIUS server parameters on the switch. This procedure is required.

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server host** {*hostname* | *ip-address*} **auth-port** *port-number* **key** *string*	Configures the RADIUS server parameters on the switch. For *hostname* | *ip-address*, specify the host name or IP address of the remote RADIUS server. For **auth-port** *port-number*, specify the UDP destination port for authentication requests. The default is 1645. For **key** *string*, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server. **Note** Always configure the key as the last item in the **radius-server host** command syntax because leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon. If you want to use multiple RADIUS servers, repeat this command.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To delete the specified RADIUS server, use the **no radius-server host** {*hostname* | *ip-address*} global configuration command.

You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius-server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and the **radius-server key** global configuration commands.

You also need to configure some settings on the RADIUS server. These settings include the IP address of the switch and the key string to be shared by both the server and the switch. For more information, refer to the RADIUS server documentation.

Enabling Periodic Reauthentication

You can enable periodic 802.1x client reauthentication and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication attempts is 3600 seconds.

Automatic 802.1x client reauthentication is a global setting and cannot be set for clients connected to individual ports.

Beginning in privileged EXEC mode, follow these steps to enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**dot1x re-authentication**	Enables periodic reauthentication of the client, which is disabled by default.
Step 3	**dot1x timeout re-authperiod** *seconds*	Sets the number of seconds between reauthentication attempts. The range is 1 to 4294967295; the default is 3600 seconds. This command affects the behavior of the switch only if periodic reauthentication is enabled.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show dot1x**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable periodic reauthentication, use the **no dot1x re-authentication** global configuration command. To return to the default number of seconds between reauthentication attempts, use the **no dot1x timeout re-authperiod** global configuration command.

Changing the Quiet Period

When the switch cannot authenticate the client, the switch remains idle for a set period of time, and then tries again. The idle time is determined by the quiet-period value. A failed authentication of the client might occur because the client provided an invalid password. You can provide a faster response time to the user by entering smaller number than the default.

Beginning in privileged EXEC mode, follow these steps to change the quiet period:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**dot1x timeout quiet-period** *seconds*	Sets the number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client. The range is 0 to 65535 seconds; the default is 60.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show dot1x**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return to the default quiet time, use the **no dot1x timeout quiet-period** global configuration command.

Changing the Switch-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the switch with an EAP-response/identity frame. If the switch does not receive this response, it waits a set period of time (known as the retransmission time), and then retransmits the frame.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

Beginning in privileged EXEC mode, follow these steps to change the amount of time that the switch waits for client notification:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**dot1x timeout tx-period** *seconds*	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request. The range is 1 to 65535 seconds; the default is 30.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show dot1x**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return to the default retransmission time, use the **no dot1x timeout tx-period** global configuration command.

Setting the Switch-to-Client Frame-Retransmission Number

In addition to changing the switch-to-client retransmission time, you can change the number of times that the switch sends an EAP-request/identity frame (assuming no response is received) to the client before restarting the authentication process.

Note You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

Beginning in privileged EXEC mode, follow these steps to set the switch-to-client frame-retransmission number:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**dot1x max-req** *count*	Sets the number of times that the switch sends an EAP-request/identity frame to the client before restarting the authentication process. The range is 1 to 10; the default is 2.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show dot1x**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return to the default retransmission number, use the **no dot1x max-req** global configuration command.

Enabling Multiple Hosts

You can attach multiple hosts to a single 802.1x-enabled port as shown in Figure 3 on page 12. In this mode, only one of the attached hosts must be successfully authorized for all hosts to be granted network access. If the port becomes unauthorized (reauthentication fails, and an EAPOL-logoff message is received), all attached clients are denied access to the network.

Beginning in privileged EXEC mode, follow these steps to allow multiple hosts (clients) on an 802.1x-authorized port that has the **dot1x port-control** interface configuration command set to **auto**.

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specify the interface to which multiple hosts are indirectly attached.
Step 3	**dot1x multiple-hosts**	Allows multiple hosts (clients) on an 802.1x-authorized port. Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show dot1x interface** *interface-id*	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable multiple hosts on the port, use the **no dot1x multiple-hosts** interface configuration command.

Resetting the 802.1x Configuration to the Default Values

You can reset the 802.1x configuration to the default values with a single command.

Beginning in privileged EXEC mode, follow these steps to reset the 802.1x configuration to the default values:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**dot1x default**	Resets the configurable 802.1x parameters to the default values.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show dot1x**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Displaying 802.1x Statistics and Status

To display 802.1x statistics for all interfaces, use the **show dot1x statistics** privileged EXEC command. To display 802.1x statistics for a specific interface, use the **show dot1x statistics interface** *interface-id* privileged EXEC command.

To display the 802.1x administrative and operational status for the switch, use the **show dot1x** privileged EXEC command. To display the 802.1x administrative and operational status for a specific interface, use the **show dot1x interface** *interface-id* privileged EXEC command.

Configuring Spanning Tree

Enabling Spanning Tree

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, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`spanning-tree vlan`** *`vlan-id`*	Enables spanning tree on a per-VLAN basis.
Step 2	`Router(config)#` **`end`**	Exits configuration mode.

Verify Spanning Tree

Step 1 Use the **show spanning-tree vlan** command to verify spanning tree configuration:

```
Router# 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
```

Configuring Spanning Tree Port Priority

To configure the spanning tree port priority of an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`ethernet`** `|` **`fastethernet`**`}` *`slot/port}`* `|` *`{port-channel port-channel`*`-number}`	Selects an interface to configure.
Step 2	`Router(config-if)# [`**`no`**`]` **`spanning-tree port-priority`** *`port-priority`*	Configures the port priority for an interface. The of *port-priority* value can be from 1 to 255 in increments of 4. Use the **no** form of this command to restore the defaults.
Step 3	`Router(config-if)# [`**`no`**`]` **`spanning-tree vlan`** *`vlan-id`* **`port-priority`** *`port-priority`*	Configures the VLAN port priority for an interface. The *port-priority* value can be from 1 to 255 in increments of 4. Use the **no** form of this command to restore the defaults.
Step 4	`Router(config-if)#` **`end`**	Exits configuration mode.

Verify Spanning Tree Port Priority

Step 1 Use the **show spanning-tree interface** command to verify spanning-tree interface and the spanning-tree port priority configuration:

```
Router# 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
Router#
```

Configuring Spanning Tree Port Cost

To configure the spanning tree port cost of an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** `{{`**`ethernet`** `|` **`fastethernet`**`}` *`slot/port`*`}` `|` `{`**`port-channel`** *`port-channel-number`*`}`	Selects an interface to configure.
Step 2	`Router(config-if)# [`**`no`**`]` **`spanning-tree cost`** *`port-cost`*	Configures the port cost for an interface. The value of *port-cost* can be from 1 to 200,000,000 (1 to 65,535 in Cisco IOS Releases 12.1(2)E and earlier). Use the **no** form of this command to restore the defaults.

	Command	Purpose
Step 3	Router(config-if)# [**no**] **spanning-tree vlan** *vlan-id* **cost** *port-cost*	Configures the VLAN port cost for an interface. The value of *port-cost* can be from 1 to 65,535. Use the **no** form of this command to restore the defaults.
Step 4	Router(config-if)# **end**	Exits configuration mode.

Verifying Spanning Tree Port Cost

Step 1 Use the **show spanning-tree vlan** command to verify the spanning-tree port cost configuration:

```
Router# show spanning-tree vlan 200

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
Router#
```

Configuring the Bridge Priority of a VLAN

Caution Exercise care when using this command. For most situations **spanning-tree vlan** *vlan-id* **root primary** and the **spanning-tree vlan** *vlan-id* **root secondary** are the preferred commands to modify the bridge priority.

To configure the spanning tree bridge priority of a VLAN, use the following commands in global configuration mode:

	Command	Purpose
Step 1	Router(config)# [**no**] **spanning-tree vlan** *vlan-id* **priority** *bridge-priority*	Configures the bridge priority of a VLAN. The *bridge-priority* value can be from 1 to 65535. Use the **no** keyword to restore the defaults.
Step 2	Router(config)# **end**	Exits configuration mode.

Verifying the Bridge Priority of a VLAN

Step 1 Use the **show spanning-tree vlan bridge** command to verify the bridge priority:

```
Router# 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
```

Configuring the Hello Time

To configure the hello interval for the spanning tree, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# [no] spanning-tree vlan vlan-id hello-time hello-time`	Configures the hello time of a VLAN. The *hello-time* value can be from 1 to 10 seconds. Use the **no** form of this command to restore the defaults.
Step 2	`Router(config)# end`	Exits configuration mode.

Configuring the Forward-Delay Time for a VLAN

To configure the forward delay for the spanning tree, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# [no] spanning-tree vlan vlan-id forward-time forward-time`	Configures the forward time of a VLAN. The value of *forward-time* can be from 4 to 30 seconds. Use the **no** form of this command to restore the defaults.
Step 2	`Router(config)# end`	Exits configuration mode.

Configuring the Maximum Aging Time for a VLAN

To configure the maximum age interval for the spanning tree, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# [no] spanning-tree vlan vlan-id max-age max-age`	Configures the maximum aging time of a VLAN. The value of *max-age* can be from 6 to 40 seconds. Use the **no** form of this command to restore the defaults.
Step 2	`Router(config)# end`	Exits configuration mode.

Configuring the Root Bridge

The Ethernet switch network module maintains a separate 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.

To configure a VLAN instance to become the root bridge, the bridge priority can be modified from the default value (32768) to a significantly lower value so that the bridge becomes the root bridge for the specified VLAN. Use the **spanning-tree vlan** *vlan-id* **root** command to alter the bridge priority.

The switch checks the bridge priority of the current root bridges for each VLAN. The bridge priority for the specified VLANs is set to 8192 if this value will cause the switch to become the root for the specified VLANs.

If any root switch for the specified VLANs has a bridge priority lower than 8192, the switch sets the bridge priority for the specified VLANs to 1 less than the lowest bridge priority.

For example, if all switches in the network have the bridge priority for VLAN 100 set to the default value of 32768, entering the **spanning-tree vlan 100 root primary** command on a switch will set the bridge priority for VLAN 100 to 8192, causing the 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 (that is, the maximum number of bridge hops between any two end stations in the Layer 2 network). When you specify the network diameter, the switch automatically picks an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the spanning tree convergence time. You can use the hello keyword to override the automatically calculated hello time.

Note You should avoid configuring the hello time, forward delay time, and maximum age time manually after configuring the switch as the root bridge.

To configure the switch as the root, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# [no] spanning-tree vlan vlan-id root primary [diameter hops [hello-time seconds]]`	Configures a switch as the root switch. Use the **no** form of this command to restore the defaults.
Step 2	`Router(config)# end`	Exits configuration mode.

Configuring BackboneFast

You can enable BackboneFast to detect indirect link failures and to start the spanning-tree reconfiguration sooner.

Note If you use BackboneFast, you must enable it on all switches in the network. BackboneFast is not supported on Token Ring VLANs. This feature is supported for use with third-party switches.

Beginning in privileged EXEC mode, follow these steps to enable BackboneFast:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**spanning-tree backbonefast**	Enables BackboneFast on the switch.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show spanning-tree vlan** *vlan-id*	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To disable the BackboneFast feature, use the **no spanning-tree backbonefast** global configuration command.

Disabling Spanning Tree

To disable spanning tree on a per-VLAN basis, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# no spanning-tree vlan vlan-id`	Disables spanning tree on a per-VLAN basis.
Step 2	`Router(config)# end`	Exits configuration mode.

Verifying that Spanning Tree is Disabled

Step 1 Use the **show spanning-tree vlan** command to verify the that the spanning tree is disabled:

```
Router# show spanning-tree vlan 200
<...output truncated...>
Spanning tree instance for VLAN 200 does not exist.
Router#
```

Configuring MAC Table Manipulation — Port Security

Port security is implemented by providing the user with the option to make a port secure by allowing only well-known MAC addresses to send in data traffic.

- Enabling Known MAC Address Traffic, page 73
- Creating a Static or Dynamic Entry in the MAC Address Table, page 73
- Configuring Aging Timer-timer, page 74

Enabling Known MAC Address Traffic

To enable the MAC address secure option, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router#` **`mac-address-table secure`** *`mac-address`* **`fastethernet`** *`slot/port`* `[`**`vlan`** *`vlan id`*`]`	Secures the MAC address traffic on the port. Use the **no** form of this command to restore the defaults.
Step 3	`Router(config)#` **`end`**	Exits configuration mode.

Verifying the MAC Address Secure Configuration

Step 1 Use the **show mac-address-table secure** command to verify the configuration:

```
Router# show mac-address-table secure

Secure Address Table:
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
0003.0003.0003          Secure 1  FastEthernet    2/8
```

Creating a Static or Dynamic Entry in the MAC Address Table

To create a static or dynamic entry in the mac address table, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router#` **`mac-address-table`** `[`**`dynamic`** `|` **`static`**`]` *`mac-address`* **`fastethernet`** *`slot/port`* `[`**`vlan`** *`vlan id`*`>]`	Creates static or dynamic entry in the MAC address table.
Step 3	`Router(config)#` **`end`**	Exits configuration mode.

Note Only the port where the link is up will see the dynamic entry validated in the Ethernet switch network module.

Verifying the MAC Address Table

Step 1 Use the **show mac** command to verify the MAC Address Table:

```
Router# show mac

Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
0001.6443.6440          Static       1      Vlan1
0004.c16d.9be1          Dynamic      1      FastEthernet2/13
0004.ddf0.0282          Dynamic      1      FastEthernet2/13
0006.0006.0006          Dynamic      1      FastEthernet2/13
001b.001b.ad45          Dynamic      1      FastEthernet2/13
```

Configuring Aging Timer-timer

To configure the aging timer, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# mac-address-table aging-time seconds`	Configures the MAC address aging-timer age in seconds
Step 3	`Router(config)# end`	Exits configuration mode.

Caution Cisco advises that you not change the aging timer because the Ethernet switch network module could go out of synchronization.

Verifying the Aging Timer

Step 1 Use the **show mac-address-table aging-time** command to verify the aging timer:

```
Router # show mac-address-table aging-time

Mac address aging time 23
```

Configuring Cisco Discovery Protocol

- Enabling Cisco Discovery Protocol, page 75
- Enabling CDP on an Interface, page 75

Enabling Cisco Discovery Protocol

To enable Cisco Discovery Protocol (CDP) globally, use the following command in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# cdp run`	Enables CDP globally.

Verifying the CDP Global Configuration

Step 1 Use the **show cdp** command to verify the CDP configuration:

```
Router# show cdp

Global CDP information:
        Sending CDP packets every 120 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is enabled
Router#
```

Enabling CDP on an Interface

To enable CDP on an interface, use the following command in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# cdp enable`	Enables CDP on an interface.

The following example shows how to enable CDP on Fast Ethernet interface 5/1:

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

Verifying the CDP Interface Configuration

Step 1 Use the **show cdp interface** command to verify the CDP configuration for an interface:

```
Router# show cdp interface fastethernet 5/1

FastEthernet5/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 120 seconds
  Holdtime is 180 seconds
Router#
```

Verifying CDP Neighbors

Step 1 Use the **show cdp neighbors** command to verify information about the neighboring equipment:

```
Router# show cdp neighbors

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
JAB023807H1      Fas 5/3            127          T S       WS-C2948  2/46
JAB023807H1      Fas 5/2            127          T S       WS-C2948  2/45
JAB023807H1      Fas 5/1            127          T S       WS-C2948  2/44
JAB023807H1      Gig 1/2            122          T S       WS-C2948  2/50
JAB023807H1      Gig 1/1            122          T S       WS-C2948  2/49
JAB03130104      Fas 5/8            167          T S       WS-C4003  2/47
JAB03130104      Fas 5/9            152          T S       WS-C4003  2/48
```

Monitoring and Maintaining CDP

To monitor and maintain CDP on your device, use one or more of the following commands in privileged EXEC mode:

Command	Purpose
`Router#` **`clear cdp counters`**	Resets the traffic counters to zero.
`Router#` **`clear cdp table`**	Deletes the CDP table of information about neighbors.
`Router#` **`show cdp`**	Verifies global information such as frequency of transmissions and the holdtime for packets being transmitted.
`Router#` **`show cdp entry entry_name`** `[`*`protocol`*` | `*`version`*`]`	Verifies information about a specific neighbor. The display can be limited to protocol version information.
`Router#` **`show cdp interface`** `[`*`slot/port`*`]`	Verifies information about interfaces on which CDP is enabled.
`Router#` **`show cdp neighbors`** `[`*`slot/port`*`] [`*`detail`*`]`	Verifies information about neighbors. The display can be limited to neighbors on a specific interface and can be expanded to provide more detailed information.
`Router#` **`show cdp traffic`**	Verifies CDP counters, including the number of packets sent and received and checksum errors.

Configuring Switched Port Analyzer

- Specifying the Switched Port Analyzer Session, page 77
- Configuring SPAN Destinations, page 77
- Removing Sources or Destinations from a SPAN Session, page 77

Specifying the Switched Port Analyzer Session

To configure the source for a Switched Port Analyzer (SPAN) session, use the following command in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# monitor session {session-number} {source {interface slot/port} | {vlan vlan-id}} [, | - | rx | tx | both]`	Specifies the SPAN session number (1 or 2), the source interfaces or VLANs, and the traffic direction to be monitored.

Note Multiple SPAN sessions can be configured. But only one SPAN session is supported at a time.

The following example shows how to configure SPAN session 1 to monitor bidirectional traffic from source interface Fast Ethernet 5/1:

```
Router(config)# monitor session 1 source interface fastethernet 5/1
```

Configuring SPAN Destinations

To configure the destination for a SPAN session, use the following command in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# monitor session {session-number} {destination {interface type/num} [, | -] | {vlan vlan-id}}`	Specifies the SPAN session number (1 or 2) and the destination interfaces or VLANs.

Removing Sources or Destinations from a SPAN Session

To remove sources or destinations from a SPAN session, use the following command in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# no monitor session session-number`	Clears existing SPAN configuration for a session.

Configuring Network Security with ACLs

Configuring ACLs on Layer 2 interfaces is the same as configuring ACLs on Cisco routers. The process is briefly described here. For more detailed information on configuring router ACLs, refer to the “Configuring IP Services” chapter in the *Cisco IP Configuration Guide* for Cisco IOS Release 12.2. For detailed information about the commands, refer to *Cisco IOS IP Command Reference* for Cisco IOS Release 12.2. For a list of Cisco IOS features not supported on the Ethernet switch network module, see the following section.

Unsupported Features

The Ethernet switch network module does not support these Cisco IOS router ACL-related features:

- Non-IP protocol ACLs (see Table 11 on page 79).
- Bridge-group ACLs.
- IP accounting.
- ACL support on the outbound direction.
- Inbound and outbound rate limiting (except with QoS ACLs).
- IP packets with a header length of less than five are not be access-controlled.
- Reflexive ACLs.
- Dynamic ACLs.
- ICMP-based filtering.
- IGMP-based filtering.

Creating Standard and Extended IP ACLs

This section describes how to create switch IP ACLs. An ACL is a sequential collection of permit and deny conditions. The switch tests packets against the conditions in an access list one by one. The first match determines whether the switch accepts or rejects the packet. Because the switch stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the switch denies the packet.

An ACL must first be created by specifying an access list number or name and access conditions. The ACL can then be applied to interfaces or terminal lines.

The software supports these styles of ACLs or IP access lists:

- Standard IP access lists use source addresses for matching operations.
- Extended IP access lists use source and destination addresses for matching operations and optional protocol-type information for finer granularity of control.

The next sections describe access lists and the steps for using them.

ACL Numbers

The number you use to denote your ACL shows the type of access list that you are creating. Table 11 lists the access list number and corresponding type and shows whether or not they are supported by the switch. The Ethernet switch network module supports IP standard and IP extended access lists, numbers 1 to 199 and 1300 to 2699.

Table 11 Access List Numbers

ACL Number	Type	Supported
1–99	IP standard access list	Yes
100–199	IP extended access list	Yes
200–299	Protocol type-code access list	No
300–399	DECnet access list	No
400–499	XNS standard access list	No
500–599	XNS extended access list	No
600–699	AppleTalk access list	No
700–799	48-bit MAC address access list	No
800–899	IPX standard access list	No
900–999	IPX extended access list	No
1000–1099	IPX SAP access list	No
1100–1199	Extended 48-bit MAC address access list	No
1200–1299	IPX summary address access list	No
1300–1999	IP standard access list (expanded range)	Yes
2000–2699	IP extended access list (expanded range)	Yes

Note In addition to numbered standard and extended ACLs, you can also create standard and extended named IP ACLs by using the supported numbers. That is, the name of a standard IP ACL can be 1 to 99; the name of an extended IP ACL can be 100 to 199. The advantage of using named ACLs instead of numbered lists is that you can delete individual entries from a named list.

Note An attempt to apply an unsupported ACL feature to an interface produces an error message.

Creating a Numbered Standard ACL

Beginning in privileged EXEC mode, follow these steps to create a numbered standard ACL:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit** | **remark**} {*source source-wildcard* | **host** *source* | **any**}	Defines a standard IP ACL by using a source address and wildcard. The *access-list-number* is a decimal number from 1 to 99 or 1300 to 1999. Enter **deny** or **permit** to specify whether to deny or permit access if conditions are matched. The *source* is the source address of the network or host from which the packet is being sent: • The 32-bit quantity in dotted-decimal format. • The keyword **any** as an abbreviation for *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. You do not need to enter a source-wildcard. • The keyword **host** as an abbreviation for source and source-wildcard of *source* 0.0.0.0. (Optional) The *source-wildcard* applies wildcard bits to the source. (See first bullet item.) **Note** The **log** option is not supported on Ethernet switch network modules.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show access-lists** [*number* | *name*]	Displays the access list configuration.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no access-list** *access-list-number* global configuration command to delete the entire ACL. You cannot delete individual ACEs from numbered access lists.

Note When creating an ACL, remember that, by default, the end of the ACL contains an implicit deny statement for all packets that it did not find a match for before reaching the end. With standard access lists, if you omit the ask from an associated IP host address ACL specification, 0.0.0.0 is assumed to be the mask.

Creating a Numbered Extended ACL

Although standard ACLs use only source addresses for matching, you can use an extended ACL source and destination addresses for matching operations and optional protocol type information for finer granularity of control. Some protocols also have specific parameters and keywords that apply to that protocol.

These IP protocols are supported (protocol keywords are in parentheses in bold): Internet Protocol (**ip**), Transmission Control Protocol (**tcp**), or User Datagram Protocol (**udp**).

Supported parameters can be grouped into these categories:

- TCP
- UDP

Table 12 lists the possible filtering parameters for ACEs for each protocol type.

Table 12 Filtering Parameter ACEs Supported by Different IP Protocols

Filtering Parameter	TCP	UDP
Layer 3 Parameters:		
IP ToS byte[1]	No	No
Differentiated Services Code Point (DSCP)	No	No
IP source address	Yes	Yes
IP destination address	Yes	Yes
Fragments	No	No
TCP or UDP	Yes	Yes
Layer 4 Parameters		
Source port operator	Yes	Yes
Source port	Yes	Yes
Destination port operator	Yes	Yes
Destination port	Yes	Yes
TCP flag	No	No

[1] No support for type of service (TOS) minimize monetary cost bit.

For more details on the specific keywords relative to each protocol, refer to the *Cisco IP Command Reference* for Cisco IOS Release 12.2.

Note The Ethernet switch network module does not support dynamic or reflexive access lists. It also does not support filtering based on the minimize-monetary-cost type of service (TOS) bit.

When creating ACEs in numbered extended access lists, remember that after you create the list, any additions are placed at the end of the list. You cannot reorder the list or selectively add or remove ACEs from a numbered list.

Beginning in privileged EXEC mode, follow these steps to create an extended ACL:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit** | **remark**} *protocol* {*source source-wildcard* | **host** *source* | **any**} [*operator port*] {*destination destination-wildcard* | **host** *destination* | **any**} [*operator port*]	Defines an extended IP access list and the access conditions. The *access-list-number* is a decimal number from 100 to 199 or 2000 to 2699. Enter **deny** or **permit** to specify whether to deny or permit the packet if conditions are matched. For *protocol*, enter the name or number of an IP protocol: **ip**, **tcp**, or **udp**. To match any Internet protocol (including TCP and UDP), use the keyword **ip**. **Note** This step includes options for most IP protocols. The *source* is the number of the network or host from which the packet is sent. The *source-wildcard* applies wildcard bits to the source. The *destination* is the network or host number to which the packet is sent. Defines a destination or source port. • The *operator* can be only **eq** (equal). • If operator is after *source source-wildcard*, conditions match when the source port matches the defined port. • If operator is after *destination destination-wildcard*, conditions match when the destination port matches the defined port. • The *port* is a decimal number or name of a TCP or UDP port. The number can be from 0 to 65535. • Use TCP port names only for TCP traffic. • Use UDP port names only for UDP traffic. The *destination-wildcard* applies wildcard bits to the destination. *Source*, *source-wildcard*, *destination*, and *destination-wildcard* can be specified in three ways: • The 32-bit quantity in dotted-decimal format. • The keyword **any** as an abbreviation for *source* and *source-wildcard* of 0.0.0.0 255.255.255.255 or any source host. • The keyword **host,** followed by the 32-bit quantity in dotted-decimal format, as an abbreviation for a single host with source and source-wildcard of *source* 0.0.0.0. **Note** Only the **ip**, **tcp**, and **udp** protocols are supported on Ethernet switch interfaces.
Step 3	**show access-lists** [*number* | *name*]	Verifies the access list configuration.
Step 4	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no access-list** *access-list-number* global configuration command to delete the entire access list. You cannot delete individual ACEs from numbered access lists.

After an ACL is created, any additions (possibly entered from the terminal) are placed at the end of the list. You can add ACEs to an ACL, but deleting any ACE deletes the entire ACL.

Note When creating an ACL, remember that, by default, the end of the access list contains an implicit deny statement for all packets if it did not find a match before reaching the end.

After creating an ACL, you must apply it to an interface, as described in the "Applying the ACL to an Interface" section on page 85.

Creating Named Standard and Extended ACLs

You can identify IP ACLs with an alphanumeric string (a name) rather than a number. You can use named ACLs to configure more IP access lists on a switch than if you use numbered access lists. If you identify your access list with a name rather than a number, the mode and command syntax are slightly different. However, not all commands that use IP access lists accept a named ACL.

Note The name you give to a standard ACL or extended ACL can also be a number in the supported range of access list numbers. That is, the name of a standard IP ACL can be 1 to 99; the name of an extended IP ACL can be 100 to 199. The advantage of using named ACLs instead of numbered lists is that you can delete individual entries from a named list.

Consider these guidelines and limitations before configuring named ACLs:

- A standard ACL and an extended ACL cannot have the same name.
- Numbered ACLs are also available, as described in the "Creating Standard and Extended IP ACLs" section on page 78.

Beginning in privileged EXEC mode, follow these steps to create a standard access list using names:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip access-list standard** {*name* | *access-list-number*}	Defines a standard IP access list using a name, and enter access-list configuration mode. **Note** The name can be a number from 1 to 99.

Step 3	**deny** {*source source-wildcard* | **host** *source* | **any**}
	or
	permit {*source source-wildcard* | **host** *source* | **any**}
		In access-list configuration mode, specifies one or more conditions denied or permitted to determine if the packet is forwarded or dropped.
		• **host** *source* represents a source and source wildcard of *source* 0.0.0.0.
		• **any** represents a source and source wildcard of 0.0.0.0 255.255.255.255.
		Note The **log** option is not supported on Ethernet switch interfaces.

Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show access-lists** [*number* | *name*]	Displays the access list configuration.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Beginning in privileged EXEC mode, follow these steps to create an extended ACL using names:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip access-list extended** {*name* | *access-list-number*}	Defines an extended IP access list by using a name, and enter access-list configuration mode. **Note** The name can be a number from 100 to 199.
Step 3	{**deny** | **permit**} *protocol* {*source source-wildcard* | **host** *source* | **any**} [*operator port*] {*destination destination-wildcard* | **host** *destination* | **any**} [*operator port*]	In access-list configuration mode, specifies the conditions allowed or denied. See the "Creating a Numbered Extended ACL" section on page 80 for definitions of protocols and other keywords. • **host** *source* represents a source and source wildcard of *source* 0.0.0.0, and **host** *destination* represents a destination and destination wildcard of *destination* 0.0.0.0. • **any** represents a source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show access-lists** [*number* | *name*]	Displays the access list configuration.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

When making the standard and extended ACL, remember that, by default, the end of the ACL contains an implicit deny statement for everything if it did not find a match before reaching the end. For standard ACLs, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask.

After you create an ACL, any additions are placed at the end of the list. You cannot selectively add ACEs to a specific ACL. However, you can use **no permit** and **no deny** commands to remove ACEs from a named ACL. Being able to selectively remove lines from a named ACL is one reason you might use named ACLs instead of numbered ACLs.

After creating an ACL, you must apply it to a line or interface, as described in the "Applying the ACL to an Interface" section on page 85.

Including Comments About Entries in ACLs

You can use the **remark** command to include comments (remarks) about entries in any IP standard or extended ACL. The remarks make the ACL easier for you to understand and scan. Each remark line is limited to 100 characters.

The remark can go before or after a permit or deny statement. You should be consistent about where you put the remark so that it is clear which remark describes which permit or deny statement. For example, it would be confusing to have some remarks before the associated permit or deny statements and some remarks after the associated statements.

For IP numbered standard or extended ACLs, use the **access-list** *access-list number* **remark** *remark* global configuration command to include a comment about an access list. To remove the remark, use the **no** form of this command.

For an entry in a named IP ACL, use the **remark** *access-list* global configuration command. To remove the remark, use the **no** form of this command.

Applying the ACL to an Interface

After you create an ACL, you can apply it to one or more interfaces. ACLs can be applied on inbound interfaces. This section describes how to accomplish this task for network interfaces. Note these guidelines:

- When controlling access to a line, you must use a number. Numbered ACLs can be applied to lines.
- When controlling access to an interface, you can use a name or number.

Beginning in privileged EXEC mode, follow these steps to control access to a Layer 2 or Layer 3 interface:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Identifies a specific interface for configuration and enter interface configuration mode. The interface must be a Layer 2 interface or routed port.
Step 3	**ip access-group** {*access-list-number* | *name*} {**in**}	Controls access to the specified interface.
Step 4	**end**	Returns to privileged EXEC mode.

	Command	Purpose
Step 5	**show running-config**	Displays the access list configuration.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Note The **ip access-group** interface configuration command is only valid when applied to a Layer 2 interface or a Layer 3 interface. If applied to a Layer 3 interface, the interface must have been configured with an IP address. ACLs cannot be applied to interface port-channels.

For inbound ACLs, after receiving a packet, the switch checks the packet against the ACL. If the ACL permits the packet, the switch continues to process the packet. If the ACL rejects the packet, the switch discards the packet.

When you apply an undefined ACL to an interface, the switch acts as if the ACL has not been applied to the interface and permits all packets. Remember this behavior if you use undefined ACLs for network security.

Displaying ACLs

You can display existing ACLs by using **show** commands.

Beginning in privileged EXEC mode, follow these steps to display access lists:

	Command	Purpose
Step 1	**show access-lists** [*number* | *name*]	Displays information about all IP access lists or about a specific access list (numbered or named).
Step 2	**show ip access-list** [*number* | *name*]	Displays information about all IP address access lists or about a specific IP ACL (numbered or named).

Configuring Quality of Service (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.

This section describes how to configure QoS on your Ethernet switch network module:

- Understanding the Default QoS Configuration, page 87
- Configuring Classification Using Port Trust States, page 87
- Configuring a QoS Policy, page 90
- Configuring CoS Maps, page 96
- Displaying QoS Information, page 97

Understanding the Default QoS Configuration

- The default port CoS value is 0.
- The default port trust state is untrusted.
- No policy maps are configured.
- No policers are configured.
- The default CoS-to-DSCP map is shown in Table 13 on page 96.
- The default DSCP-to-CoS map is shown in Table 14 on page 97.

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, policing, mapping, and queueing on the individual physical ports that comprise the EtherChannel. You must decide whether the QoS configuration should match on all ports in 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.
- Control traffic (such as spanning-tree Bridge Protocol Data Units (BPDUs) and routing update packets) received by the switch are subject to all ingress QoS processing.
- Only one ACL per class map and only one **match** command per class map are supported. The ACL can have multiple access control entries, which are commands that match fields against the contents of the packet.
- Policy maps with ACL classification in the egress direction are not supported and cannot be attached to an interface by using the **service-policy input** *policy-map-name* interface configuration command.
- In a policy map, the class named class-default is not supported. The switch does not filter traffic based on the policy map defined by the **class class-default** policy-map configuration command.

For more information on guidelines for configuring ACLs, see the "Classification Based on QoS ACLs" section on page 32.

Configuring Classification Using Port Trust States

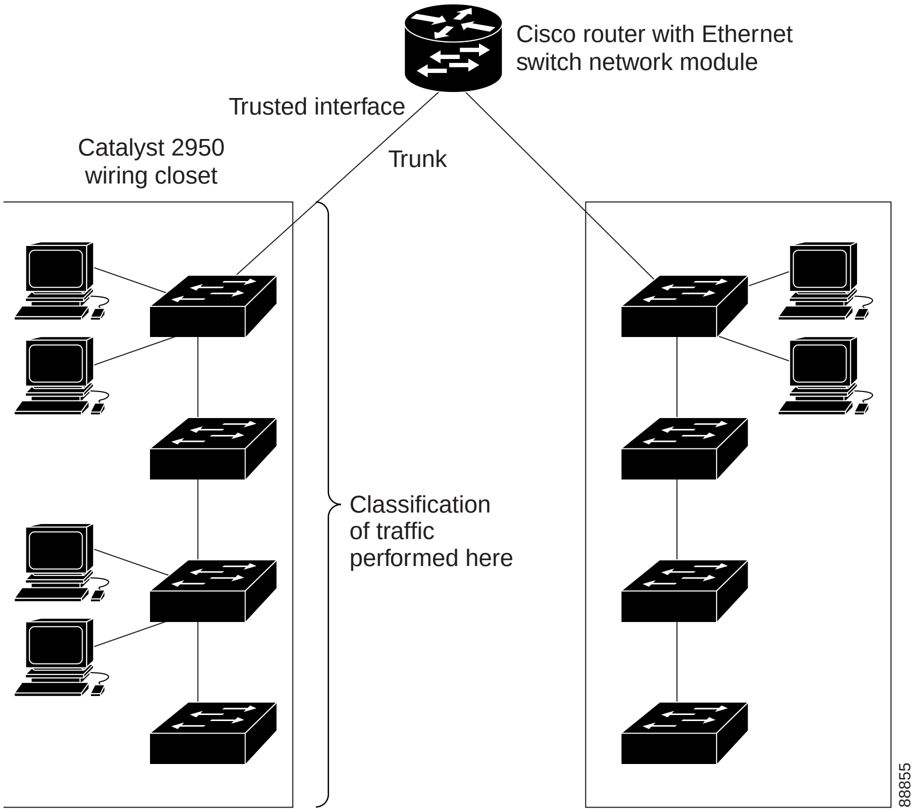
This section describes how to classify incoming traffic by using port trust states:

- Configuring the Trust State on Ports and SVIs within the QoS Domain, page 87
- Configuring the CoS Value for an Interface, page 89

Configuring the Trust State on Ports and SVIs within the QoS Domain

Packets entering a QoS domain are classified at the edge of the QoS domain. When the packets are classified at the edge, the switch port within the QoS domain can be configured to one of the trusted states because there is no need to classify the packets at every switch within the QoS domain. Figure 20 shows a sample network topology.

Figure 20 Port Trusted States within the QoS Domain


Cisco router with Ethernet switch network module
Trusted interface
Catalyst 2950 wiring closet
Trunk
Classification of traffic performed here
88855

Beginning in privileged EXEC mode, follow these steps to configure the port to trust the classification of the traffic that it receives:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode and specify the interface to be trusted. Valid interfaces include physical interfaces and SVIs.

	Command	Purpose
Step 3	**mls qos trust** {**cos** | **dscp**}	Configures the port trust state. By default, the port is not trusted. Use the **cos** keyword setting if your network is composed of Ethernet LANs, Catalyst 2950 switches, and has no more than two types of traffic. Use the **dscp** keyword if your network is not composed of only Ethernet LANs and if you are familiar with sophisticated QoS features and implementations. Enter the **cos** keyword if you want ingress packets to be classified with the packet CoS values. For tagged IP packets, the DSCP value of the packet is modified based on the CoS-to-DSCP map. The egress queue assigned to the packet is based on the packet CoS value. Enter the **dscp** keyword if you want ingress packets to be classified with packet DSCP values. For non-IP packets, the packet CoS value is used for tagged packets; the default port CoS is used for untagged packets. Internally, the switch modifies the CoS value by using the DSCP-to-CoS map. Enter the **dscp** keyword if you are using an SVI that is a VLAN interface that you created by using the **interface vlan** *vlan-id* global configuration command. The DCSP-to-CoS map will be applied to packets arriving from a router to the Ethernet switch network module through an SVI.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show mls qos interface** [*interface-id*] [**policers**]	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return a port to its untrusted state, use the **no mls qos trust** interface configuration command.

For information on how to change the default CoS value, see the "Configuring the CoS Value for an Interface" section on page 89. For information on how to configure the CoS-to-DSCP map, see the "Configuring the CoS-to-DSCP Map" section on page 96.

Configuring the CoS Value for an Interface

QoS assigns the CoS value specified with the **mls qos cos** interface configuration command to untagged frames received on trusted and untrusted ports.

Beginning in privileged EXEC mode, follow these steps to define the default CoS value of a port or to assign the default CoS to all incoming packets on the port:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specify the interface to be trusted. Valid interfaces include physical interfaces.

	Command	Purpose
Step 3	**mls qos cos** {*default-cos* | **override**}	Configures the default CoS value for the port. For *default-cos*, specify a default CoS value to be assigned to a port. If the port is CoS trusted and packets are untagged, the default CoS value becomes the CoS value for the packet. The CoS range is 0 to 7. The default is 0. Use the **override** keyword to override the previously configured trust state of the incoming packets and to apply the default port CoS value to all incoming packets. By default, CoS override is disabled. Use the **override** keyword when all incoming packets on certain ports deserve higher priority than packets entering from other ports. Even if a port was previously set to trust DSCP, this command overrides the previously configured trust state, and all the incoming CoS values are assigned the default CoS value configured with this command. If an incoming packet is tagged, the CoS value of the packet is modified with the default CoS of the port at the ingress port.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show mls qos interface**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return to the default setting, use the **no mls qos cos** {*default-cos* | **override**} interface configuration command.

Note The **mls qos cos** command replaced the **switchport priority** command in Cisco IOS Release 12.1(6)EA2.

Configuring a QoS Policy

Configuring a QoS policy typically requires classifying traffic into classes, configuring policies applied to those traffic classes, and attaching policies to interfaces.

For background information, see the "Classification" section on page 32 and the "Policing and Marking" section on page 34.

This section contains this configuration information:

- Classifying Traffic by Using ACLs, page 91
- Classifying Traffic by Using Class Maps, page 93
- Classifying, Policing, and Marking Traffic by Using Policy Maps, page 94

Classifying Traffic by Using ACLs

You can classify IP traffic by using IP standard or IP extended ACLs.

Beginning in privileged EXEC mode, follow these steps to create an IP standard ACL for IP traffic:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit** | **remark**} {*source source-wildcard* | **host** *source* | **any**}	Creates an IP standard ACL, repeating the command as many times as necessary. For *access-list-number*, enter the ACL number. The range is 1 to 99 and 1300 to 1999. Enter **deny** or **permit** to specify whether to deny or permit access if The *source* is the source address of the network or host from which the packet is being sent, specified in one of three ways: • The 32-bit quantity in dotted-decimal format. • The keyword **any** as an abbreviation for *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. You do not need to enter a source-wildcard. • The keyword **host** as an abbreviation for source and source-wildcard of *source* 0.0.0.0. (Optional) The *source-wildcard* applies wildcard bits to the source (see first bullet item). **Note** Deny statements are not supported for QoS ACLS. See the "Classification Based on QoS ACLs" section on page 32 for more details.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show access-lists**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To delete an ACL, use the **no access-list** *access-list-number* global configuration command.

Beginning in privileged EXEC mode, follow these steps to create an IP extended ACL for IP traffic:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit** | **remark**} *protocol* {*source source-wildcard* | **host** *source* | **any**}[*operator port*] {*destination destination-wildcard* | **host** *destination* | **any**} [*operator port*]	Creates an IP extended ACL, repeating the command as many times as necessary. For *access-list-number*, enter the ACL number. The range is 100 to 199 and 2000 to 2699. Enter **deny** or **permit** to specify whether to deny or permit access if conditions are matched. For *protocol*, enter the name or number of an IP protocol. Use the question mark (?) to see a list of available protocol keywords. For *source*, enter the network or host from which the packet is being sent. You specify this by using dotted decimal notation by using the **any** keyword as an abbreviation for *source* 0.0.0.0 *source-wildcard* 255.255.255.255, or by using the **host** keyword for *source* 0.0.0.0. For *source-wildcard*, enter the wildcard bits by placing ones in the bit positions that you want to ignore. You specify the wildcard by using dotted decimal notation, by using the **any** keyword as an abbreviation for *source* 0.0.0.0 *source-wildcard* 255.255.255.255, or by using the **host** keyword for *source* 0.0.0.0. For *destination*, enter the network or host to which the packet is being sent. You have the same options for specifying the *destination* and *destination-wildcard* as those described by *source* and *source-wildcard*. Defines a destination or source port. • The *operator* can be only **eq** (equal). • If operator is after *source source-wildcard*, conditions match when the source port matches the defined port. • If operator is after *destination destination-wildcard*, conditions match when the destination port matches the defined port. • The *port* is a decimal number or name of a TCP or UDP port. The number can be from 0 to 65535. • Use TCP port names only for TCP traffic. • Use UDP port names only for UDP traffic. **Note** Deny statements are not supported for QoS ACLS. See the "Classification Based on QoS ACLs" section on page 32 for more details.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show access-lists**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To delete an ACL, use the **no access-list** *access-list-number* global configuration command.

Classifying Traffic by Using Class Maps

You use the **class-map** global configuration command to isolate a specific traffic flow (or class) from all other traffic and to name it. The class map defines the criteria to use to match against a specific traffic flow to further classify it. Match statements can include criteria such as an ACL. The match criterion is defined with one match statement entered within the class-map configuration mode.

Note You can also create class maps during policy map creation by using the **class** policy-map configuration command. For more information, see the "Classifying, Policing, and Marking Traffic by Using Policy Maps" section on page 94.

Beginning in privileged EXEC mode, follow these steps to create a class map and to define the match criterion to classify traffic:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} {*source source-wildcard* | **host** *source* | **any**} or **access-list** *access-list-number* {**deny** | **permit** | **remark**} *protocol* {*source source-wildcard* | **host** *source* | **any**} [*operator port*] {*destination destination-wildcard* | **host** *destination* | **any**} [*operator port*]	Creates an IP standard or extended ACL for IP traffic, repeating the command as many times as necessary. For more information, see the "Classifying Traffic by Using ACLs" section on page 91. **Note** Deny statements are not supported for QoS ACLS. See the "Classification Based on QoS ACLs" section on page 32 for more details.
Step 3	**class-map** *class-map-name*	Creates a class map, and enter class-map configuration mode. By default, no class maps are defined. For *class-map-name*, specify the name of the class map.
Step 4	**match access-group** *acl-index-or-name*	Defines the match criterion to classify traffic. By default, no match criterion is supported. Only one match criterion per class map is supported, and only one ACL per class map is supported. For **access-group** *acl-index-or-name*, specify the number or name of the ACL created in Step 3.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show class-map** [*class-map-name*]	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To delete an existing class map, use the **no class-map** *class-map-name* global configuration command. To remove a match criterion, use the **no match access-group** *acl-index-or-name* class-map configuration command.

Classifying, Policing, and Marking Traffic by Using Policy Maps

A policy map specifies which traffic class to act on. Actions can include trusting the CoS or DSCP values in the traffic class; setting a specific DSCP value in the traffic class; and specifying the traffic bandwidth limitations for each matched traffic class (policer) and the action to take when the traffic is out of profile (marking).

A separate policy-map class can exist for each type of traffic received through an interface. You can attach only one policy map per interface in the input direction.

Beginning in privileged EXEC mode, follow these steps to create a policy map:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} {*source source-wildcard* | **host** *source* | **any**} or **access-list** *access-list-number* {**deny** | **permit** | **remark**} *protocol* {*source source-wildcard* | **host** *source* | **any**}[*operator port*] {*destination destination-wildcard* | **host** *destination* | **any}** [*operator port*]	Creates an IP standard or extended ACL for IP traffic, repeating the command as many times as necessary. For more information, see the "Classifying Traffic by Using ACLs" section on page 91. **Note** Deny statements are not supported for QoS ACLS. See the "Classification Based on QoS ACLs" section on page 32 for more details.
Step 3	**policy-map** *policy-map-name*	Creates a policy map by entering the policy map name, and enter policy-map configuration mode. By default, no policy maps are defined. The default behavior of a policy map is to set the DSCP to 0 if the packet is an IP packet and to set the CoS to 0 if the packet is tagged. No policing is performed.
Step 4	**class** *class-map-name* [**access-group** *acl-index-or-name*]	Defines a traffic classification, and enter policy-map class configuration mode. By default, no policy map class maps are defined. If a traffic class has already been defined by using the **class-map** global configuration command, specify its name for *class-map-name* in this command. For **access-group** *acl-index-or-name*, specify the number or name of the ACL created in Step 2. **Note** In a policy map, the class named class-default is not supported. The switch does not filter traffic based on the policy map defined by the **class class-default** policy-map configuration command.

	Command	Purpose
Step 5	**police** {*bps* | **cir** *bps*} [*burst-byte* | **bc** *burst-byte*] **conform-action transmit** [**exceed-action** {**drop** | **dscp** *dscp-value*}]	Defines a policer for the classified traffic. You can configure up to 60 policers on ingress Gigabit-capable Ethernet ports and up to 6 policers on ingress 10/100 Ethernet ports. For *bps*, specify average traffic rate or committed information rate in bits per second (bps). The range is 1 Mbps to 100 Mbps for 10/100 Ethernet ports and 8 Mbps to 1000 Mbps for the Gigabit-capable Ethernet ports. For *burst-byte*, specify the normal burst size or burst count in bytes. (Optional) Specify the action to take when the rates are exceeded. Use the **exceed-action drop** keywords to drop the packet. Use the **exceed-action dscp** *dscp-value* keywords to mark down the DSCP value and transmit the packet.
Step 6	**exit**	Returns to policy-map configuration mode.
Step 7	**exit**	Returns to global configuration mode.
Step 8	**interface** *interface-id*	Enters interface configuration mode, and specify the interface to attach to the policy map. Valid interfaces include physical interfaces.
Step 9	**service-policy input** *policy-map-name*	Applies a policy map to the input of a particular interface. Only one policy map per interface per direction is supported. Use **input** *policy-map-name* to apply the specified policy map to the input of an interface.
Step 10	**end**	Returns to privileged EXEC mode.
Step 11	**show policy-map** [*policy-map-name* **class** *class-name*]	Verifies your entries.
Step 12	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class map, use the **no class** *class-map-name* policy-map configuration command. To remove an existing policer, use the **no police** *rate-bps burst-byte* [**exceed-action** {**drop** | **dscp** *dscp-value*}] policy-map configuration command. To remove the policy map and interface association, use the **no service-policy input** *policy-map-name* interface configuration command.

Configuring CoS Maps

This section describes how to configure the DSCP maps:

- Configuring the CoS-to-DSCP Map, page 96
- Configuring the DSCP-to-CoS Map, page 96

All the maps are globally defined.

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 13 shows the default CoS-to-DSCP map.

Table 13 Default CoS-to-DSCP Map

CoS value	0	1	2	3	4	5	6	7
DSCP value	0	8	16	26	32	46	48	56

If these values are not appropriate for your network, you need to modify them. These CoS-to-DSCP mapping numbers follow the numbers used in deploying Cisco AVVID and may be different from the mapping numbers used by the Catalyst 2950, Catalyst 3550, and other switches.

Beginning in privileged EXEC mode, follow these steps to modify the CoS-to-DSCP map:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**mls qos map cos-dscp** *dscp1...dscp8*	Modifies the CoS-to-DSCP map. For *dscp1...dscp8*, enter 8 DSCP values that correspond to CoS values 0 to 7. Separate each DSCP value with a space. The supported DSCP values are 0, 8, 10, 16, 18, 24, 26, 32, 34, 40, 46, 48, and 56.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show mls qos maps cos-dscp**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return to the default map, use the **no mls qos map cos-dscp** global configuration command.

Configuring the DSCP-to-CoS Map

You use the DSCP-to-CoS map to map DSCP values in incoming packets to a CoS value, which is used to select one of the four egress queues.

The Ethernet switch network modules support these DSCP values: 0, 8, 10, 16, 18, 24, 26, 32, 34, 40, 46, 48, and 56.

Table 14 shows the default DSCP-to-CoS map.

Table 14 Default DSCP-to-CoS Map

DSCP values	0	8, 10	16, 18	24, 26	32, 34	40, 46	48	56
CoS values	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:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**mls qos map dscp-cos** *dscp-list* **to** *cos*	Modifies the DSCP-to-CoS map. For *dscp-list*, enter up to 13 DSCP values separated by spaces. Then enter the **to** keyword. For *cos*, enter the CoS value to which the DSCP values correspond. The supported DSCP values are 0, 8, 10, 16, 18, 24, 26, 32, 34, 40, 46, 48, and 56. The CoS range is 0 to 7.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show mls qos maps dscp-to-cos**	Verifies your entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To return to the default map, use the **no mls qos map dscp-cos** global configuration command.

Displaying QoS Information

To display the current QoS information, use one or more of the privileged EXEC commands in Table 15:

Table 15 Commands for Displaying QoS Information

Command	Purpose
show class-map [*class-map-name*]	Displays QoS class maps, which define the match criteria to classify traffic.
show policy-map [*policy-map-name* [**class** *class-name*]]	Displays QoS policy maps, which define classification criteria for incoming traffic.
show mls qos maps [**cos-dscp** | **dscp-cos**]	Displays QoS mapping information. Maps are used to generate an internal DSCP value, which represents the priority of the traffic.
show mls qos interface [*interface-id*] [**policers**]	Displays QoS information at the interface level.
show mls masks [**qos** | **security**]	Displays details regarding the masks[1] used for QoS and security ACLs.

[1] Access Control Parameters are called masks in the switch CLI commands and output.

Configuring Power Management on the Interface

To manage the powering of the Cisco IP phones, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# int fastethernet port/slot`	Selects a particular Fast Ethernet interface for configuration.
Step 3	`Router(config)# power inline {auto | never}`	Configures the port to supply inline power automatically to a Cisco IP phone. Use **never** to permanently disable inline power on the port.

Verifying Power Management on the Interface

Step 1 Use the **show power inline** command to verify the power configuration on the ports:

```
Router# show power inline

PowerSupply   SlotNum.   Maximum   Allocated       Status
-----------   --------   -------   ---------       ------
 EXT-PS          1       165.000    20.000         PS1 GOOD PS2 ABSENT

Interface           Config   Phone     Powered    PowerAllocated
---------           ------   -----     -------    --------------
FastEthernet1/0      auto    no          off       0.000 Watts
FastEthernet1/1      auto    no          off       0.000 Watts
FastEthernet1/2      auto    no          off       0.000 Watts
FastEthernet1/3      auto    no          off       0.000 Watts
FastEthernet1/4      auto    unknown     off       0.000 Watts
FastEthernet1/5      auto    unknown     off       0.000 Watts
FastEthernet1/6      auto    unknown     off       0.000 Watts
FastEthernet1/7      auto    unknown     off       0.000 Watts
FastEthernet1/8      auto    unknown     off       0.000 Watts
FastEthernet1/9      auto    unknown     off       0.000 Watts
FastEthernet1/10     auto    unknown     off       0.000 Watts
FastEthernet1/11     auto    yes         on        6.400 Watts
FastEthernet1/12     auto    yes         on        6.400 Watts
FastEthernet1/13     auto    no          off       0.000 Watts
FastEthernet1/14     auto    unknown     off       0.000 Watts
FastEthernet1/15     auto    unknown     off       0.000 Watts
```

Configuring IP Multicast Layer 3 Switching

These sections describe how to configure IP multicast Layer 3 switching:

- Enabling IP Multicast Routing Globally, page 99
- Enabling IP PIM on Layer 3 Interfaces, page 99
- Verifying IP Multicast Layer 3 Hardware Switching Summary, page 100
- Verifying the IP Multicast Routing Table, page 101

Enabling IP Multicast Routing Globally

You must enable IP multicast routing globally before you can enable IP multicast Layer 3 switching on Layer 3 interfaces.

For complete information and procedures, refer to these publications:

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

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/
- *Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*, Release 12.2 at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipras_r/index.htm
- *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2 at this URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fiprrp_r/index.htm
- *Cisco IOS IP Command Reference, Volume 3 of 3: Multicast*, Release 12.2 at this URL:

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

To enable IP multicast routing globally, Use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **`ip multicast-routing`**	Enables IP multicast routing globally.

Enabling IP PIM on Layer 3 Interfaces

You must enable PIM on the Layer 3 interfaces before IP multicast Layer 3 switching functions on those interfaces.

To enable IP PIM on a Layer 3 interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface vlan`** *`vlan-id`* `slot/port`	Selects the interface to be configured.
Step 2	`Router(config-if)#` **`ip pim`** {**`dense-mode`** | **`sparse-mode`** | **`sparse-dense-mode`**}	Enables IP PIM on a Layer 3 interface.

The following example shows how to enable PIM on an interface using the default mode (**sparse-dense-mode**):

```
Router(config-if)# ip pim
Router(config-if)#
```

The following example shows how to enable PIM sparse mode on an interface:

```
Router(config-if)# ip pim sparse-mode
Router(config-if)#
```

Verifying IP Multicast Layer 3 Hardware Switching Summary

Note The **show interface statistics** command does not verify hardware-switched packets, only packets switched by software.

The **show ip pim interface count** command verifies the IP multicast Layer 3 switching enable state on IP PIM interfaces and the number of packets received and sent on the interface.

Use the following **show** commands to verify IP multicast Layer 3 switching information for an IP PIM Layer 3 interface:

Step 1

```
Router# show ip pim interface count

State:* - Fast Switched, D - Distributed Fast Switched
      H - Hardware Switching Enabled
Address          Interface          FS  Mpackets In/Out
10.15.1.20       GigabitEthernet4/8 * H 952/4237130770
10.20.1.7        GigabitEthernet4/9 * H 1385673757/34
10.25.1.7        GigabitEthernet4/10* H 0/34
10.11.1.30       FastEthernet6/26   * H 0/0
10.37.1.1        FastEthernet6/37   * H 0/0
1.22.33.44       FastEthernet6/47   * H 514/68
```

Step 2

```
Router# show ip mroute count

IP Multicast Statistics
56 routes using 28552 bytes of memory
13 groups, 3.30 average sources per group
Forwarding Counts:Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts:Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group:224.2.136.89, Source count:1, Group pkt count:29051
  Source:132.206.72.28/32, Forwarding:29051/-278/1186/0, Other:85724/8/56665
Router#
```

Note The negative counter means that the outgoing interface list of the corresponding entry is NULL, and this indicates that this flow is still active.

Step 3

```
Router# show ip interface vlan 10

Vlan10 is up, line protocol is up
  Internet address is 10.0.0.6/8
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.1 224.0.0.2 224.0.0.13 224.0.0.10
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are never sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP Fast switching turbo vector
  IP Normal CEF switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
IP multicast multilayer switching is enabled
IP mls switching is enabled
Router#
```

Verifying the IP Multicast Routing Table

Step 1 Use the **show ip mroute** command to verify the IP multicast routing table:

```
Router# show ip mroute 230.13.13.1

IP Multicast Routing Table
Flags:D - Dense, S - Sparse, s - SSM Group, C - Connected, L - Local,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP, U - URD, I - Received Source Specific Host
           Report
Outgoing interface flags:H - Hardware switched
Timers:Uptime/Expires
Interface state:Interface, Next-Hop or VCD, State/Mode

(*, 230.13.13.1), 00:16:41/00:00:00, RP 10.15.1.20, flags:SJC
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20
  Outgoing interface list:
 GigabitEthernet4/9, Forward/Sparse-Dense, 00:16:41/00:00:00, H

(*, 230.13.13.2), 00:16:41/00:00:00, RP 10.15.1.20, flags:SJC
 Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/9, Forward/Sparse-Dense, 00:16:41/00:00:00, H

(10.20.1.15, 230.13.13.1), 00:14:31/00:01:40, flags:CJT
 Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
  Outgoing interface list:
    GigabitEthernet4/9, Forward/Sparse-Dense, 00:14:31/00:00:00, H
(132.206.72.28, 224.2.136.89), 00:14:31/00:01:40, flags:CJT
  Incoming interface:GigabitEthernet4/8, RPF nbr 10.15.1.20, RPF-MFD
 Outgoing interface list:Null
Router#
```

Note The RPF-MFD flag indicates that the flow is completely hardware switched. The H flag indicates that the flow is hardware-switched on the outgoing interface.

Configuring IGMP Snooping

This section describes how to configure IGMP snooping on your router and consists of the following configuration information and procedures:

- Enabling or Disabling IGMP Snooping, page 102
- Enabling IGMP Immediate-Leave Processing, page 103
- Statically Configuring an Interface to Join a Group, page 103
- Configuring a Multicast Router Port, page 104

Enabling or Disabling IGMP Snooping

By default, IGMP snooping is globally enabled on the Ethernet switch network module. When globally enabled or disabled, it is also enabled or disabled in all existing VLAN interfaces. By default, IGMP snooping is enabled on all VLANs, but it can be enabled and disabled on a per-VLAN basis.

Global IGMP snooping overrides the per-VLAN IGMP snooping capability. If global snooping is disabled, you cannot enable VLAN snooping. If global snooping is enabled, you can enable or disable snooping on a VLAN basis.

Beginning in privileged EXEC mode, follow these steps to globally enable IGMP snooping on the Ethernet switch network module:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip igmp snooping**	Globally enables IGMP snooping in all existing VLAN interfaces.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show ip igmp snooping**	Displays snooping configuration.
Step 5	**copy running-config startup-config**	(Optional) Saves your configuration to the startup configuration.

To globally disable IGMP snooping on all VLAN interfaces, use the **no ip igmp snooping** global command.

Beginning in privileged EXEC mode, follow these steps to enable IGMP snooping on a VLAN interface:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip igmp snooping vlan** *vlan-id*	Enables IGMP snooping on the VLAN interface.
Step 3	**end**	Returns to privileged EXEC mode.

	Command	Purpose
Step 4	**show ip igmp snooping** [**vlan** *vlan-id*]	Displays snooping configuration. (Optional) *vlan-id* is the number of the VLAN.
Step 5	**copy running-config startup-config**	(Optional) Saves your configuration to the startup configuration.

To disable IGMP snooping on a VLAN interface, use the **no ip igmp snooping vlan** *vlan-id* global configuration command for the specified VLAN number (for example, vlan1).

Enabling IGMP Immediate-Leave Processing

When you enable IGMP Immediate-Leave processing, the Ethernet switch network module immediately removes a port from the IP multicast group when it detects an IGMP version 2 leave message on that port. Immediate-Leave processing allows the switch to remove an interface that sends a leave message from the forwarding table without first sending out group-specific queries to the interface. You should use the Immediate-Leave feature only when there is only a single receiver present on every port in the VLAN.

Beginning in privileged EXEC mode, follow these steps to enable IGMP Immediate-Leave processing:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip igmp snooping vlan** *vlan-id* **immediate-leave**	Enables IGMP Immediate-Leave processing on the VLAN interface.
Step 3	**end**	Returns to privileged EXEC mode.

To disable Immediate-Leave processing, follow Steps 1 and 2 to enter interface configuration mode, and use the **no ip igmp snooping vlan** *vlan-id* **immediate-leave** global configuration command.

Statically Configuring an Interface to Join a Group

Ports normally join multicast groups through the IGMP report message, but you can also statically configure a host on an interface.

Beginning in privileged EXEC mode, follow these steps to add a port as a member of a multicast group:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode
Step 2	**ip igmp snooping vlan** *vlan-id* **static** *mac-address* **interface** *interface-id*	Statically configures a port as a member of a multicast group: • *vlan-id* is the multicast group VLAN ID. • *mac-address* is the group MAC address. • *interface-id* is the member port.
Step 3	**end**	Returns to privileged EXEC mode.

	Command	Purpose
Step 4	**show mac-address-table multicast** [**vlan** *vlan-id*] [**user** | **igmp-snooping**] [**count**]	Displays MAC address table entries for a VLAN. • *vlan-id* is the multicast group VLAN ID. • **user** displays only the user-configured multicast entries. • **igmp-snooping** displays entries learned via IGMP snooping. • **count** displays only the total number of entries for the selected criteria, not the actual entries.
Step 5	**copy running-config startup-config**	(Optional) Saves your configuration to the startup configuration.

Configuring a Multicast Router Port

Beginning in privileged EXEC mode, follow these steps to enable a static connection to a multicast router:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip igmp snooping vlan** *vlan-id* **mrouter** {**interface** *interface-id* | **learn pim-dvmrp**}	Specify the multicast router VLAN ID (1 to 1001). Specify the interface to the multicast router.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show ip igmp snooping** [**vlan** *vlan-id*]	Verifies that IGMP snooping is enabled on the VLAN interface.
Step 5	**show ip igmp snooping mrouter** [**vlan** *vlan-id*]	Displays information on dynamically learned and manually configured multicast router interfaces.
Step 6	**copy running-config startup-config**	(Optional) Saves your configuration to the startup configuration.

Configuring Global Storm-Control

This section describes how to configure global storm-control and characteristics on your router and consists of the following configuration procedures:

- Enabling Global Storm-Control, page 105
- Verifying Global Storm-Control, page 105

By default, unicast, broadcast, and multicast suppression is disabled on the switch.

Enabling Global Storm-Control

Enable **global storm-control** globally and enter the percentage of total available bandwidth that you want to be used by all traffic (multicast, unicast,); entering 100 percent would allow all traffic.

To enable a particular type of global storm-control, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` [**`no`**] **`storm-control broadcast level`** *`level`*	Specifies the broadcast suppression level for an interface as a percentage of total bandwidth. A threshold value of 100 percent means that no limit is placed on broadcast traffic. Use the **no** keyword to restore the defaults.
Step 3	`Router(config)#` [**`no`**] **`storm-control multicast level`** *`level`*	Specifies the multicast suppression level for an interface as a percentage of total bandwidth. Use the **no** keyword to restore the defaults.
Step 4	`Router(config)#` [**`no`**] **`storm-control unicast level`** *`level`*	Specifies the unicast suppression level for an interface as a percentage of total bandwidth. Use the **no** keyword to restore the defaults.
Step 5	`Router(config)#` **`end`**	Returns to privileged EXEC mode.

Verifying Global Storm-Control

Step 1 Use the **show storm-control** command to view switchport characteristics, including storm-control levels set on the interface:

```
Router# show storm-control
```

Step 2 Use the **show interface counters** privileged EXEC commands to display the count of discarded packets.

To verify global storm-control statistics on an interface, use the following commands beginning in privileged EXEC mode:

Command	Purpose
`show interface` [*`interface-id`*] **`counters broadcast`**	Verifies the broadcast suppression discard counter for all interfaces or a specific interface. Verifies the number of packets discarded.
`show interface` [*`interface-id`*] **`counters multicast`**	Verifies the multicast suppression discard counter for all interfaces or a specific interface. Verifies the number of packets discarded.
`show interface` [*`interface-id`*] **`counters unicast`**	Verifies the unicast suppression discard counter for all interfaces or a specific interface. Verifies the number of packets discarded.

The following is sample output from the **show interface counters broadcast** privileged EXEC command:

```
Router# show interface counters broadcast

Port      BcastSuppDiscards
Fa0/1                     0
Fa0/2                     0
```

Configuring Per-Port Storm-Control

You can use these techniques to block the forwarding of unnecessary flooded traffic. This section describes how to configure per-port storm-control and characteristics on your router and consists of the following configuration procedures:

- Enabling Per-Port Storm-Control, page 106
- Disabling Per-Port Storm-Control, page 107

By default, unicast, broadcast, and multicast suppression is disabled.

Enabling Per-Port Storm-Control

Beginning in privileged EXEC mode, follow these steps to enable per-port storm-control:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface*	Enters interface configuration mode, and enter the port to configure.
Step 3	**storm-control** {**broadcast** | **multicast** | **unicast**} **level** *level* [*level-low*]	Configures broadcast, multicast, or unicast per-port storm-control. Specify the rising threshold level for either broadcast, multicast, or unicast traffic. The storm control action occurs when traffic utilization reaches this level. (Optional) Specify the falling threshold level. The normal transmission restarts (if the action is filtering) when traffic drops below this level.
Step 4	**storm-control action shutdown**	Selects the **shutdown** keyword to disable the port during a storm. The default is to filter out the traffic.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show storm-control** [**interface**] [{**broadcast** | **multicast** | **unicast** | **history**}]	Verifies your entries.

Disabling Per-Port Storm-Control

Beginning in privileged EXEC mode, follow these steps to disable per-port storm-control:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface*	Enters interface configuration mode, and enter the port to configure.
Step 3	**no storm-control** {**broadcast** | **multicast** | **unicast**} **level**	Disables per-port storm control.
Step 4	**no storm-control action shutdown**	Disables the specified storm control action.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show storm-control** {**broadcast** | **multicast** | **unicast**}	Verifies your entries.

Configuring Separate Voice and Data Subnets

For ease of network administration and increased scalability, network managers can configure the Ethernet switch network module to support Cisco IP phones such that the voice and data traffic reside on separate subnets. You should always use separate VLANs when you are able to segment the existing IP address space of your branch office.

User priority bits in the 802.1p portion of the 802.1Q standard header are used to provide prioritization in Ethernet switches. This is a vital component in designing Cisco AVVID networks.

The Ethernet switch network module provides the performance and intelligent services of Cisco IOS software for branch office applications. The Ethernet switch network module can identify user applications—such as voice or multicast video—and classify traffic with the appropriate priority levels. QoS policies are enforced using Layer 2 and 3 information such as 802.1p, IP precedence, and DSCP.

Note Refer to the *Cisco AVVID QoS Design Guide* for more information on how to implement end-to-end QoS as you deploy Cisco AVVID solutions.

To automatically configure Cisco IP phones to send voice traffic on the voice VLAN ID (VVID) on a per-port basis (see the "Voice Traffic and VVID" section on page 108), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# enable`	Enters the privileged EXEC mode. A preset password may be required to enter this mode.
Step 2	`Router(config)# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# interface interface`	Enters the interface configuration mode and the port to be configured (for example, interface fa5/1).

	Command	Purpose
Step 4	`Router(config)# `**`switchport access vlan`** *`vlan-id`*	Configures the port as "access" and assigns a data VLAN.
Step 5	`Router(config)# `**`switchport voice vlan`** *`vlan-id`*	Configures the voice port with a VVID that will be used exclusively for voice traffic.

Voice Traffic and VVID

The Ethernet switch network module can automatically configure voice VLAN. This capability overcomes the management complexity of overlaying a voice topology onto a data network while maintaining the quality of voice traffic. With the automatically configured voice VLAN feature, network administrators can segment phones into separate logical networks, even though the data and voice infrastructure is physically the same. The voice VLAN feature places the phones into their own VLANs without the need for end-user intervention. A user can plug the phone into the switch, and the switch provides the phone with the necessary VLAN information.

Configuring a Single Subnet for Voice and Data

For network designs with incremental IP telephony deployment, network managers can configure the Ethernet switch network module so that the voice and data traffic coexist on the same subnet. This might be necessary when it is impractical either to allocate an additional IP subnet for IP phones or to divide the existing IP address space into an additional subnet at the remote branch, it might be necessary to use a single IP address space for branch offices. (This is one of the simpler ways to deploy IP telephony.) When this is the case, you must still prioritize voice above data at both Layer 2 and Layer 3.

Layer 3 classification is already handled because the phone sets the Type of Service (ToS) bits in all media streams to an IP Precedence value of 5. (With Cisco CallManager Release 3.0(5), this marking changed to a Differentiated Services Code Point ([DSCP]) value of EF.) However, to ensure that there is Layer 2 classification for admission to the multiple queues in the branch office switches, the phone must also use the User Priority bits in the Layer 2 802.1p header to provide Class of Service (CoS) marking. Setting the bits to provide marking can be done by having the switch look for 802.1p headers on the native VLAN.

This configuration approach must address two key considerations:

- Network managers should ensure that existing subnets have enough available IP addresses for the new Cisco IP phones, each of which requires a unique IP address.
- Administering a network with a mix of IP phones and workstations on the same subnet might pose a challenge.

To automatically configure Cisco IP phones to send voice and data traffic on the same VLAN, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# `**`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)# `**`interface`** *`interface`*	Enters the interface configuration mode and the port to be configured (e.g., interface fa5/1).

	Command	Purpose
Step 3	`Router(config)# switchport access vlan vlan-id`	Sets the native VLAN for untagged traffic. The value of *vlan-id* represents the ID of the VLAN that is sending and receiving untagged traffic on the port. Valid IDs are from 1 to 1001. Leading zeroes are not accepted.
Step 4	`Router(config)# switchport voice vlan dot1p`	Configures the Cisco IP phone to send voice traffic with higher priority (CoS=5 on 802.1Q tag) on the access VLAN. Data traffic (from an attached PC) is sent untagged for lower priority (port default=0).
Step 5	`Router# end`	Returns to the privileged EXEC mode.

Verifying Switchport Configuration

Step 1 Use the **show run interface** command to verify the switch port configuration and the **write memory** command to save the current configuration in flash memory:

```
Router# show run interface interface
```

Step 2

```
Router# write memory
```

Configuring Ethernet Ports to Support Cisco IP Phones with Multiple Ports

You might want to use multiple ports to connect the Cisco IP phones if any of the following conditions apply to your Cisco IP telephony network:

- You are connecting Cisco IP phones that do not have a second Ethernet port for attaching a PC.
- You want to create a physical separation between the voice and data networks.
- You want to provide in-line power easily to the IP phones without having to upgrade the data infrastructure.
- You want to limit the number of switches that need Uninterruptible Power Supply (UPS) power.

IP Addressing

The recommended configuration for using multiple cables to connect IP phones to the Cisco AVVID network is to use a separate IP subnet and separate VLANs for IP telephony.

Managing the Ethernet Switch Network Module

This section describes how to perform basic management tasks on the Ethernet switch network module with the Cisco IOS CLI. You might find this information useful when you configure the switch for the previous scenarios.

The following topics are included:

- Adding Trap Managers, page 110
- Configuring IP Information, page 110

- Configuring Voice Ports, page 112
- Enabling Switch Port Analyzer, page 114
- Managing the ARP Table, page 114
- Managing the MAC Address Tables, page 115

Adding Trap Managers

A trap manager is a management station that receives and processes traps. When you configure a trap manager, community strings for each member switch must be unique. If a member switch has an IP address assigned to it, the management station accesses the switch by using its assigned IP address.

By default, no trap manager is defined, and no traps are issued.

To add a trap manager and community string, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# config terminal`	Enters global configuration mode.
Step 2	`Router(config)# snmp-server host 172.2.128.263 traps1 snmp vlan-membership`	Enters the trap manager IP address, community string, and the traps to generate.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.

Verifying Trap Managers

Step 1 Use the **show running-config** command to verify that the information was entered correctly by displaying the running configuration:

```
Router# show running-config
```

Configuring IP Information

This section describes how to assign IP information on the Ethernet switch network module. The following topics are included:

- Assigning IP Information to the Switch, page 110
- Specifying a Domain Name and Configuring the DNS, page 111

Assigning IP Information to the Switch

You can use a BOOTP server to automatically assign IP information to the switch; however, the BOOTP server must be set up in advance with a database of physical MAC addresses and corresponding IP addresses, subnet masks, and default gateway addresses. In addition, the switch must be able to access the BOOTP server through one of its ports. At startup, a switch without an IP address requests the information from the BOOTP server; the requested information is saved in the switch running the configuration file. To ensure that the IP information is saved when the switch is restarted, save the configuration by entering the **write memory** command in privileged EXEC mode.

You can change the information in these fields. The mask identifies the bits that denote the network number in the IP address. When you use the mask to subnet a network, the mask is then referred to as a subnet mask. The broadcast address is reserved for sending messages to all hosts. The CPU sends traffic to an unknown IP address through the default gateway.

To enter the IP information, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface vlan 1`**	Enters interface configuration mode, and enter the VLAN to which the IP information is assigned. VLAN 1 is the management VLAN, but you can configure any VLAN from IDs 1 to 1001.
Step 3	`Router(config)#` **`ip address`** *`ip-address subnet-mask`*	Enters the IP address and subnet mask.
Step 4	`Router(config)#` **`exit`**	Returns to global configuration mode.
Step 5	`Router#` **`ip default-gateway`** *`ip-address`*	Enters the IP address of the default router.
Step 6	`Router#` **`end`**	Returns to privileged EXEC mode.

Use the following procedure to remove the IP information from a switch.

Note Using the **no ip address** command in configuration mode disables the IP protocol stack and removes the IP information. Cluster members without IP addresses rely on the IP protocol stack being enabled.

To remove an IP address, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface vlan 1`**	Enters interface configuration mode, and enters the VLAN to which the IP information is assigned. VLAN 1 is the management VLAN, but you can configure any VLAN from IDs 1 to 1001.
Step 2	`Router(config-subif)#` **`no ip address`**	Removes the IP address and subnet mask.
Step 3	`Router(config-subif)#` **`end`**	Returns to privileged EXEC mode.

Caution If you are removing the IP address through a telnet session, your connection to the switch will be lost.

Specifying a Domain Name and Configuring the DNS

Each unique IP address can have a host name associated with it. The Cisco IOS software maintains a EC mode, and related Telnet support operations. This cache speeds the process of converting names to addresses.

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

To track domain names, IP has defined the concept of a domain name server (DNS), the purpose of which is to hold a cache (or database) of names mapped to IP addresses. To map domain names to IP addresses, you must first identify the host names and then specify a name server and enable the DNS, the Internet's global naming scheme that uniquely identifies network devices.

Specifying the Domain Name

You can specify a default domain name that the software uses to complete domain name requests. You can specify either a single domain name or a list of domain names. When you specify a domain name, any IP host name without a domain name has that domain name appended to it before being added to the host table.

Specifying a Name Server

You can specify up to six hosts that can function as a name server to supply name information for the DNS.

Enabling the DNS

If your network devices require connectivity with devices in networks for which you do not control name assignment, you can assign device names that uniquely identify your devices within the entire internetwork. The Internet's global naming scheme, the DNS, accomplishes this task. This service is enabled by default.

Configuring Voice Ports

This section describes how to configure voice ports on the Ethernet switch network module. The following topics are included:

- Configuring a Port to Connect to a Cisco 7960 IP phone, page 113
- Disabling Inline Power on a Ethernet switch network module, page 113

The Ethernet switch network module can connect to a Cisco 7960 IP phone and carry IP voice traffic. If necessary, the Ethernet switch network module can supply electrical power to the circuit connecting it to the Cisco 7960 IP phone.

Because the sound quality of an IP telephone call can deteriorate if the data is unevenly transmitted, the current release of the Cisco IOS software supports QoS based on IEEE 802.1p CoS. QoS uses classification and scheduling to transmit network traffic from the switch in a predictable manner.

The Cisco 7960 IP phone contains an integrated three-port 10/100 switch. The ports are dedicated to connect to the following devices:

- Port 1 connects to the Ethernet switch network module switch 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

Configuring a Port to Connect to a Cisco 7960 IP phone

Because a Cisco 7960 IP phone also supports connection to a PC or other device, a port connecting a Ethernet switch network module to a Cisco 7960 IP phone can carry a mix of traffic. There are three ways to configure a port connected to a Cisco 7960 IP phone:

- All traffic is transmitted according to the default COS priority (0) 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.

To instruct the phone to give voice traffic a higher priority and to forward all traffic through the 802.1Q native VLAN, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode, and enter the port to be configured.
Step 3	`Router(config-if)#` **`switchport voice vlan dot1p`**	Instructs the switch to use 802.1p priority tagging for voice traffic and to use VLAN 0 (default native VLAN) to carry all traffic.
Step 4	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Router#` **`show interface switchport`**	Displays the administrative and operational status of a switching (nonrouting) port.

Disabling Inline Power on a Ethernet switch network module

The Ethernet switch network module can supply inline power to a 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. When the Cisco 7960 IP phone is supplying its own power, a Ethernet switch network module can forward IP voice traffic to and from the phone.

A detection mechanism on the Ethernet switch network module determines whether it is connected to a Cisco 7960 IP phone. If the switch senses that 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.

To configure a port to never supply power to Cisco 7960 IP phones, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode, and enter the port to be configured.
Step 3	`Router(config-if)#` **`power inline never`**	Permanently disables inline power on the port.
Step 4	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.

Verifying Inline Power Configuration

Step 1 Use the **show power inline** *interface* **configured** command to verifies the change by displaying the setting as configured:

```
Router# show power inline interface configured
```

Enabling Switch Port Analyzer

You can monitor traffic on a given port by forwarding incoming and outgoing traffic on the port to another port in the same VLAN. A Switch Port Analyzer (SPAN) port cannot monitor ports in a different VLAN, and a SPAN port must be a static-access port. Any number of ports can be defined as SPAN ports, and any combination of ports can be monitored. SPAN is supported for up to 2 sessions.

To enable SPAN, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# monitor session session-id {destination	source} {interface	vlan interface-id	vlan-id}} [,	-	both	tx	rx]`	Enables port monitoring for a specific session (“*number”).* Optionally, supply a SPAN *destination* interface, and a *source* interface.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.

To disable SPAN, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# no monitor session session-id`	Disables port monitoring for a specific session.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.

Managing the ARP Table

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

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

When you manually add entries to the ARP Table by using the CLI, you must be aware that these entries do not age and must be manually removed.

Managing the MAC Address Tables

This section describes how to manage the MAC address tables on the Ethernet switch network module. The following topics are included:

- Understanding MAC Addresses and VLANs, page 115
- Changing the Address Aging Time, page 116
- Configuring the Aging Time, page 116

The switch uses the MAC address tables to forward traffic between ports. All MAC addresses in the address tables are associated with one or more ports. These MAC tables include the following types of addresses:

- Dynamic address—a source MAC address that the switch learns and then drops when it is not in use.
- Secure address—a manually entered unicast address that is usually associated with a secured port. Secure addresses do not age.
- Static address—a manually entered unicast or multicast address that does not age and that is not lost when the switch resets.

The address tables list the destination MAC address and the associated VLAN ID, module, and port number associated with the address. The following shows an example of a list of addresses as they would appear in the dynamic, secure, or static address table.

```
Router# show mac

4d01h:%SYS-5-CONFIG_I:Configured from console by consolec
Slot # :0
--------------
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
0004.272f.49de          Dynamic      1      FastEthernet0/8
0004.2762.3235          Dynamic      1      FastEthernet0/3
0004.4d07.6960          Dynamic      1      FastEthernet0/0
0004.ddbb.6700          Self         1      Vlan1
0020.18d7.4304          Dynamic      1      FastEthernet0/2
beef.beef.beef          Static       1      FastEthernet0/11
0004.2762.3235          Dynamic      2      FastEthernet0/3
0004.ddbb.6700          Self         2      Vlan2
0002.7e48.cc38          Dynamic      3      FastEthernet0/4
    0002.7e48.cc39          Dynamic      3      FastEthernet0/5
```

Understanding MAC Addresses and VLANs

All addresses are associated with a VLAN. An address can exist in more than one VLAN and have different destinations in each. Multicast addresses, for example, could be forwarded to port 1 in VLAN 1 and ports 9, 10, and 11 in VLAN 5.

Each VLAN maintains its own logical address table. A known address in one VLAN is unknown in another until it is learned or statically associated with a port in the other VLAN. An address can be secure in one VLAN and dynamic in another. Addresses that are statically entered in one VLAN must be static addresses in all other VLANs.

Changing the Address Aging Time

Dynamic addresses are source MAC addresses that the switch learns and then drops when they are not in use. Use the Aging Time field to define how long the switch retains unseen addresses in the table. This parameter applies to all VLANs.

Configuring the Aging Time

Setting too short an aging time can cause addresses to be prematurely removed from the table. Then when the switch receives a packet for an unknown destination, it floods the packet to all ports in the same VLAN as the receiving port. This unnecessary flooding can impact performance. Setting too long an aging time can cause the address table to be filled with unused addresses; it can cause delays in establishing connectivity when a workstation is moved to a new port.

To configure the dynamic address table aging time, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# mac-address-table aging-time seconds`	Enters the number of seconds that dynamic addresses are to be retained in the address table. Valid entries are from 10 to 1000000.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.

Verifying Aging-Time Configuration

Step 1 Use the **show mac-address-table aging-time** command to verify configuration:

```
Router# show mac-address-table aging-time
```

Removing Dynamic Addresses

To remove a dynamic address entry, follow these steps beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# no mac-address-table dynamic hw-addr`	Enters the MAC address to be removed from dynamic MAC address table.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.

You can remove all dynamic entries by using the **clear mac-address-table dynamic** command in privileged EXEC mode.

Verifying Dynamic Addresses

Step 1 Use the **show mac-address-table dynamic** command to verify configuration:

```
Router# show mac-address-table dynamic
```

Adding Secure Addresses

The secure address table contains secure MAC addresses and their associated ports and VLANs. A secure address is a manually entered unicast address that is forwarded to only one port per VLAN. If you enter an address that is already assigned to another port, the switch reassigns the secure address to the new port.

You can enter a secure port address even when the port does not yet belong to a VLAN. When the port is later assigned to a VLAN, packets destined for that address are forwarded to the port.

To add a secure address, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# mac-address-table secure address hw-addr interface interface-id vlan vlan-id`	Enters the MAC address, its associated port, and the VLAN ID.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.

To remove a secure address, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# no mac-address-table secure hw-addr vlan vlan-id`	Enters the secure MAC address, its associated port, and the VLAN ID to be removed.
Step 3	`Router(config)# end`	Returns to privileged EXEC mode.

You can remove all secure addresses by using the **clear mac-address-table secure** command in privileged EXEC mode.

Verifying Secure Addresses

Step 1 Use the **show mac-address-table secure** command to verify configuration:

```
Router# show mac-address-table secure
```

Configuring Static Addresses

A static address has the following characteristics:

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

Because all ports are associated with at least one VLAN, the switch acquires the VLAN ID for the address from the ports that you select on the forwarding map. A static address in one VLAN must be a static address in other VLANs. A packet with a static address that arrives on a VLAN where it has not been statically entered is flooded to all ports and not learned.

To add a static address, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`mac-address-table static`** *`hw-addr`* `[`**`interface`**`]` *`interface-id`* `[`**`vlan`**`]` *`vlan-id`*	Enters the static MAC address, the interface, and the VLAN ID of those ports.
Step 3	`Router(config)#` **`end`**	Returns to privileged EXEC mode.

To remove a static address, use the following commands beginning in privileged EXEC mode

:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`no mac-address-table static`** *`hw-addr`* `[`**`interface`**`]` *`interface-id`* `[`**`vlan`**`]` *`vlan-id`*	Enters the static MAC address, the interface, and the VLAN ID of the port to be removed.
Step 3	`Router(config)#` **`end`**	Returns to privileged EXEC mode.

You can remove all secure addresses by using the **clear mac-address-table static** command in privileged EXEC mode.

Verifying Static Addresses

Step 1 Use the **show mac-address-table static** command to verify configuration:

```
Router # show mac-address-table static

4d01h:%SYS-5-CONFIG_I:Configured from console by consolec
Slot # :0
--------------
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
0004.272f.49de          Dynamic      1      FastEthernet0/8
0004.2762.3235          Dynamic      1      FastEthernet0/3
0004.4d07.6960          Dynamic      1      FastEthernet0/0
0004.ddbb.6700          Self         1      Vlan1
0020.18d7.4304          Dynamic      1      FastEthernet0/2
beef.beef.beef          Static       1      FastEthernet0/11
0004.2762.3235          Dynamic      2      FastEthernet0/3
```

```
0004.ddbb.6700          Self         2      Vlan2
0002.7e48.cc38          Dynamic      3      FastEthernet0/4
0002.7e48.cc39          Dynamic      3      FastEthernet0/5
```

Clearing all MAC Address Tables

To remove all addresses, use the **clear mac-address** command in privileged EXEC mode:

	Command	Purpose
Step 1	`Router#` **`clear mac-address-table`**	Enters to clear all MAC address tables.
Step 2	`Router#` **`end`**	Returns to privileged EXEC mode.

Configuring Intrachassis Stacking

To extend Layer 2 switching in the router by connecting the Gigabit Ethernet ports of the Ethernet switch network module, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`interface Gigabit`** *`slot/port`*	Enters the current Gigabit Ethernet interface being used for intrachassis stacking.
Step 1	`Router(config-if)# [`**`no`**`]` **`switchport stacking-link interface Gigabit`** *`slot/port`*	Creates the intrachassis stacking between the current GE interface and the stacking link partner GE interface. To restore the defaults, use the **no** form of this command .
Step 2	`Router(config)#` **`end`**	Exits configuration mode.

Verifying Intra-chassis Stacking

Step 1 Use the **show interface** command to verify configuration:

```
Router# show interface Gigabit slot/port
```

Configuring Flow Control on Gigabit Ethernet Ports

To configure flow control on a Gigabit Ethernet port, use the following commands in privileged mode:

	Command	Purpose
Step 1	`Router#` **`set port flowcontrol`** `{`**`receive`** `|` **`send`**`}` *`mod-num/port-num`* `{`**`off`** `|` **`on`** `|` **`desired`**`}`	Sets the flow control parameters on a Gigabit Ethernet port.
Step 2	`Router#` **`show port flowcontrol`**	Verifies the flow control configuration.

Configuring Layer 3 Interfaces

The Ethernet switch network module supports two types of Layer 3 interfaces for routing and bridging:

- SVIs: You should configure SVIs for any VLANs for which you want to route traffic. SVIs are created when you enter a VLAN ID following the **interface vlan** global configuration command. To delete an SVI, use the **no interface vlan** global configuration command.
- Routed ports: Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command.

Note A Layer 3 switch can have an IP address assigned to each routed port and SVI. The number of routed ports and SVIs that you can configure is not limited by software; however, the interrelationship between this number and the number of other features being configured might have an impact on CPU utilization because of hardware limitations.

All Layer 3 interfaces require an IP address to route traffic (a routed port cannot obtain an IP address from a DHCP server, but the router can act as a DHCP server and serve IP addresses through a routed port). The following procedure shows how to configure an interface as a Layer 3 interface and how to assign an IP addresses to an interface.

Routed ports support only CEF switching (IP fast switching is not supported).

Note If the physical port is in Layer 2 mode (the default), you must enter the **no switchport** interface configuration command to put the interface into Layer 3 mode. Entering a **no switchport** command disables and then reenables the interface, which might generate messages on the device to which the interface is connected. When you use this command to put the interface into Layer 3 mode, you are also deleting any Layer 2 characteristics configured on the interface. (Also, when you return the interface to Layer 2 mode, you are deleting any Layer 3 characteristics configured on the interface.)

Beginning in privileged EXEC mode, follow these steps to configure a Layer 3 interface:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** {{**fastethernet** | **gigabitethernet**} *interface-id*} | {**vlan** *vlan-id*} | {**port-channel** *port-channel-number*}	Enters interface configuration mode, and enter the interface to be configured as a Layer 3 interface.
Step 3	**no switchport**	For physical ports only, enters Layer 3 mode.
Step 4	**ip address** *ip-address subnet-mask*	Configures the IP address and IP subnet.
Step 5	**no shutdown**	Activates the interface. (Required only if you shut down the interface.)
Step 6	**end**	Returns to privileged EXEC mode.
Step 7	**show interfaces** [*interface-id*] **show ip interface** [*interface-id*] **show running-config interface** [*interface-id*]	Verifies the configuration.
Step 8	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To remove an IP address from an interface, use the **no ip address** interface configuration command.

Configuring Fallback Bridging

This section describes how to configure fallback bridging on your switch. It contains this configuration information:

Understanding the Default Fallback Bridging Configuration

Table 16 shows the default fallback bridging configuration.

Table 16 Default Fallback Bridging Configuration

Feature	Default Setting
Bridge groups	None are defined or assigned to an interface. No VLAN-bridge STP is defined.
Switch forwards frames for stations that it has dynamically learned	Enabled.
Bridge table aging time for dynamic entries	300 seconds.
MAC-layer frame filtering	Disabled.
Spanning tree parameters:	
• Switch priority	• 32768.
• Interface priority	• 128.
• Interface path cost	• 10 Mbps: 100. 100 Mbps: 19. 1000 Mbps: 4.
• Hello BPDU interval	• 2 seconds.
• Forward-delay interval	• 20 seconds.
• Maximum idle interval	• 30 seconds.

Creating a Bridge Group

To configure fallback bridging for a set of SVIs or routed ports, these interfaces must be assigned to bridge groups. All interfaces in the same group belong to the same bridge domain. Each SVI or routed port can be assigned to only one bridge group. A maximum of 31 bridge groups can be configured on the switch.

Note The protected port feature is not compatible with fallback bridging. When fallback bridging is enabled, it is possible for packets to be forwarded from one protected port on a switch to another protected port on the same switch if the ports are in different VLANs.

Beginning in privileged EXEC mode, follow these steps to create a bridge group and assign an interface to it:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**bridge** *bridge-group* **protocol vlan-bridge**	Assigns a bridge group number, and specify the VLAN-bridge spanning-tree protocol to run in the bridge group. The **ibm** and **dec** keywords are not supported. For *bridge-group*, specify the bridge group number. The range is 1 to 255. You can create up to 31 bridge groups. Frames are bridged only among interfaces in the same group.
Step 3	**interface** *interface-id*	Enters interface configuration mode, and specify the interface on which you want to assign the bridge group. The specified interface must be one of these: • A routed port: a physical port that you have configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI: a VLAN interface that you created by using the **interface vlan** *vlan-id* global configuration command. These ports must have IP addresses assigned to them.
Step 4	**bridge-group** *bridge-group*	Assigns the interface to the bridge group created in Step 2. By default, the interface is not assigned to any bridge group. An interface can be assigned to only one bridge group.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

To remove a bridge group, use the **no bridge** *bridge-group* **protocol vlan-bridge** global configuration command. To remove an interface from a bridge group, use the **no bridge-group** *bridge-group* interface configuration command.

Preventing the Forwarding of Dynamically Learned Stations

By default, the switch forwards any frames for stations that it has dynamically learned. By disabling this activity, the switch only forwards frames whose addresses have been statically configured into the forwarding cache.

Beginning in privileged EXEC mode, follow these steps to prevent the switch from forwarding frames for stations that it has dynamically learned:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**no bridge** *bridge-group* **acquire**	Enables the switch to stop forwarding any frames for stations that it has dynamically learned through the discovery process and to limit frame forwarding to statically configured stations. The switch filters all frames except those whose destined-to addresses have been statically configured into the forwarding cache. To configure a static address, use the **bridge** *bridge-group* **address** *mac-address* {**forward** | **discard**} global configuration command. For *bridge-group*, specify the bridge group number. The range is 1 to 255.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entry.
Step 5	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To cause the switch to forward frames to stations that it has dynamically learned, use the **bridge** *bridge-group* **acquire** global configuration command.

Configuring the Bridge Table Aging Time

A switch forwards, floods, or drops packets based on the bridge table. The bridge table maintains both static and dynamic entries. Static entries are entered by you or learned by the switch. Dynamic entries are entered by the bridge learning process. A dynamic entry is automatically removed after a specified length of time, known as aging time, from the time the entry was created or last updated.

If you are likely to move hosts on a switched network, decrease the aging-time to enable the switch to quickly adapt to the change. If hosts on a switched network do not continuously send packets, increase the aging time to keep the dynamic entries for a longer time and thus reduce the possibility of flooding when the hosts send again.

Beginning in privileged EXEC mode, follow these steps to configure the aging time:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**bridge** *bridge-group* **aging-time** *seconds*	Specifies the length of time that a dynamic entry remains in the bridge table from the time the entry was created or last updated. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *seconds*, enter a number from 0 to 1000000. The default is 300 seconds.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entry.
Step 5	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To return to the default aging-time interval, use the **no bridge** *bridge-group* **aging-time** global configuration command.

Filtering Frames by a Specific MAC Address

A switch examines frames and sends them through the internetwork according to the destination address; a switch does not forward a frame back to its originating network segment. You can use the software to configure specific administrative filters that filter frames based on information other than the paths to their destinations.

You can filter frames with a particular MAC-layer station destination address. Any number of addresses can be configured in the system without a performance penalty.

Beginning in privileged EXEC mode, follow these steps to filter by the MAC-layer address:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**bridge** *bridge-group* **address** *mac-address* {**forward** | **discard**} [*interface-id*]	Specifies the MAC address to discard or forward. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For **address** *mac-address*, specify the MAC-layer destination address to be filtered. • Specify **forward** if you want the frame destined to the specified interface to be forwarded. Specify **discard** if you want the frame to be discarded. • (Optional) For *interface-id*, specify the interface on which the address can be reached.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entry.
Step 5	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To disable the frame forwarding ability, use the **no bridge** *bridge-group* **address** *mac-address* global configuration command.

Adjusting Spanning-Tree Parameters

You might need to adjust certain spanning-tree parameters if the default values are not suitable for your switch configuration. Parameters affecting the entire spanning tree are configured with variations of the **bridge** global configuration command. Interface-specific parameters are configured with variations of the **bridge-group** interface configuration command.

You can adjust spanning-tree parameters by performing any of the tasks in these sections:

- Changing the Switch Priority, page 125
- Changing the Interface Priority, page 125
- Assigning a Path Cost, page 126
- Adjusting BPDU Intervals, page 126
- Disabling the Spanning Tree on an Interface, page 128

Note Only network administrators with a good understanding of how switches and STP function should make adjustments to spanning-tree parameters. Poorly planned adjustments can have a negative impact on performance. A good source on switching is the IEEE 802.1d specification; for more information, refer to the "References and Recommended Reading" appendix in the *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.2.

Changing the Switch Priority

You can globally configure the priority of an individual switch when two switches tie for position as the root switch, or you can configure the likelihood that a switch will be selected as the root switch. This priority is determined by default; however, you can change it.

Beginning in privileged EXEC mode, follow these steps to change the switch priority:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**bridge** *bridge-group* **priority** *number*	Changes the priority of the switch. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *number*, enter a number from 0 to 65535. The default is 32768. The lower the number, the more likely the switch will be chosen as the root.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entry.
Step 5	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

No **no** form of this command exists. To return to the default setting, use the **bridge** *bridge-group* **priority** *number* global configuration command, and set the priority to the default value. To change the priority on an interface, use the **bridge-group priority** interface configuration command (described in the next section).

Changing the Interface Priority

You can change the priority for an interface. When two switches tie for position as the root switch, you configure an interface priority to break the tie. The switch with the lowest interface value is elected.

Beginning in privileged EXEC mode, follow these steps to change the interface priority:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specifies the interface to set the priority.

	Command	Purpose
Step 3	**bridge-group** *bridge-group* **priority** *number*	Changes the priority of an interface. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *number*, enter a number from 0 to 255. The lower the number, the more likely that the interface on the switch will be chosen as the root. The default is 128.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entry.
Step 6	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To return to the default setting, use the **bridge-group** *bridge-group* **priority** *number* interface configuration command.

Assigning a Path Cost

Each interface has a path cost associated with it. By convention, the path cost is 1000/data rate of the attached LAN, in Mbps.

Beginning in privileged EXEC mode, follow these steps to assign a path cost:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specify the interface to set the path cost.
Step 3	**bridge-group** *bridge-group* **path-cost** *cost*	Assigns the path cost of an interface. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *cost*, enter a number from 1 to 65536. The higher the value, the higher the cost. – For 10 Mbps, the default path cost is 100. – For 100 Mbps, the default path cost is 19. – For 1000 Mbps, the default path cost is 4.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entry.
Step 6	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To return to the default path cost, use the **no bridge-group** *bridge-group* **path-cost** *cost* interface configuration command.

Adjusting BPDU Intervals

You can adjust BPDU intervals as described in these sections:

- Adjusting the Interval between Hello BPDUs, page 127

- Changing the Forward-Delay Interval, page 127
- Changing the Maximum-Idle Interval, page 128

Note Each switch in a spanning tree adopts the interval between hello BPDUs, the forward delay interval, and the maximum idle interval parameters of the root switch, regardless of what its individual configuration might be.

Adjusting the Interval between Hello BPDUs

Beginning in privileged EXEC mode, follow these step to adjust the interval between hello BPDUs:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**bridge** *bridge-group* **hello-time** *seconds*	Specifies the interval between hello BPDUs. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *seconds*, enter a number from 1 to 10. The default is 2 seconds.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entry.
Step 5	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To return to the default setting, use the **no bridge** *bridge-group* **hello-time** global configuration command.

Changing the Forward-Delay Interval

The forward-delay interval is the amount of time spent listening for topology change information after an interface has been activated for switching and before forwarding actually begins.

Beginning in privileged EXEC mode, follow these steps to change the forward-delay interval:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**bridge** *bridge-group* **forward-time** *seconds*	Specifies the forward-delay interval. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *seconds*, enter a number from 10 to 200. The default is 20 seconds.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entry.
Step 5	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To return to the default setting, use the **no bridge** *bridge-group* **forward-time** *seconds* global configuration command.

Changing the Maximum-Idle Interval

If a switch does not hear BPDUs from the root switch within a specified interval, it recomputes the spanning-tree topology.

Beginning in privileged EXEC mode, follow these steps to change the maximum-idle interval (maximum aging time):

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**bridge** *bridge-group* **max-age** *seconds*	Specifies the interval the switch waits to hear BPDUs from the root switch. • For *bridge-group*, specify the bridge group number. The range is 1 to 255. • For *seconds*, enter a number from 10 to 200. The default is 30 seconds.
Step 3	**end**	Returns to privileged EXEC mode.
Step 4	**show running-config**	Verifies your entry.
Step 5	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To return to the default setting, use the **no bridge** *bridge-group* **max-age** global configuration command.

Disabling the Spanning Tree on an Interface

When a loop-free path exists between any two switched subnetworks, you can prevent BPDUs generated in one switching subnetwork from impacting devices in the other switching subnetwork, yet still permit switching throughout the network as a whole. For example, when switched LAN subnetworks are separated by a WAN, BPDUs can be prevented from traveling across the WAN link.

Beginning in privileged EXEC mode, follow these steps to disable spanning tree on an interface:

	Command	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *interface-id*	Enters interface configuration mode, and specify the interface ID.
Step 3	**bridge-group** *bridge-group* **spanning-disabled**	Disables spanning tree on the interface. For *bridge-group*, specify the bridge group number. The range is 1 to 255.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show running-config**	Verifies your entry.
Step 6	**copy running-config startup-config**	(Optional) Saves your entry in the configuration file.

To reenable spanning tree on the interface, use the **no bridge-group** *bridge-group* **spanning-disabled** interface configuration command.

Monitoring and Maintaining the Network

To monitor and maintain the network, use one or more of the privileged EXEC commands in Table 17:

Table 17 Fallback Bridging Commands for Monitoring and Maintaining the Network

Command	Purpose
clear bridge *bridge-group*	Removes any learned entries from the forwarding database and clears the transmit and receive counts for any statically configured entries.
show bridge [*bridge-group*]	Displays details about the bridge group.
show bridge [*bridge-group*] [*interface-id*] [*address*] [**group**] [**verbose**]	Displays classes of entries in the bridge forwarding database.

Configuration Examples for the 16- and 36-Port Ethernet Switch Module

This section provides the following configuration examples:

Range of Interface Examples

Single Range Configuration Example

The following example shows all Fast Ethernet interfaces 5/1 to 5/5 being reenabled:

```
Router(config)# interface range fastethernet 5/1 - 5
Router(config-if)# no shutdown
Router(config-if)#
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/2, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/3, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/4, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
```

```
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/5, changed
state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/3, changed
state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/4, changed
state to up
Router(config-if)#
```

Multiple Range Configuration Example

The following 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:

```
Router(config-if)# interface range fastethernet 5/1 - 5, gigabitethernet 1/1 - 2
Router(config-if)# no shutdown
Router(config-if)#
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/1, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/2, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/3, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/4, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet1/1, changed state to up
*Oct  6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet1/2, changed state to up
*Oct  6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/5, changed
state to up
*Oct  6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/3, changed
state to up
*Oct  6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/4, changed
state to up
Router(config-if)#
```

Range Macro Definition Example

The following example shows an interface-range macro named enet_list being defined to select Fast Ethernet interfaces 5/1 through 5/4:

```
Router(config)# define interface-range enet_list fastethernet 5/1 - 4

Router(config)#
```

The following example shows how to change to the interface-range configuration mode using the interface-range macro enet_list:

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

Router(config-if)#
```

Optional Interface Feature Examples

- Interface Speed Example, page 132
- Setting the Interface Duplex Mode Example, page 132
- Adding a Description for an Interface Example, page 132
- Configuring an Ethernet Interface as a Layer 2 Trunk Example, page 132

Interface Speed Example

The following example shows the interface speed being set to 100 Mbps on the Fast Ethernet interface 5/4:

```
Router(config)# interface fastethernet 5/4

Router(config-if)# speed 100
```

Setting the Interface Duplex Mode Example

The following example shows the interface duplex mode being set to full on Fast Ethernet interface 5/4:

```
Router(config)# interface fastethernet 5/4

Router(config-if)# duplex full
```

Adding a Description for an Interface Example

The following example shows how to add a description on Fast Ethernet interface 5/5:

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

Configuring an Ethernet Interface as a Layer 2 Trunk Example

The following 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:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastethernet 5/8
Router(config-if)# shutdown
Router(config-if)# switchport trunk encapsulation dot1q
Router(config-if)# switchport mode trunk
Router(config-if)# no shutdown
Router(config-if)# end
Router# exit
```

VLAN Configuration Example

The following example shows how to configure the VLAN:

```
Router# vlan database
Router(vlan)# vlan 3
VLAN 3 added:
Name: VLAN0003
Router(vlan)# exit
APPLY completed.
Exiting....
```

VTP Examples

VTP Server Example

The following example shows how to configure the switch as a VTP server:

```
Router# vlan database
Router(vlan)# vtp server
Setting device to VTP SERVER mode.
Router(vlan)# vtp domain Lab_Network
Setting VTP domain name to Lab_Network
Router(vlan)# vtp password WATER
Setting device VLAN database password to WATER.
Router(vlan)# exit
APPLY completed.
Exiting....
Router#
```

VTP Client Example

The following example shows how to configure the switch as a VTP client:

```
Router# vlan database
Router(vlan)# vtp client
Setting device to VTP CLIENT mode.
Router(vlan)# exit

In CLIENT state, no apply attempted.
Exiting....
Router#
```

Disabling VTP (VTP Transparent Mode) Example

The following example shows how to configure the switch as VTP transparent:

```
Router# vlan database
Router(vlan)# vtp transparent
Setting device to VTP TRANSPARENT mode.
Router(vlan)# exit
APPLY completed.
Exiting....
Router#
```

VTP version 2 Example

The following example shows VTP version 2 being enabled:

```
Router# vlan database

Router(vlan)# vtp v2-mode

V2 mode enabled.
```

```
Router(vlan)# exit

APPLY completed.
Exiting....
Router#
```

EtherChannel Load Balancing Example

- Layer 2 EtherChannels Example, page 134
- EtherChannel Load Balancing Example, page 134
- Removing an EtherChannel Example, page 134

Layer 2 EtherChannels Example

The following example shows Fast Ethernet interfaces 5/6 and 5/7 being configured into port-channel 2 with PAgP mode desirable:

```
Router# configure terminal
Router(config)# interface range fastethernet 5/6 -7
Router(config-if)# channel-group 2 mode desirable
Router(config-if)# end
```

EtherChannel Load Balancing Example

The following example shows EtherChannel being configured to use source and destination IP addresses:

```
Router# configure terminal
Router(config)# port-channel load-balance src-dst-ip
Router(config)# end
Router(config)#
```

Removing an EtherChannel Example

The following example shows port-channel 1 being removed:

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

Note Removing the port-channel also removes the channel-group command from the interfaces belonging to it.

802.1x Authentication Examples

- Enabling 802.1x Authentication Example, page 135
- Configuring the Switch-to-RADIUS-Server Communication Example, page 135
- Enabling Periodic Re-Authentication Example, page 135
- Changing the Quiet Period Example, page 135
- Changing the Switch-to-Client Retransmission Time Example, page 135

Enabling 802.1x Authentication Example

The following example shows how to enable AAA and 802.1x on Fast Ethernet port 0/1:

```
Switch# configure terminal
Switch(config)# aaa new-model
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# interface fastethernet0/1
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
```

Configuring the Switch-to-RADIUS-Server Communication Example

The following example shows how to specify the server with IP address 172.20.39.46 as the RADIUS server, to use port 1612 as the authorization port, and to set the encryption key to rad123, matching the key on the RADIUS server:

```
Switch(config)# radius-server host 172.120.39.46 auth-port 1612 key rad123
```

Enabling Periodic Re-Authentication Example

The following example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000:

```
Switch(config)# dot1x re-authentication
Switch(config)# dot1x timeout re-authperiod 4000
```

Changing the Quiet Period Example

The following example shows how to set the quiet time on the switch to 30 seconds:

```
Switch(config)# dot1x timeout quiet-period 30
```

Changing the Switch-to-Client Retransmission Time Example

The following example shows how to set 60 seconds as the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request:

```
Switch(config)# dot1x timeout tx-period 60
```

Setting the Switch-to-Client Frame-Retransmission Number Example

The following example shows how to set 5 as the number of times that the switch sends an EAP-request/identity request before restarting the authentication process:

```
Switch(config)# dot1x max-req 5
```

Enabling Multiple Hosts Example

The following example shows how to enable 802.1x on Fast Ethernet interface 0/1 and to allow multiple hosts:

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x multiple-hosts
```

Spanning Tree Examples

Spanning-Tree Interface and Spanning-Tree Port Priority Example

The following example shows the VLAN port priority of an interface being configured:

```
Router# configure terminal
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree vlan 200 port-priority 64
Router(config-if)# end
Router#
```

The following example shows how to verify the configuration of VLAN 200 on the interface when it is configured as a trunk port:

```
Router# show spanning-tree vlan 200
!
!
!
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
Router#
!
!
!
```

Spanning-Tree Port Cost Example

The following example shows how to change the spanning-tree port cost of a Fast Ethernet interface:

```
Router# configure terminal
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree cost 18
Router(config-if)# end
Router#
```

The following example shows how to verify the configuration of the interface when it is configured as an access port:

```
Router# 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
Router#
```

The following example shows how to configure the spanning-tree VLAN port cost of a Fast Ethernet interface:

```
Router# configure terminal
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree vlan 200 cost 17
Router(config-if)# exit
Router(config)# exit
Router#
```

Bridge Priority of a VLAN

The following example shows the bridge priority of VLAN 200 being configured to 33792:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 priority 33792
Router(config)# end
Router#
```

Hello Time Example

The following example shows the hello time for VLAN 200 being configured to 7 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 hello-time 7
Router(config)# end
Router#
```

Forward-Delay Time for a VLAN Example

The following example shows the forward delay time for VLAN 200 being configured to 21 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 forward-time 21
Router(config)# end
Router#
```

Maximum Aging Time for a VLAN Example

The following example configures the maximum aging time for VLAN 200 to 36 seconds:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200 max-age 36
Router(config)# end
Router#
```

BackboneFast Example

The following example shows BackboneFast being enabled on the Ethernet switch module:

```
Router# configure terminal
Router(config)# spanning-tree backbonefast
Router(config)# end
Router#
```

Spanning Tree Examples

The following example shows spanning tree being enabled on VLAN 200:

```
Router# configure terminal
Router(config)# spanning-tree vlan 200
Router(config)# end
Router#
```

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.

The following example shows spanning tree being disabled on VLAN 200:

```
Router# configure terminal
Router(config)# no spanning-tree vlan 200
Router(config)# end
Router#
```

Spanning Tree Root Example

The following example shows the switch being configured as the root bridge for VLAN 10, with a network diameter of 4:

```
Router# configure terminal
Router(config)# spanning-tree vlan 10 root primary diameter 4
Router(config)# exit
Router#
```

Mac Table Manipulation Examples

The following example shows a dynamic entry being configured in the MAC address table:

```
Router# configure terminal
Router (config)# mac-address-table dynamic 6.6.6 fastEthernet 2/13 vlan 1
Router (config)# end
```

The following example shows a static entry being configured in the MAC address table:

```
Router(config)# mac-address-table static beef.beef.beef int fa0/11 vlan 1
Router(config)# end
```

Cisco Discovery Protocol (CDP) Example

The following example shows CDP counter configuration being configured on the NM-16ESW:

```
Router# clear cdp counters
```

Switched Port Analyzer (SPAN) Source Examples

- SPAN Source Configuration Example, page 139
- SPAN Destinations Example, page 139
- Removing Sources or Destinations from a SPAN Session Example, page 139

SPAN Source Configuration Example

The following example shows SPAN session 1 being configured to monitor bidirectional traffic from source interface Fast Ethernet 5/1:

```
Router(config)# monitor session 1 source interface fastethernet 5/1
```

SPAN Destinations Example

The following example shows interface Fast Ethernet 5/48 being configured as the destination for SPAN session 1:

```
Router(config)# monitor session 1 destination interface fastethernet 5/48
```

Removing Sources or Destinations from a SPAN Session Example

The following example shows interface Fast Ethernet 5/2 being removed as a SPAN source for SPAN session 1:

```
Router(config)# no monitor session 1 source interface fastethernet 5/2
```

Network Security and ACL Configuration Examples

- Creating Numbered Standard and Extended ACLs Example, page 139
- Creating Named Standard and Extended ACLs Example, page 140
- Including Comments About Entries in ACLs Example, page 141
- Applying the ACL to an Interface Example, page 141
- Displaying Standard and Extended ACLs Example, page 141
- Displaying Access Groups Example, page 142
- Compiling ACLs Example, page 143

Creating Numbered Standard and Extended ACLs Example

The following example shows how to create a standard ACL to deny access to IP host 171.69.198.102, permit access to any others, and display the results:

```
Switch (config)# access-list 2 deny host 171.69.198.102
Switch (config)# access-list 2 permit any
Switch(config)# end
Switch# show access-lists
Standard IP access list 2
    deny   171.69.198.102
    permit any
```

The following example shows that the switch accepts addresses on network 36.0.0.0 subnets and denies all packets coming from 56.0.0.0 subnets. The ACL is then applied to packets entering Gigabit Ethernet interface 0/1:

```
Switch(config)# access-list 2 permit 36.0.0.0 0.255.255.255
Switch(config)# access-list 2 deny 56.0.0.0 0.255.255.255
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group 2 in
```

The following example shows how to create and display an extended access list to deny Telnet access from any host in network 171.69.198.0 to any host in network 172.20.52.0 and permit any others (the **eq** keyword after the destination address means to test for the TCP destination port number equaling Telnet):

```
Switch(config)# access-list 102 deny tcp 171.69.198.0 0.0.0.255 172.20.52.0 0.0.0.255 eq
telnet
Switch(config)# access-list 102 permit tcp any any
Switch(config)# end
Switch# show access-lists
Extended IP access list 102
    deny tcp 171.69.198.0 0.0.0.255 172.20.52.0 0.0.0.255 eq telnet
    permit tcp any any
```

The following example shows an extended ACL with a network connected to the Internet, and any host on the network being able to form TCP Telnet and SMTP connections to any host on the Internet:

```
Switch(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 23
Switch(config)# access-list 102 permit tcp any 128.88.0.0 0.0.255.255 eq 25
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group 102 in
```

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same port numbers are used throughout the life of the connection. Mail packets coming in from the Internet have a destination port of 25. Because the secure system behind the switch always accepts mail connections on port 25, the incoming services are controlled.

Creating Named Standard and Extended ACLs Example

The following example shows how you can delete individual ACEs from a named ACL:

```
Switch(config)# ip access-list extended border-list
Switch(config-ext-nacl)# no permit ip host 10.1.1.3 any
```

The following example shows the Marketing_group ACL allowing any TCP Telnet traffic to the destination address and wildcard 171.69.0.0 0.0.255.255 and denying any other TCP traffic. It permits any other IP traffic:

```
Switch(config)# ip access-list extended marketing_group
Switch(config-ext-nacl)# permit tcp any 171.69.0.0 0.0.255.255 eq telnet
Switch(config-ext-nacl)# deny tcp any any
Switch(config-ext-nacl)# permit ip any any
```

The ACLs are applied to permit Gigabit Ethernet port 0/1, which is configured as a Layer 2 port, with the Marketing_group ACL applied to incoming traffic.

```
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group marketing_group in
...
```

Including Comments About Entries in ACLs Example

The following example shows an IP numbered standard ACL using the **access-list** *access-list number* **remark** *remark* global configuration command to include a comment about an access list. In this example, the workstation belonging to Jones is allowed access, and the workstation belonging to Smith is not allowed access:

```
Switch(config)# access-list 1 remark Permit only Jones workstation through
Switch(config)# access-list 1 permit 171.69.2.88
Switch(config)# access-list 1 remark Do not allow Smith workstation through
Switch(config)# access-list 1 deny 171.69.3.13
```

The following example shows an entry in a named IP ACL using the **remark** access-list global configuration command to include a comment about an access list. In this example, the Jones subnet is not allowed to use outbound Telnet:

```
Switch(config)# ip access-list extended telnetting
Switch(config-ext-nacl)# remark Do not allow Jones subnet to telnet out
Switch(config-ext-nacl)# deny tcp host 171.69.2.88 any eq telnet
```

In this example of a numbered ACL, the workstation belonging to Jones is allowed access, and the workstation belonging to Smith is not allowed access:

```
Switch(config)# access-list 1 remark Permit only Jones workstation through
Switch(config)# access-list 1 permit 171.69.2.88
Switch(config)# access-list 1 remark Do not allow Smith workstation through
Switch(config)# access-list 1 deny 171.69.3.13
```

In this example of a numbered ACL, the Winter and Smith workstations are not allowed to browse the web:

```
Switch(config)# access-list 100 remark Do not allow Winter to browse the web
Switch(config)# access-list 100 deny host 171.69.3.85 any eq www
Switch(config)# access-list 100 remark Do not allow Smith to browse the web
Switch(config)# access-list 100 deny host 171.69.3.13 any eq www
```

Applying the ACL to an Interface Example

The following example shows how to apply access list 2 on Gigabit Ethernet interface 0/3 to filter packets entering the interface:

```
Switch(config)# interface gigabitethernet0/3
Router(config-if)# ip access-group 2 in
```

Displaying Standard and Extended ACLs Example

The following example displays all standard and extended ACLs:

```
Switch# show access-lists
Standard IP access list 1
    permit 172.20.10.10
Standard IP ACL 10
    permit 12.12.12.12
Standard IP access list 12
    deny   1.3.3.2
Standard IP access list 32
    permit 172.20.20.20
Standard IP access list 34
    permit 10.24.35.56
    permit 23.45.56.34
Extended IP access list 120
```

The following example displays only IP standard and extended ACLs:

```
Switch# show ip access-lists
Standard IP access list 1
    permit 172.20.10.10
Standard IP access list 10
    permit 12.12.12.12
Standard IP access list 12
    deny   1.3.3.2
Standard IP access list 32
    permit 172.20.20.20
Standard IP access list 34
    permit 10.24.35.56
    permit 23.45.56.34
Extended IP access list 120
```

Displaying Access Groups Example

You use the **ip access-group** interface configuration command to apply ACLs to a Layer 3 interface. When IP is enabled on an interface, you can use the **show ip interface** *interface-id* privileged EXEC command to view the input and output access lists on the interface, as well as other interface characteristics. If IP is not enabled on the interface, the access lists are not shown.

The following example shows how to view all access groups configured for VLAN 1 and for Gigabit Ethernet interface 0/2:

```
Switch# show ip interface vlan 1
GigabitEthernet0/2 is up, line protocol is down
  Internet address is 10.20.30.1/16
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is permit Any
  Inbound  access list is 13

<information truncated>

Switch# show ip interface f0/9
FastEthernet0/9 is down, line protocol is down
  Inbound  access list is ip1
```

The only way to ensure that you can view all configured access groups under all circumstances is to use the **show running-config** privileged EXEC command. To display the ACL configuration of a single interface, use the **show running-config interface** *interface-id* command.

The following example shows how to display the ACL configuration of Gigabit Ethernet interface 0/1:

```
Switch# show running-config interface gigabitethernet0/1
Building configuration...

Current configuration :112 bytes
!
interface GigabitEthernet0/1
 ip access-group 11 in
 snmp trap link-status
 no cdp enable
end!
```

Compiling ACLs Example

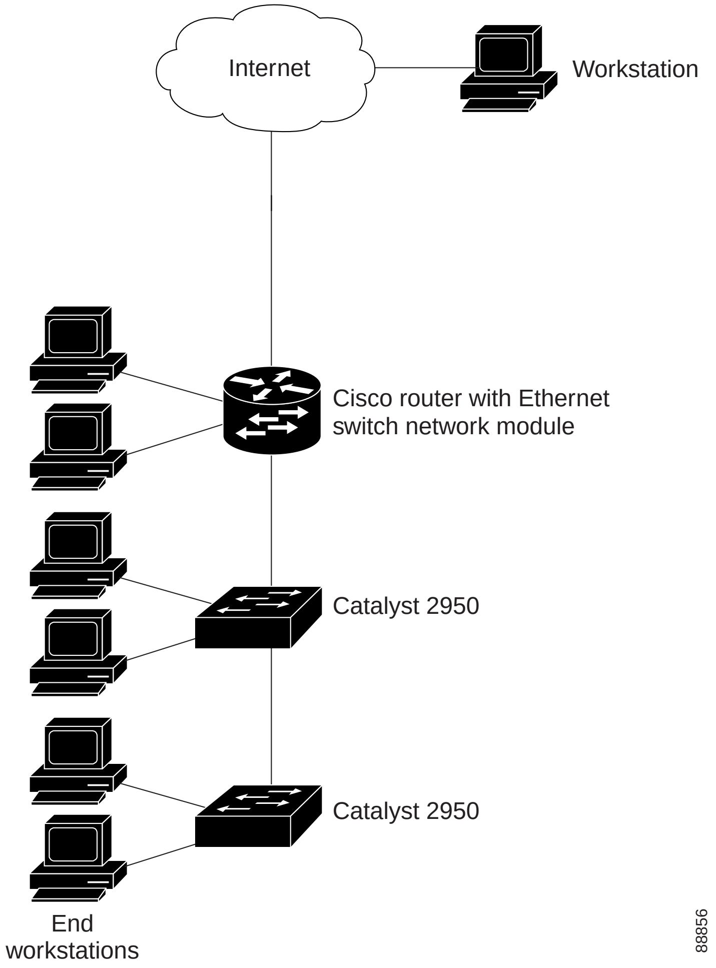
For detailed information about compiling ACLs, refer to the *Security Configuration Guide* and the “IP Services” chapter of the *Cisco IOS IP and IP Routing Configuration Guide* for Cisco IOS Release 12.2.

Figure 21 shows a small networked office with a stack of Catalyst 2950 switches that are connected to a Cisco router with an Ethernet switch network module installed. A host is connected to the network through the Internet using a WAN link.

Use switch ACLs to do these:

- Create a standard ACL, and filter traffic from a specific Internet host with an address 172.20.128.64.
- Create an extended ACL, and filter traffic to deny HTTP access to all Internet hosts but allow all other types of access.

Figure 21 Using Switch ACLs to Control Traffic



The following example uses a standard ACL to allow access to a specific Internet host with the address 172.20.128.64:

```
Switch(config)# access-list 6 permit 172.20.128.64 0.0.0.0
Switch(config)# end
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group 6 in
```

The following example uses an extended ACL to deny traffic from port 80 (HTTP). It permits all other types of traffic:

```
Switch(config)# access-list 106 deny tcp any any eq 80
Switch(config)# access-list 106 permit ip any any
Switch(config)# interface gigabitethernet0/2
Switch(config-if)# ip access-group 106 in
```

QoS Configuration Examples

Classifying Traffic by Using ACL Example

The following example shows how to allow access for only those hosts on the two specified networks. The wildcard bits apply to the host portions of the network addresses. Any host with a source address that does not match the ACL statements is rejected.

```
Switch(config)# access-list 1 permit 192.5.255.0 0.0.0.255
Switch(config)# access-list 1 permit 36.0.0.0 0.0.0.255
```

Classifying Traffic by Using Class Maps Example

The following example shows how to configure the class map called class1. The class1 has one match criterion, which is an ACL called 103.

```
Switch(config)# access-list 103 permit any any tcp eq 80
Switch(config)# class-map class1
Switch(config-cmap)# match access-group 103
Switch(config-cmap)# end
Switch#
```

Classifying, Policing, and Marking Traffic by Using Policy Maps Example

The following example shows how to create a policy map and attach it to an ingress interface. In the configuration, the IP standard ACL permits traffic from network 10.1.0.0. For traffic matching this classification, the DSCP value in the incoming packet is trusted. If the matched traffic exceeds an average traffic rate of 48000 bps and a normal burst size of 8000 bytes, its DSCP is marked down to a value of 10 and transmitted.

```
Switch(config)# access-list 1 permit 10.1.0.0 0.0.255.255
Switch(config)# class-map ipclass1
Switch(config-cmap)# match access-group 1
Switch(config-cmap)# exit
Switch(config)# policy-map flow1t
Switch(config-pmap)# class ipclass1
Switch(config-pmap-c)# police 5000000 8192 exceed-action dscp 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# service-policy input flow1t
```

Configuring the CoS-to-DSCP Map Example

The following example shows how to modify and display the CoS-to-DSCP map:

```
Switch# configure terminal
Switch(config)# mls qos map cos-dscp 8 8 8 8 24 32 56 56
Switch(config)# end
Switch# show mls qos maps cos-dscp

Cos-dscp map:
        cos:  0  1  2  3  4  5  6  7
     --------------------------------
       dscp:  8  8  8  8 24 32 56 56
```

Configuring the DSCP-to-CoS Map Example

The following example shows how the DSCP values 26 and 48 are mapped to CoS value 7. For the remaining DSCP values, the DSCP-to-CoS mapping is the default.

```
Switch(config)# mls qos map dscp-cos 26 48 to 7
Switch(config)# exit

Switch# show mls qos maps dscp-cos

Dscp-cos map:
       dscp:  0  8 10 16 18 24 26 32 34 40 46 48 56
     -----------------------------------------------
        cos:  0  1  1  2  2  3  7  4  4  5  5  7  7
```

Displaying QoS Information Example

The following example shows how to display the DSCP-to-CoS maps:

```
Switch# show mls qos maps dscp-cos

Dscp-cos map:
       dscp:  0  8 10 16 18 24 26 32 34 40 46 48 56
     -----------------------------------------------
        cos:  0  1  1  2  2  3  3  4  4  5  5  6  7
```

IGMP Snooping Example

Default IGMP Snooping Configuration

IGMP Snooping is enabled by default on a VLAN or subnet basis. Multicast routing has to be enabled on the router first and then PIM (Multicast routing protocol) has to be enabled on the VLAN interface so that the Ethernet switch network module acknowledges the IGMP join and leave messages that are sent from the hosts connected to the Ethernet switch network module.

```
Router(config)# ip multicast-routing
Router(config-if)# interface VLAN1
Router(config-if)# ip-address 192.168.10.1 255.255.255.0
Router(config-if)# ip pim sparse-mode
```

The following example shows the output from configuring IGMP snooping:

```
Router# show mac-address-table multicast igmp-snooping

Slot # :3
--------------
    MACADDR     VLANID     INTERFACES

0100.5e00.0001    1
0100.5e00.0002    1
0100.5e00.000d    1
0100.5e00.0016    1
0100.5e05.0505    1        Fa3/12
0100.5e06.0606    1        Fa3/13
0100.5e7f.ffff    1        Fa3/13
0100.5e00.0001    2
0100.5e00.0002    2
0100.5e00.000d    2
0100.5e00.0016    2
0100.5e00.0128    2
0100.5e05.0505    2        Fa3/10
0100.5e06.0606    2        Fa3/11
Router#
```

The following example shows output from the **show running-config interface** privileged EXEC command for VLAN 1:

```
Router# show running-config interface vlan 1

Building configuration...

Current configuration :82 bytes
!
interface Vlan1
 ip address 192.168.4.90 255.255.255.0
 ip pim sparse-mode
end
```

The following example shows output from the **show running-config interface** privileged EXEC command for VLAN 2:

```
Router# show running-config interface vlan 2

Building configuration...

Current configuration :82 bytes
!
interface Vlan2
 ip address 192.168.5.90 255.255.255.0
 ip pim sparse-mode
end
```

The following example shows output verifying multicasting support:

```
Router# show ip igmp group

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.255.255.255  Vlan1                    01:06:40  00:02:20  192.168.41.101
224.0.1.40       Vlan2                    01:07:50  00:02:17  192.168.5.90
224.5.5.5        Vlan1                    01:06:37  00:02:25  192.168.41.100
224.5.5.5        Vlan2                    01:07:40  00:02:21  192.168.31.100
224.6.6.6        Vlan1                    01:06:36  00:02:22  192.168.41.101
224.6.6.6        Vlan2                    01:06:39  00:02:20  192.168.31.101
```

The following example shows output from the multicast routing table:

```
Router# show ip mroute

IP Multicast Routing Table
Flags:D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C -
Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report
Outgoing interface flags:H - Hardware switched
Timers:Uptime/Expires
Interface state:Interface, Next-Hop or VCD, State/Mode

(*, 239.255.255.255), 01:06:43/00:02:17, RP 0.0.0.0, flags:DC
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan1, Forward/Sparse, 01:06:43/00:02:17

(*, 224.0.1.40), 01:12:42/00:00:00, RP 0.0.0.0, flags:DCL
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan2, Forward/Sparse, 01:07:53/00:02:14

(*, 224.5.5.5), 01:07:43/00:02:22, RP 0.0.0.0, flags:DC
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan1, Forward/Sparse, 01:06:40/00:02:22
    Vlan2, Forward/Sparse, 01:07:44/00:02:17

(*, 224.6.6.6), 01:06:43/00:02:18, RP 0.0.0.0, flags:DC
  Incoming interface:Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan1, Forward/Sparse, 01:06:40/00:02:18
    Vlan2, Forward/Sparse, 01:06:43/00:02:16
```

Storm-Control Example

The following example shows global bandwidth-based multicast suppression being enabled at 70 percent on Gigabit Ethernet interface 1 and the configuration being verified:

```
Router# configure terminal
Router(config)# interface gigabitethernet0/2
Router(config-if)# storm-control threshold 70
Router(config-if)# end
Router# show storm-control

Name: Gi0/2
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001

Port Protected: Off
Unknown Unicast Traffic: Allowed
Unknown Multicast Traffic: Not Allowed
```

```
Broadcast Suppression Level: 100
Multicast Suppression Level: 70
Unicast Suppression Level: 100
```

Ethernet Switching Examples

Subnets for Voice and Data Example

The following example shows separate subnets being configured for voice and data on the Ethernet switch network module:

```
interface FastEthernet5/1
     description DOT1Q port to IP Phone
     switchport native vlan 50
     switchport mode trunk
     switchport voice vlan 150

interface Vlan 150
     description voice vlan
     ip address 10.150.1.1 255.255.255.0
     ip helper-address 172.20.73.14 (See Note below)

interface Vlan 50
     description data vlan
     ip address 10.50.1.1 255.255.255.0
```

This configuration instructs the IP phone to generate a packet with an 802.1Q VLAN ID of 150 with an 802.1p value of 5 (default for voice bearer traffic).

Note In a centralized CallManager deployment model, the DHCP server might be located across the WAN link. If so, an **ip helper-address** command pointing to the DHCP server should be included on the voice VLAN interface for the IP phone. This is done to obtain its IP address as well as the address of the TFTP server required for its configuration.

Cisco IOS supports a DHCP server function. If this function is used, the Ethernet switch network module serves as a local DHCP server and a helper address would not be required.

Inter-VLAN Routing Example

Configuring inter-VLAN routing is identical to the configuration on a Ethernet switch network module with an MSFC. Configuring an interface for WAN routing is consistent with other Cisco IOS platforms.

The following example provides a sample configuration:

```
interface Vlan 160
     description voice vlan
     ip address 10.6.1.1 255.255.255.0
```

```
interface Vlan 60
     description data vlan
     ip address 10.60.1.1 255.255.255.0

interface Serial1/0
     ip address 160.3.1.2 255.255.255.0
```

Note Standard IGP routing protocols such as RIP, IGRP, EIGRP, and OSPF are supported on the Ethernet switch network module. Multicast routing is also supported for PIM dense mode, sparse mode, and sparse-dense mode.

Single Subnet Configuration Example

The Ethernet switch network module supports the use of an 802.1p-only option when configuring the voice VLAN. Using this option allows the IP phone to tag VoIP packets with a CoS of 5 on the native VLAN, while all PC data traffic is sent untagged.

The following example shows a single subnet configuration for the Ethernet switch network module switch:

```
Router# FastEthernet 5/2
description Port to IP Phone in single subnet
     switchport access vlan 40
     switchport voice vlan dot1p
     spanning-tree portfast
```

The Ethernet switch network module instructs the IP phone to generate an 802.1Q frame with a null VLAN ID value but with an 802.1p value (default is COS of 5 for bearer traffic). The voice and data VLANs are both 40 in this example.

Ethernet Ports on IP Phones with Multiple Ports Example

The following example illustrates the configuration on the IP phone:

```
interface FastEthernet2/2
     switchport voice vlan 5
     switchport mode trunk
```

The following example illustrates the configuration on the PC:

```
interface FastEthernet2/3
     switchport access vlan 10
```

Note Using a separate subnet, and possibly a separate IP address space, may not be an option for some small branch offices due to the IP routing configuration. If the IP routing can handle an additional subnet at the remote branch, you can use Cisco Network Registrar and secondary addressing.

Intrachassis Stacking Example

The following example shows how to stack GE port 2/0 to GE port 3/0 to form an extended VLAN within one chassis:

```
Router #config terminal
Router(config)# interface Gigabit 2/0
Router(config-if)# switchport stacking-link interface Gigabit3/0
```

The following example shows interchassis stacking being verified between GE port 2/0 and GE port 3/0:

```
Router# show interface gigabit 2/0


 GigabitEthernet2/0 is up, line protocol is down
   Internal Stacking Link Active : Gi2/0 is stacked with Gi3/0
   Hardware is Gigabit Ethernet, address is 001b.3f2b.2c24 (bia 001b.3f2b.2c24)
   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)
   Full-duplex mode, link type is force-up, media type is unknown 0
   output flow-control is off, input flow-control is off
   Full-duplex, 1000Mb/s
   ARP type: ARPA, ARP Timeout 04:00:00
   Last input 1d22h, output never, output hang never
   Last clearing of "show interface" counters 1d22h
   Queueing strategy: fifo
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec
      250707 packets input, 19562597 bytes, 0 no buffer
      Received 7 broadcasts, 0 runts, 0 giants, 0 throttles
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
      0 watchdog, 0 multicast, 0 pause input
      0 input packets with dribble condition detected
      7469804 packets output, 582910831 bytes, 0 underruns(0/0/0)
      0 output errors, 0 collisions, 0 interface resets
      0 babbles, 0 late collision, 0 deferred
      0 lost carrier, 0 no carrier, 0 pause output
      0 output buffer failures, 0 output buffers swapped out
```

Flow Control on Gigabit Ethernet Ports Example

The following examples show how to turn transmit and receive flow control on and how to verify the flow-control configuration:

Port 4/0 flow control send administration status set to on (port will send flowcontrol to far end):

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet4/0
Switch(config-if)# flowcontrol send on
Switch(config-if)# end
```

Port 4/0 flow control receive administration status set to on (port will require far end to send flowcontrol):

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet4/0
Switch(config-if)# flowcontrol receive on
Switch(config-if)# end
```

The following example shows flow control configuration being verified:

```
Switch# show interface gigabitethernet4/0
GigabitEthernet4/0 is up, line protocol is up
  Hardware is Gigabit Ethernet, address is 0087.c08b.4824 (bia
0087.c08b.4824)
  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)
  output flow-control is off, input flow-control is on
  0 pause input, 0 pause output
  Full-duplex, 1000Mb/s
  ARP type:ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output never, output hang never
  Last clearing of "show interface" counters never
  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, 1 packets/sec
     398301 packets input, 29528679 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
     790904 packets output, 54653461 bytes, 0 underruns
     0 output errors, 0 collisions, 5 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

The following example shows how to configure Gigabit Ethernet interface 0/10 as a routed port and to assign it an IP address:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet0/10
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.1.2.3 255.255.0.0
Switch(config-if)# no shutdown
Switch(config-if)# end
```

The following is sample output from the **show interfaces** privileged EXEC command for Gigabit Ethernet interface 0/2:

```
Switch(config)# show interfaces gigabitethernet0/2
GigabitEthernet0/2 is up, line protocol is up
  Hardware is Gigabit Ethernet, address is 0002.4b29.4400 (bia 0002.4b29.4400)
  Internet address is 192.20.135.21/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s
  input flow-control is off, output flow-control is off
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:02, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     89604 packets input, 8480109 bytes, 0 no buffer
     Received 81848 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
     60665 packets output, 6029820 bytes, 0 underruns
     0 output errors, 0 collisions, 16 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

The following is sample output from the **show ip interface** privileged EXEC command for Gigabit Ethernet interface 0/2:

```
Switch# show ip interface gigabitethernet0/2
GigabitEthernet0/2 is up, line protocol is up
  Internet address is 192.20.135.21/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.5 224.0.0.6
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP CEF Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

The following is sample output for the **show running-config** privileged EXEC command for Gigabit Ethernet interface 0/2:

```
Switch# show running-config interface gigabitethernet0/2
Building configuration...

Current configuration : 122 bytes
!
interface GigabitEthernet0/2
 no switchport
 ip address 192.20.135.21 255.255.255.0
 speed 100
 mls qos trust dscp
end
```

Configuring Layer 3 Interfaces Example

The following example shows how to configure Gigabit Ethernet interface 0/10 as a routed port and to assign it an IP address:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet0/10
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.1.2.3 255.255.0.0
Switch(config-if)# no shutdown
Switch(config-if)# end
```

The following is sample output from the **show interfaces** privileged EXEC command for Gigabit Ethernet interface 0/2:

```
Switch(config)# show interfaces gigabitethernet0/2
GigabitEthernet0/2 is up, line protocol is up
  Hardware is Gigabit Ethernet, address is 0002.4b29.4400 (bia 0002.4b29.4400)
  Internet address is 192.20.135.21/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s
  input flow-control is off, output flow-control is off
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:02, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     89604 packets input, 8480109 bytes, 0 no buffer
     Received 81848 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
     60665 packets output, 6029820 bytes, 0 underruns
     0 output errors, 0 collisions, 16 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

The following is sample output from the **show ip interface** privileged EXEC command for Gigabit Ethernet interface 0/2:

```
Switch# show ip interface gigabitethernet0/2
GigabitEthernet0/2 is up, line protocol is up
  Internet address is 192.20.135.21/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.5 224.0.0.6
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP CEF Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Probe proxy name replies are disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

The following is sample output for the **show running-config** privileged EXEC command for Gigabit Ethernet interface 0/2:

```
Switch# show running-config interface gigabitethernet0/2
Building configuration...

Current configuration : 122 bytes
!
interface GigabitEthernet0/2
 no switchport
 ip address 192.20.135.21 255.255.255.0
 speed 100
 mls qos trust dscp
end
```

Fallback Bridging Example

This section describes how to configure fallback bridging on your switch. It contains this configuration information:

- Creating a Bridge Group Example, page 155
- Preventing the Forwarding of Dynamically Learned Stations Example, page 155
- Configuring the Bridge Table Aging Time Example, page 155
- Filtering Frames by a Specific MAC Address Example, page 155
- Adjusting Spanning-Tree Parameters Examples, page 155

Creating a Bridge Group Example

The following example shows how to create bridge group 10, specify the VLAN-bridge STP to run in the bridge group, and assign an interface to the bridge group:

```
Switch(config)# bridge 10 protocol vlan-bridge
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# no switchport
Switch(config-if)# bridge-group 10
```

Preventing the Forwarding of Dynamically Learned Stations Example

The following example shows how to prevent the switch from forwarding frames for stations that it has dynamically learned in bridge group 10:

```
Switch(config)# no bridge 10 acquire
```

Configuring the Bridge Table Aging Time Example

The following example shows how to change the bridge table aging time to 200 seconds for bridge group 10:

```
Switch(config)# bridge 10 aging-time 200
```

Filtering Frames by a Specific MAC Address Example

The following example shows how to forward a frame with MAC address 0800.cb00.45e9 through an interface in bridge group 1:

```
Switch(config)# bridge 1 address 0800.cb00.45e9 forward gigabitethernet0/1
```

Adjusting Spanning-Tree Parameters Examples

The following examples show how to adjust spanning-tree parameters:

- Changing the Switch Priority Example, page 156
- Changing the Interface Priority Example, page 156
- Assigning a Path Cost Example, page 156
- Adjusting BPDU Intervals Example, page 156
- Disabling the Spanning Tree on an Interface Example, page 156

Changing the Switch Priority Example

The following example shows how to set the switch priority to 100 for bridge group 10:

```
Switch(config)# bridge 10 priority 100
```

Changing the Interface Priority Example

The following example shows how to change the priority of an interface to 20 in bridge group 10:

```
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# bridge-group 10 priority 20
```

Assigning a Path Cost Example

The following example shows how to change the path cost on an interface to 10 in bridge group 10:

```
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# bridge-group 10 path-cost 20
```

Adjusting BPDU Intervals Example

You can adjust BPDU intervals as described in these sections:

- Adjusting the Interval between Hello BPDUs Example, page 156
- Changing the Forward-Delay Interval Example, page 156
- Changing the Maximum-Idle Interval Example, page 156

Adjusting the Interval between Hello BPDUs Example

The following example shows how to change the hello interval to 5 seconds in bridge group 10:

```
Switch(config)# bridge 10 hello-time 5
```

Changing the Forward-Delay Interval Example

The following example shows how to change the forward-delay interval to 10 seconds in bridge group 10:

```
Switch(config)# bridge 10 forward-time 10
```

Changing the Maximum-Idle Interval Example

The following example shows how to change the maximum-idle interval to 30 seconds in bridge group 10:

```
Switch(config)# bridge 10 max-age 30
```

Disabling the Spanning Tree on an Interface Example

The following example shows how to disable spanning tree on an interface in bridge group 10:

```
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# bridge group 10 spanning-disabled
```

Command Reference

This section documents new commands or existing commands that are newly ported to the 16- and 36-port Ethernet switch module. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command reference publications.

- **aaa authentication dot1x**
- **class**
- **class-map**
- **debug dot1x**
- **debug eswilp**
- **debug ip igmp snooping**
- **debug spanning-tree**
- **deny (access-list configuration)**
- **dot1x default**
- **dot1x max-req**
- **dot1x multiple-hosts**
- **dot1x port-control**
- **dot1x re-authenticate**
- **dot1x re-authentication**
- **dot1x timeout quiet-period**
- **dot1x timeout re-authperiod**
- **dot1x timeout tx-period**
- **ip access-group**
- **ip access-list**
- **ip igmp snooping**
- **ip igmp snooping vlan**
- **ip igmp snooping vlan immediate-leave**
- **ip igmp snooping vlan mrouter**
- **ip igmp snooping vlan static**
- **match (class-map configuration)**
- **mls qos cos**
- **mls qos map**
- **mls qos trust**
- **permit (access-list configuration)**
- **police**
- **policy-map**
- **service-policy**
- **show access-lists**

- **show class-map**
- **show dot1x**
- **show ip access-lists**
- **show ip igmp snooping**
- **show ip igmp snooping mrouter**
- **show mls masks**
- **show mls qos interface**
- **show mls qos maps**
- **show policy-map**
- **show spanning-tree**
- **show storm-control**
- **spanning-tree backbonefast**
- **storm-control**
- **switchport**

aaa authentication dot1x

To specify one or more authentication, authorization, and accounting (AAA) methods for use on interfaces running IEEE 802.1x, use the **aaa authentication dot1x** command in global configuration mode. To disable authentication, use the **no** form of this command.

aaa authentication dot1x {**default** | *listname*} *method1* [*method2...*]

no aaa authentication dot1x {**default** | *listname*} *method1* [*method2...*]

Syntax Description

default	Uses the listed authentication methods that follow this argument as the default list of methods when a user logs in.
listname	Character string used to name the list of authentication methods tried when a user logs in.
method1 [*method2...*]	At least one of these keywords: • **enable**—Uses the enable password for authentication. • **group radius**—Uses the list of all Remote Authentication Dial-In User Service (RADIUS) servers for authentication. • **line—**Uses the line password for authentication. • **local—**Uses the local username database for authentication. • **local-case—**Uses the case-sensitive local username database for authentication. • **none—**Uses no authentication. The client is automatically authenticated by the switch without using the information supplied by the client.

Defaults

No authentication is performed.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

The *method* argument identifies the list of methods that the authentication algorithm tries in the given sequence to validate the password provided by the client. The only method that is truly 802.1x-compliant is the **group radius** method, in which the client data is validated against a RADIUS authentication server. The remaining methods enable AAA to authenticate the client by using locally configured data. For example, the **local** and **local-case** methods use the username and password that are saved in the Cisco IOS configuration file. The **enable** and **line** methods use the **enable** and **line** passwords for authentication.

If you specify **group radius**, you must configure the RADIUS server by entering the **radius-server host** global configuration command.

If you are not using a RADIUS server, you can use the **local** or **local-case** methods, which access the local username database to perform authentication. By specifying the **enable** or **line** methods, you can supply the clients with a password to provide access to the switch.

Use the **show running-config** privileged EXEC command to display the configured lists of authentication methods.

Examples

The following example shows how to enable AAA and how to create an authentication list for 802.1x. This authentication first tries to contact a RADIUS server. If this action returns an error, the user is allowed access with no authentication:

```
Switch(config)# aaa new model
Switch(config)# aaa authentication dot1x default group radius none
```

You can verify your settings by entering the **show running-config** privileged EXEC command.

Related Commands

Command	Description
aaa new-model	Enables the AAA access control model.
show running-config	Displays the running configuration on the switch.

class

To define a traffic classification for the policy to act on using the class-map name or access group, use the **class** policy-map configuration command. To delete an existing class map, use the **no** form of this command.

class *class-map-name* [**access-group** *acl-index-or-name*]

no class *class-map-name*

Syntax Description

class-map-name	Name of the class map.
access-group *acl-index-or-name*	(Optional) Number or name of an IP standard or extended access control list (ACL). For an IP standard ACL, the index range is 1 to 99 and 1300 to 1999; for an IP extended ACL, the index range is 100 to 199 and 2000 to 2699.

Defaults

No policy-map class maps are defined.

Command Modes

Policy-map configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Before you use the **class** command, use the **policy-map** global configuration command to identify the policy map and to enter policy-map configuration mode. After you specify a policy map, you can configure a policy for new classes or modify a policy for any existing classes in that policy map. You attach the policy map to an interface by using the **service-policy** interface configuration command; however, you cannot attach one that uses an ACL classification to the egress direction.

The class name that you specify in the policy map ties the characteristics for that class to the class map and its match criteria as configured by using the **class-map** global configuration command.

The **class** command performs the same function as the **class-map** global configuration command. Use the **class** command when a new classification, which is not shared with any other ports, is needed. Use the **class-map** command when the map is shared among many ports.

Note In a policy map, the class named class-default is not supported. The switch does not filter traffic based on the policy map defined by the **class class-default** policy-map configuration command.

After entering the **class** command, you enter policy-map class configuration mode. When you are in this mode, these configuration commands are available:

- **default**: sets a command to its default.

- **exit**: exits policy-map class configuration mode and returns to policy-map configuration mode.
- **no**: returns a command to its default setting.
- **police**: defines a policer for the classified traffic. The policer specifies the bandwidth limitations and the action to take when the limits are exceeded. For more information, see the **police** command.

To return to policy-map configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.

Note For more information about configuring IP ACLs, refer to the "Configuring IP Services" chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Examples

The following example shows how to create a policy map named policy1. When attached to the ingress port, it matches all the incoming traffic defined in class1 and polices the traffic at an average rate of 1 Mbps and bursts at 131072 bytes. Traffic exceeding the profile is dropped:

```
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# police 1000000 131072 exceed-action drop
Switch(config-pmap-c)# exit
Switch(config-pmap)#
```

You can verify your settings by entering the **show policy-map** privileged EXEC command.

Related Commands

Command	Description
class-map	Creates a class map to be used for matching packets to the class whose name you specify.
match (class-map configuration)	Defines the match criteria to classify traffic.
policy-map	Creates or modifies a policy map that can be attached to multiple interfaces to specify a service policy.
show policy-map	Displays QoS policy maps.

class-map

To create a class map to be used for matching packets and to enter class-map configuration mode, use the **class-map** command in global configuration mode. To delete an existing class map, use the **no** form of this command.

class-map *class-map-name*

no class-map *class-map-name*

Syntax Description

class-map-name	Name of the class map.

Defaults

No class maps are defined.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use this command to specify the name of the class for which you want to create or modify class-map match criteria and to enter class-map configuration mode. In this mode, you can enter one **match** command to configure the match criteria for this class.

The **class-map** command and its subcommands are used to define packet classification and marking as part of a globally named service policy applied on a per-interface basis.

In quality of service (QoS) class-map configuration mode, these configuration commands are available:

- **exit**: exits from QoS class-map configuration mode.
- **no**: removes a match statement from a class map.
- **match**: configures classification criteria. For more information, see the **match** class-map configuration command.

Only one match criteria per class map is supported. For example, when defining a class map, only one **match** command can be entered.

Only one access control list (ACL) can be configured in a class map. The ACL can have multiple access control entries (ACEs).

Note The switch does not support any deny conditions in an ACL configured in a class map.

Note For more information about configuring IP ACLs, refer to the “Configuring IP Services” chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Examples

The following example shows how to configure the class map named class1. Class1 has one match criteria, which is a numbered ACL:

```
Switch(config)# access-list 103 permit tcp any any eq 80
Switch(config)# class-map class1
Switch(config-cmap)# match access-group 103
Switch(config-cmap)# exit
```

You can verify your settings by entering the **show class-map** privileged EXEC command.

Related Commands

Command	Description
class	Defines a traffic classification for the policy to act on by using the class-map name or access group.
match (class-map configuration)	Defines the match criteria to classify traffic.
policy-map	Creates or modifies a policy map that can be attached to multiple interfaces to specify a service policy.
show class-map	Displays QoS class maps.

debug dot1x

To enable debugging of the 802.1x feature, use the **debug dot1x** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug dot1x {**all** | **authsm** | **backend** | **besm** | **core** | **reauthsm**}

no debug dot1x {**all** | **authsm** | **backend** | **besm** | **core** | **reauthsm**}

Syntax Description

all	Enables debugging of all conditions.
authsm	Enables debugging of the authenticator state machine, which is responsible for controlling access to the network through 802.1x-enabled ports.
backend	Enables debugging of the interaction between the 802.1x process and the switch (Remote Authentication Dial-In User Service [RADIUS] client).
besm	Enables debugging of the backend state machine, which is responsible for relaying authentication request between the client and the authentication server.
core	Enables debugging of the 802.1x process, which includes 802.1x initialization, configuration, and the interaction with the port manager module.
reauthsm	Enables debugging of the reauthentication state machine, which manages periodic reauthentication of the client.

Defaults

Debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

The **undebug dot1x** command is the same as the **no debug dot1x** command.

Related Commands

Command	Description
show debugging	Displays information about the types of debugging that are enabled.
show dot1x	Displays 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

debug eswilp

To enable debugging of Ethernet switch network module features, use the **debug eswilp** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug eswilp {**dot1x** | **filtermgr** | **fltdrv** | **igmp** | **port-driver** | **power-supply** | **span** | **switch-pm**}

no debug eswilp {**dot1x** | **filtermgr** | **fltdrv** | **igmp** | **port-driver** | **power-supply** | **span** | **switch-pm**}

Syntax Description

dot1x	Displays ESWILP 802.1x debugging messages.
filtermgr	Displays ESWILP filter manager debugging messages.
fltdrv	Displays ESWILP filter driver debugging messages.
igmp	Displays ESWILP IGMP debugging messages.
port-driver	Displays ESWILP port driver debugging messages.
power-supply	Displays ESWILP power supply information debugging messages.
span	Displays ESWILP SPAN debugging messages.
switch-pm	Displays ESWILP switch port manager debugging messages.

Defaults

Debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers. The **dot1x**, **filtermgr**, and **fltdrv** keywords were added.

Usage Guidelines

The **undebug eswilp** command is the same as the **no debug eswilp** command.

Examples

The following example shows debugging messages for the IGMP snooping services on the Ethernet switch network module being displayed:

```
Router# debug eswilp igmp
```

Related Commands

Command	Description
show debugging	Displays information about the types of debugging that are enabled.

debug ip igmp snooping

To display debugging messages about Internet Group Management Protocol (IGMP) snooping services, use the **debug ip igmp snooping** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip igmp snooping {**group** | **management** | **router** | **timer**}

no debug ip igmp snooping {**group** | **management** | **router** | **timer**}

Syntax Description

group	Displays debugging messages related to multicast groups.
management	Displays debugging messages related to IGMP management services.
router	Displays debugging messages related to the local router.
timer	Displays debugging messages related to the IGMP timer.

Defaults

Debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use the **debug ip igmp snooping** command to troubleshoot the IGMP snooping feature.

Examples

The following example shows debugging messages for the IGMP snooping services being displayed:

```
Router# debug ip igmp snooping

IGMP snooping enabled
```

Related Commands

Command	Description
show ip igmp snooping	Displays the IGMP snooping configuration.

debug spanning-tree

To debug spanning-tree activities, use the **debug spanning-tree** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug spanning-tree {**all** | **backbonefast** | **bpdu** | **bpdu-opt** | **config** | **etherchannel** | **events** | **exceptions** | **general** | **pvst+** | **root** | **snmp** | **uplinkfast**}

no debug spanning-tree {**all** | **backbonefast** | **bpdu** | **bpdu-opt** | **config** | **etherchannel** | **events** | **exceptions** | **general** | **pvst+** | **root** | **snmp** | **uplinkfast**}

Syntax Description

all	Displays all spanning-tree debugging messages.
backbonefast	Displays debugging messages for BackboneFast events.
bpdu	Displays debugging messages for spanning-tree Bridge Protocol Data Units (BPDUs).
bpdu-opt	Displays debugging messages for optimized BPDU handling.
config	Displays debugging messages for spanning-tree configuration changes.
etherchannel	Displays debugging messages for EtherChannel support.
events	Displays debugging messages for spanning-tree topology events.
exceptions	Displays debugging messages for spanning-tree exceptions.
general	Displays debugging messages for general spanning-tree activity.
pvst+	Displays debugging messages for per-VLAN Spanning Tree Plus (PVST+) events.
root	Displays debugging messages for spanning-tree root events.
snmp	Displays debugging messages for spanning-tree Simple Network Management Protocol (SNMP) handling.
uplinkfast	Displays debugging messages for UplinkFast events.

Defaults

Debugging is disabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

The **undebug spanning-tree** command is the same as the **no debug spanning-tree** command.

Related Commands

Command	Description
show debugging	Displays information about the types of debugging that are enabled.
show spanning-tree	Displays spanning-tree state information.

deny (access-list configuration)

To configure conditions for a named or numbered IP access control list (ACL), use the **deny** command in access-list configuration mode. To remove a deny condition from the IP ACL, use the **no** form of the command.

Use these commands with standard IP ACLs:

deny {*source source-wildcard* | **host** *source* | **any**}

no deny {*source source-wildcard* | **host** *source* | **any**}

Use these commands with extended IP ACLs:

deny *protocol* {*source source-wildcard* | **host** *source* | **any**} [*operator port*] {*destination destination-wildcard* | **host** *source* | **any**} [*operator port*]

no deny *protocol* {*source source-wildcard* | **host** *source* | **any**} [*operator port*] {*destination destination-wildcard* | **host** *source* | **any**} [*operator port*]

Syntax Description

source source-wildcard | **host** *source* | **any**	Source IP address and wildcard. The *source* is the source address of the network or host from which the packet is being sent, specified in one of these ways: • The 32-bit quantity in dotted-decimal format. The *source-wildcard* applies wildcard bits to the source. • The keyword **host**, followed by the 32-bit quantity in dotted-decimal format, as an abbreviation for source and source-wildcard of *source* 0.0.0.0. • The keyword **any** as an abbreviation for *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. You do not need to enter a source-wildcard.
protocol	Name of an IP protocol. *protocol* can be **ip**, **tcp**, or **udp**.
destination destination-wildcard | **host** *source* | **any**	Destination IP address and wildcard. The *destination* is the destination address of the network or host to which the packet is being sent, specified in one of these ways: • The 32-bit quantity in dotted-decimal format. The *destination-wildcard* applies wildcard bits to the destination. • The keyword **host**, followed by the 32-bit quantity in dotted-decimal format, as an abbreviation for source and source-wildcard of *source* 0.0.0.0. • The keyword **any** as an abbreviation for *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. You do not need to enter a destination-wildcard.

operator port	(Optional) Source or destination port. The *operator* can be only **eq** (equal). If *operator* is after the source IP address and wildcard, conditions match when the source port matches the defined port. If *operator* is after the destination IP address and wildcard, conditions match when the destination port matches the defined port. The *port* is a decimal number or name of a Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port. The number can be from 0 to 65535. Use TCP port names only for TCP traffic. Use UDP port names only for UDP traffic.

Defaults

There are no specific conditions that deny packets in the named or numbered IP ACL.

The default ACL is always terminated by an implicit deny statement for all packets.

Command Modes

Access-list configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use this command after the **ip access-list** global configuration command to specify deny conditions for an IP ACL. You can specify a source IP address, destination IP address, IP protocol, TCP port, or UDP port. Specify the TCP and UDP port numbers only if *protocol* is **tcp** or **udp** and *operator* is **eq**.

Note For more information about configuring IP ACLs, refer to the “Configuring IP Services” chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Examples

The following example shows how to create an extended IP ACL and to configure deny conditions for it:

```
Switch(config)# ip access-list extended Internetfilter
Switch(config-ext-nacl)# deny tcp host 190.5.88.10 any
Switch(config-ext-nacl)# deny tcp host 192.1.10.10 any
```

The following is an example of a standard ACL that sets a deny conditions:

```
ip access-list standard Acclist1
 deny 192.5.34.0  0.0.0.255
 deny 128.88.10.0  0.0.0.255
 deny 36.1.1.0  0.0.0.255
```

Note In these examples, all other IP access is implicitly denied.

You can verify your settings by entering the **show ip access-lists** or **show access-lists** privileged EXEC command.

Related Commands

Command	Description
dot1x re-authenticate	Controls access to an interface.
ip access-list	Defines an IP ACL.
permit (access-list configuration)	Sets conditions for an IP ACL.
show access-lists	Displays ACLs configured on a switch.
show ip access-lists	Displays IP ACLs configured on the switch.

dot1x default

To reset the global 802.1x parameters to their default values, use the **dot1x default** command in global configuration mode.

dot1x default

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default setting.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Examples

The following example shows how to reset the global 802.1x parameters:

```
Switch(config)# dot1x default
```

You can verify your settings by entering the **show dot1x** privileged EXEC command.

Related Commands

Command	Description
dot1x max-req	Sets the maximum number of times that the switch sends an EAP-request/identity frame before restarting the authentication process.
dot1x re-authentication	Enables periodic reauthentication of the client.
dot1x timeout quiet-period	Sets the number of seconds that the switch remains in the quiet state following a failed authentication exchange.
dot1x timeout re-authperiod	Sets the number of seconds between reauthentication attempts.
dot1x timeout tx-period	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request.
show dot1x	Displays the 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

dot1x max-req

To set the maximum number of times that the switch sends an Extensible Authentication Protocol (EAP)-request/identity frame (assuming that no response is received) before restarting the authentication process, use the **dot1x max-req** command in global configuration mode. To return to the default setting, use the **no** form of this command.

dot1x max-req *count*

no dot1x max-req

Syntax Description

count	Number of times that the switch sends an EAP-request/identify frame before restarting the authentication process. The range is 1 to 10.

Defaults

The default is 2 times.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

Examples

The following example shows how to set the number of times that the switch sends an EAP-request/identity frame to 5 before restarting the authentication process:

```
Switch(config)# dot1x max-req 5
```

You can verify your settings by entering the **show dot1x** privileged EXEC command.

Related Commands

Command	Description
dot1x timeout tx-period	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request.
show dot1x	Displays the 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

dot1x multiple-hosts

To allow multiple hosts (clients) on an 802.1x-authorized port that has the **dot1x port-control** interface configuration command set to **auto**, use the **dot1x multiple-hosts** command in interface configuration mode. To return to the default setting, use the **no** form of this command.

dot1x multiple-hosts

no dot1x multiple-hosts

Syntax Description

This command has no arguments or keywords.

Defaults

Multiple hosts are disabled.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

This command enables you to attach multiple clients to a single 802.1x-enabled port. In this mode, only one of the attached hosts must be successfully authorized for all hosts to be granted network access. If the port becomes unauthorized (reauthentication fails, or an Extensible Authentication Protocol over LAN [EAPOL]-logoff message is received), all attached clients are denied access to the network.

Examples

The following example shows how to enable 802.1x on Fast Ethernet interface 0/1 and to allow multiple hosts:

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x multiple-hosts
```

You can verify your settings by entering the **show dot1x** [**interface** *interface-id*] privileged EXEC command.

Related Commands

Command	Description
dot1x default	Enables manual control of the authorization state of the port.
show dot1x	Displays 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

dot1x port-control

To enable manual control of the authorization state of the port, use the **dot1x port-control** command in interface configuration mode. To return to the default setting, use the **no** form of this command.

dot1x port-control {**auto** | **force-authorized** | **force-unauthorized**}

no dot1x port-control

Syntax Descriptionn

auto	Enables 802.1x on the interface and cause the port to change to the authorized or unauthorized state based on the 802.1x authentication exchange between the switch and the client.
force-authorized	Disables 802.1x on the interface and cause the port to change to the authorized state without any authentication exchange required. The port transmits and receives normal traffic without 802.1x-based authentication of the client.
force-unauthorized	Denies all access through this interface by forcing the port to change to the unauthorized state, ignoring all attempts by the client to authenticate. The switch cannot provide authentication services to the client through the interface.

Defaults

The authorization state is force-authorized.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

The 802.1x protocol is supported on Layer 2 static-access ports.

You can use the **auto** keyword only if the port is not configured as one of these types:

- Trunk port—If you try to enable 802.1x on a trunk port, an error message appears, and 802.1x is not enabled. If you try to change the mode of an 802.1x-enabled port to trunk, the port mode is not changed.
- EtherChannel port—Before enabling 802.1x on the port, you must first remove it from the EtherChannel. If you try to enable 802.1x on an EtherChannel or on an active port in an EtherChannel, an error appears, and 802.1x is not enabled. If you enable 802.1x on a not-yet active port of an EtherChannel, the port does not join the EtherChannel.
- Switch Port Analyzer (SPAN) destination port—You can enable 802.1x on a port that is a SPAN destination port; however, 802.1x is disabled until the port is removed as a SPAN destination. You can enable 802.1x on a SPAN source port.

To globally disable 802.1x on the switch, you must disable it on each port. There is no global configuration command for this task.

Examples

The following example shows how to enable 802.1x on Fast Ethernet interface 0/1:

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# dot1x port-control auto
```

You can verify your settings by entering the **show dot1x** privileged EXEC command and checking the Status column in the 802.1x Port Summary section of the display. An enabled status means the port-control value is set to **auto** or to **force-unauthorized**.

Related Commands

Command	Description
show dot1x	Displays 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

dot1x re-authenticate

To manually initiate a reauthentication of all 802.1x-enabled ports or the specified 802.1x-enabled port, use the **dot1x re-authenticate** command in privileged EXEC mode.

dot1x re-authenticate [**interface** *interface-id*]

Syntax Description

interface *interface-id*	(Optional) Slot and port number of the interface to reauthenticate.

Defaults

There is no default setting.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

You can use this command to reauthenticate a client without waiting for the configured number of seconds between reauthentication attempts (reauthperiod) and automatic reauthentication.

Examples

The following example shows how to manually reauthenticate the device connected to Fast Ethernet interface 0/1:

```
Switch# dot1x re-authenticate interface fastethernet 0/1
Starting reauthentication on FastEthernet0/1.
```

You can verify your settings by entering the **show dot1x** privileged EXEC command.

dot1x re-authentication

To enable periodic reauthentication of the client, use the **dot1x re-authentication** command in global configuration mode. To return to the default setting, use the **no** form of this command.

dot1x re-authentication

no dot1x re-authentication

Syntax Description

This command has no arguments or keywords.

Defaults

Periodic reauthentication is disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

You configure the amount of time between periodic reauthentication attempts by using the **dot1x timeout re-authperiod** global configuration command.

Examples

The following example shows how to disable periodic reauthentication of the client:

```
Switch(config)# no dot1x re-authentication
```

The following example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000 seconds:

```
Switch(config)# dot1x re-authentication
Switch(config)# dot1x timeout re-authperiod 4000
```

You can verify your settings by entering the **show dot1x** privileged EXEC command.

Related Commands

Command	Description
dot1x timeout re-authperiod	Sets the number of seconds between reauthentication attempts.
show dot1x	Displays 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

dot1x timeout quiet-period

To set the number of seconds that the switch remains in the quiet state following a failed authentication exchange (for example, the client provided an invalid password), use the **dot1x quiet-period** command in global configuration mode. To return to the default setting, use the **no** form of this command.

dot1x timeout quiet-period *seconds*

no dot1x timeout quiet-period

Syntax Description

seconds	Time in seconds that the switch remains in the quiet state following a failed authentication exchange with the client. The range is 0 to 65535 seconds.

Defaults

The default time is 60 seconds.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

During the quiet period, the switch does not accept or initiate any authentication requests.

You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

If you want to provide a faster response time to the user, enter a smaller number than the default.

Examples

The following example shows how to set the quiet time on the switch to 30 seconds:

```
Switch(config)# dot1x timeout quiet-period 30
```

You can verify your settings by entering the **show dot1x** privileged EXEC command.

Related Commands

Command	Description
show dot1x	Displays 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

dot1x timeout re-authperiod

To set the number of seconds between reauthentication attempts, use the **dot1x timeout re-authperiod** command in global configuration mode. To return to the default setting, use the **no** form of this command.

dot1x timeout re-authperiod *seconds*

no dot1x timeout re-authperiod

Syntax Description

seconds	Number of seconds between reauthentication attempts. The range is 1 to 4294967295.

Defaults

The default is 3600 seconds.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

The **dot1x timeout re-authperiod** global configuration command affects the behavior of the switch only if you have enabled periodic reauthentication by using the **dot1x re-authentication** global configuration command.

You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients or authentication servers.

Examples

The following example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000 seconds:

```
Switch(config)# dot1x re-authentication
Switch(config)# dot1x timeout re-authperiod 4000
```

You can verify your settings by entering the **show dot1x** privileged EXEC command.

Related Commands

Command	Description
dot1x re-authentication	Enables periodic reauthentication of the client.
show dot1x	Displays 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

dot1x timeout tx-period

To set the number of seconds that the switch waits for a response to an Extensible Authentication Protocol (EAP)-request /identity frame from the client before retransmitting the request, use the **dot1x timeout tx-period** command in global configuration mode. To return to the default setting, use the **no** form of this command.

dot1x timeout tx-period *seconds*

no dot1x timeout tx-period

Syntax Description

seconds	Time in seconds that the switch should wait for a response to an EAP-request/identity frame from the client before retransmitting the request. The range is 1 to 65535 seconds.

Defaults

The default is 30 seconds.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

You should change the default value of this command only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients or authentication servers.

Examples

The following example shows how to set 60 as the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request:

```
Switch(config)# dot1x timeout tx-period 60
```

You can verify your settings by entering the **show dot1x** privileged EXEC command.

Related Commands

Command	Description
dot1x max-req	Sets the maximum number of times that the switch sends an EAP-request/identity frame before restarting the authentication process.
show dot1x	Displays 802.1x statistics, administrative status, and operational status for the switch or for the specified interface.

ip access-group

To control access to an interface, use the **ip access-group** command in interface configuration mode. To remove an access group from an interface, use the **no** form of this command.

ip access-group {*access-list-number* | *name*} **in**

no ip access-group {*access-list-number* | *name*} **in**

Syntax Description

access-list-number	Number of the IP access control list (ACL), from 1 to 199 or from 1300 to 2699.
name	Name of an IP ACL, specified in the **ip access-list** command.
in	Applies the IP ACL to packets entering the interface.

Defaults

No ACL is applied to the interface.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

You can apply IP ACLs only to ingress interfaces.

The ACLs can be standard or extended.

For standard ACLs, after receiving a packet, the switch checks the packet source address. If the source address matches a defined address in the ACL and the list permits the address, the switch forwards the packet.

For extended ACLs, after receiving the packet, the switch checks the match conditions in the ACL. If the conditions are matched, the switch forwards the packet.

If the specified ACL does not exist, the switch forwards all packets.

IP access groups can be separated on Layer 2 and Layer 3 interfaces.

Note For more information about configuring IP ACLs, refer to the “Configuring IP Services” chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Examples

The following example shows how to apply a numbered ACL to an interface:

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# ip access-group 101 in
```

You can verify your settings by entering the **show access-lists** or **show ip access-lists** privileged EXEC command.

Related Commands

Command	Description
deny (access-list configuration)	Configures conditions for an IP ACL.
ip access-list	Defines an IP ACL.
permit (access-list configuration)	Configures conditions for an IP ACL.
show ip access-lists	Displays IP ACLs configured on the switch.
show access-lists	Displays ACLs configured on the switch.

ip access-list

To create an IP access control list (ACL) to be used for matching packets to an ACL whose name or number you specify and to enter access-list configuration mode, use the **ip access-list** command in global configuration mode. To delete an existing IP ACL and return to global configuration mode, use the **no** form of this command.

ip access-list {**standard** | **extended**} {*name* | *access-list-number*}

no ip access-list {**standard** | **extended**} {*name* | *access-list-number*}

Syntax Description

name	Name of an ACL. **Note** The ACL name must begin with an alphabetic character to prevent ambiguity with numbered ACLs. A name also cannot contain a space or quotation mark.
access-list-number	Number of an ACL. For standard IP ACLs, the range is from 1 to 99 and from 1300 to 1999. For extended IP ACLs, the range from 100 to 199 and from 2000 to 2699.

Defaults

No named or numbered IP ACLs are defined.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use this command to specify the name or number of the IP ACL for which you want to create or modify ACL match criteria and to enter access-list configuration mode. In this mode, you must enter the **permit** and **deny** commands to configure the permit and deny access conditions for this list.

The **ip access-list** command and its subcommands are used to define packet classification and marking as part of a globally-named service policy applied on a per-interface basis or as an IP access group applied on a per-interface basis.

Specifying **standard** or **extended** with the **ip access-list** command determines the prompt you get when you enter access-list configuration mode.

Note For more information about configuring IP ACLs, refer to the “Configuring IP Services” chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Examples

The following example shows how to configure a standard ACL named Internetfilter1:

```
Switch(config)# ip access-list standard Internetfilter1
Switch(config-std-nacl)# permit 192.5.34.0  0.0.0.255
Switch(config-std-nacl)# permit 192.5.32.0  0.0.0.255
Switch(config-std-nacl)# exit
```

The following example shows how to configure an extended ACL named Internetfilter2:

```
Switch(config)# ip access-list extended Internetfilter2
Switch(config-ext-nacl)# permit any 128.8.10.0  0.0.0.255 eq 80
Switch(config-ext-nacl)# permit any 128.5.8.0  0.0.0.255 eq 80
Switch(config-ext-nacl)# exit
```

Note In these examples, all other IP access is implicitly denied.

You can verify your settings by entering the **show access-lists** or **show ip access-lists** privileged EXEC command.

Related Commands

Command	Description
deny (access-list configuration)	Configures conditions for an IP ACL.
dot1x re-authenticate	Controls access to an interface.
permit (access-list configuration)	Configures conditions for an IP ACL.
service-policy	Applies a policy map to the input of an interface.
show access-lists	Displays ACLs configured on the switch.
show ip access-lists	Displays IP ACLs configured on the switch.

ip igmp snooping

To globally enable Internet Group Management Protocol (IGMP) snooping, use the **ip igmp snooping** command in global configuration mode. To disable IGMP snooping, use the **no** form of this command.

ip igmp snooping

no ip igmp snooping

Syntax Description

This command has no arguments or keywords.

Defaults

By default, IGMP snooping is globally enabled.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5.2)WC(1)	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

When IGMP snooping is globally enabled, it enables IGMP snooping on all the existing VLAN interfaces. When IGMP snooping is globally disabled, it disables IGMP snooping on all the existing VLAN interfaces.

The configuration is saved in nonvolatile RAM (NVRAM).

Examples

The following example shows how to globally enable IGMP snooping:

```
Switch(config)# ip igmp snooping
```

The following example shows how to globally disable IGMP snooping:

```
Switch(config)# no ip igmp snooping
```

You can verify your settings by entering the **show ip igmp snooping** privileged EXEC command.

Related Commands

Command	Description
ip igmp snooping vlan	Enables IGMP snooping on a VLAN interface.
ip igmp snooping vlan immediate-leave	Enables IGMP Immediate-Leave processing.
ip igmp snooping vlan mrouter	Configures a Layer 2 port as a multicast router port.

Command	Description
ip igmp snooping vlan static	Configures a Layer 2 port as a member of a group.
show ip igmp snooping	Displays the IGMP snooping configuration.

ip igmp snooping vlan

To enable Internet Group Management Protocol (IGMP) snooping on a specific VLAN, use the **ip igmp snooping vlan** command in global configuration mode. To disable IGMP snooping on a VLAN interface, use the **no** form of this command.

ip igmp snooping vlan *vlan-id*

no ip igmp snooping vlan *vlan-id*

Syntax Description

vlan-id	VLAN ID value. The range is from 1 to 1001. Do not enter leading zeroes.

Defaults

By default, IGMP snooping is enabled when each VLAN is created.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5.2)WC(1)	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

This command automatically configures the VLAN if it is not already configured. The configuration is saved in nonvolatile RAM (NVRAM).

Examples

The following example shows how to enable IGMP snooping on VLAN 2:

```
Switch(config)# ip igmp snooping vlan 2
```

The following example shows how to disable IGMP snooping on VLAN 2:

```
Switch(config)# no ip igmp snooping vlan 2
```

You can verify your settings by entering the **show ip igmp snooping vlan** privileged EXEC command.

Related Commands

Command	Description
ip igmp snooping	Globally enables IGMP snooping. IGMP snooping must be globally enabled in order to be enabled on a VLAN.
ip igmp snooping vlan immediate-leave	Enables IGMP Immediate-Leave processing.
ip igmp snooping vlan mrouter	Configures a Layer 2 port as a multicast router port.
ip igmp snooping vlan static	Configures a Layer 2 port as a member of a group.
show ip igmp snooping	Displays the IGMP snooping configuration.

ip igmp snooping vlan immediate-leave

To enable Internet Group Management Protocol (IGMP) Immediate-Leave processing on a VLAN interface, use the **ip igmp snooping immediate-leave** command in global configuration mode. To disable Immediate-Leave processing on the VLAN interface, use the **no** form of this command.

ip igmp snooping vlan *vlan-id* **immediate-leave**

no ip igmp snooping vlan *vlan-id* **immediate-leave**

Syntax Description

vlan-id	VLAN ID value. The range is between 1 to 1001. Do not enter leading zeroes.

Defaults

By default, IGMP Immediate-Leave processing is disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5.2)WC(1)	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use the Immediate-Leave feature only when there is only one IP multicast receiver present on every port in the VLAN. The Immediate-Leave configuration is saved in nonvolatile RAM (NVRAM).

The Immediate-Leave feature is supported only with IGMP version 2 hosts.

Examples

The following example shows how to enable IGMP Immediate-Leave processing on VLAN 1:

```
Switch(config)# ip igmp snooping vlan 1 immediate-leave
```

The following example shows how to disable IGMP Immediate-Leave processing on VLAN 1:

```
Switch(config)# no ip igmp snooping vlan 1 immediate-leave
```

You can verify your settings by entering the **show ip igmp snooping vlan** privileged EXEC command.

Related Commands

Command	Description
ip igmp snooping	Enables IGMP snooping.
ip igmp snooping vlan mrouter	Configures a Layer 2 port as a multicast router port.
ip igmp snooping vlan static	Configures a Layer 2 port as a member of a group.

Command	Description
show ip igmp snooping	Displays the IGMP snooping configuration.
show mac-address-table multicast	Displays the Layer 2 multicast entries for a VLAN.

ip igmp snooping vlan mrouter

To add a multicast router port and to configure the multicast router learning method, use the **ip igmp snooping vlan mrouter** command in global configuration mode. To remove the configuration, use the **no** form of this command.

ip igmp snooping vlan *vlan-id* **mrouter** {**interface** *interface-id* | **learn pim-dvmrp**}

no ip igmp snooping vlan *vlan-id* **mrouter** {**interface** *interface-id* | **learn pim-dvmrp**}

Syntax Description

vlan-id	Specify the VLAN ID. The range is from 1 to 1001. Do not enter leading zeroes.
interface-id	Specify the interface of the member port that is configured to a static router port.
learn pim-dvmrp	Specify the multicast router snooping PIM-DVMRP packets multicast router learning method.

Defaults

The default learning method is **pim-dvmrp**.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5.2)WC(1)	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

The CGMP learning method is useful for controlling traffic in Cisco router environments.

The configured learning method is saved in nonvolatile RAM (NVRAM).

Static connections to multicast routers are supported only on switch ports.

Examples

The following example shows how to configure Fast Ethernet interface 0/6 as a multicast router port:

```
Switch(config)# ip igmp snooping vlan 1 mrouter interface fastethernet0/6
```

You can verify your settings by entering the **show ip igmp snooping mrouter** privileged EXEC command.

Related Commands

Command	Description
ip igmp snooping	Globally enables IGMP snooping.
ip igmp snooping vlan	Enables IGMP snooping on the VLAN interface.

Command	Description
ip igmp snooping vlan immediate-leave	Configures IGMP Immediate-Leave processing.
ip igmp snooping vlan static	Configures a Layer 2 port as a member of a group.
show ip igmp snooping mrouter	Displays the statically and dynamically learned multicast router ports.

ip igmp snooping vlan static

To add a Layer 2 port as a member of a multicast group, use the **ip igmp snooping vlan** *vlan-id* **static** command in global configuration mode. To remove the configuration, use the **no** form of this command.

ip igmp snooping vlan *vlan-id* **static** *mac-address* **interface** *interface-id*

no ip igmp snooping vlan *vlan-id* **static** *mac-address* **interface** *interface-id*

Syntax Description

vlan *vlan-id*	Specifies the VLAN ID. The range is 1 to 1001. Do not enter leading zeroes.
static *mac-address*	Specifies the static group MAC address.
interface *interface-id*	Specifies the interface configured to a static router port.

Defaults

No Layer 2 ports are configured.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5.2)WC(1)	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

The command is used to statically configure the IP multicast group member ports.

The static ports and groups are saved in nonvolatile RAM (NVRAM).

Static connections to multicast routers are supported only on switch ports.

Examples

The following example shows how to statically configure a host on an interface:

```
Switch(config)# ip igmp snooping vlan 1 static 0100.5e02.0203 interface fastethernet0/6
Configuring port FastEthernet 0/6 on group 0100.5e02.0203
```

You can verify your settings by entering the **show mac-address-table multicast** privileged EXEC command.

Related Commands

Command	Description
ip igmp snooping	Enables IGMP snooping.
ip igmp snooping vlan	Enables IGMP snooping on the VLAN interface.
ip igmp snooping vlan immediate-leave	Configures IGMP Immediate-Leave processing.

Command	Description
ip igmp snooping vlan mrouter	Configures a Layer 2 port as a multicast router port.
show mac-address-table multicast	Displays the Layer 2 multicast entries for a VLAN.

match (class-map configuration)

To define the match criteria to classify traffic, use the **match** command in class-map configuration mode. To remove the match criteria, use the **no** form of this command.

match {**access-group** *acl-index-or-name*}

no match {**access-group** *acl-index-or-name*}

Syntax Description

access-group *acl-index-or-name*	Number or name of an IP standard or extended access control list (ACL). For an IP standard ACL, the ACL index range is 1 to 99 and 1300 to 1999. For an IP extended ACL, the ACL index range is 100 to 199 and 2000 to 2699.

Defaults

No match criteria are defined.

Command Modes

Class-map configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

The **match** command is used to specify which fields in the incoming packets are examined to classify the packets. Only IP access groups are supported.

Only one **match** command per class map is supported.

Note For more information about configuring IP ACLs, refer to the “Configuring IP Services” chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Examples

The following example shows how to classify traffic on an interface by using the access group named acl2:

```
Switch(config)# class-map class2
Switch(config-cmap)# match access-group acl2
Switch(config-cmap)# exit
```

You can verify your settings by entering the **show class-map** privileged EXEC command.

Related Commands

Command	Description
class	Defines a traffic classification for a policy to act on using the class-map name or access group.
class-map	Creates a class map to be used for matching packets to the class whose name you specify.
dot1x re-authenticate	Controls access to an interface.
show class-map	Displays QoS class maps.
show policy-map	Displays QoS policy maps.

mls qos cos

To define the default class of service (CoS) value of a port or to assign the default CoS to all incoming packets on the port, use the **mls qos cos** command in interface configuration mode. To return to the default setting, use the **no** form of this command.

mls qos cos {*default-cos* | **override**}

no mls qos cos {*default-cos* | **override**}

Syntax Description

default-cos	Assigns a default CoS value to a port. If the port is CoS trusted and packets are untagged, the default CoS value becomes a CoS value used to select one output queue to index into the CoS-to-Differentiated Services Code Point (DSCP) map. The CoS range is 0 to 7.
override	Overrides the CoS of the incoming packets, and applies the default CoS value on the port to all incoming packets.

Defaults

The default CoS value for a port is 0.

CoS override is disabled.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced. It replaced the **switchport priority** command.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

You can use the default value to assign a CoS and DSCP value to all packets entering a port if the port has been configured by using the **override** keyword.

Use the **override** keyword when all incoming packets on certain ports deserve higher or lower priority than packets entering from other ports. Even if a port was previously set to trust DSCP or CoS, this command overrides that trust state, and all the incoming CoS values are assigned the default CoS value configured with the **mls qos cos** command. If an incoming packet is tagged, the CoS value of the packet is modified with the default CoS of the port at the ingress port.

Examples

The following example shows how to configure the default port CoS to 4:

```
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# mls qos trust cos
Switch(config-if)# mls qos cos 4
```

The following example shows how to assign all the packets entering a port to the default port CoS value of 4:

```
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# mls qos cos 4
Switch(config-if)# mls qos cos override
```

You can verify your settings by entering the **show mls qos interface** privileged EXEC command.

Related Commands

Command	Description
mls qos map	Defines the CoS-to-DSCP map or the DSCP-to-CoS map.
mls qos trust	Configures the port trust state.
show interface fax/y switchport	Displays switchport interfaces.
show mls qos interface	Displays QoS information.

mls qos map

To define the class of service (CoS)-to-Differentiated Services Code Point (DSCP) map or DSCP-to-CoS map, use the **mls qos map** command in global configuration mode. To return to the default map, use the **no** form of this command.

mls qos map {**cos-dscp** *dscp1...dscp8* | **dscp-cos** *dscp-list* **to** *cos*}

no mls qos map {**cos-dscp** | **dscp-cos**}

Syntax Description

cos-dscp *dscp1...dscp8*	Defines the CoS-to-DSCP map. For *dscp1...dscp8*, enter eight DSCP values that correspond to CoS values 0 to 7. Separate each DSCP value with a space. The supported DSCP values are 0, 8, 10, 16, 18, 24, 26, 32, 34, 40, 46, 48, and 56.
dscp-cos *dscp-list* **to** *cos*	Defines the DSCP-to-CoS map. For *dscp-list*, enter up to 13 DSCP values separated by spaces. Then enter the **to** keyword. The supported DSCP values are 0, 8, 10, 16, 18, 24, 26, 32, 34, 40, 46, 48, and 56. For *cos*, enter the CoS value to which the DSCP values correspond. The CoS range is 0 to 7.

Defaults

Table 18 shows the default CoS-to-DSCP map:

Table 18 Default CoS-to-DSCP Map

CoS Value	0	1	2	3	4	5	6	7
DSCP Value	0	8	16	26	32	46	48	56

Table 19 shows the default DSCP-to-CoS map:

Table 19 Default DSCP-to-CoS Map

DSCP Values	0	8, 10	16, 18	24, 26	32, 34	40, 46	48	56
CoS Value	0	1	2	3	4	5	6	7

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

All the maps are globally defined. You apply all maps to all ports.

If you enter the **mls qos trust cos** command, the default CoS-to-DSCP map is applied.

If you enter the **mls qos trust dscp** command, the default DSCP-to-CoS map is applied.

After a default map is applied, you can define the CoS-to-DSCP or DSCP-to-CoS map by entering consecutive **mls qos map** commands.

The supported DSCP values are 0, 8, 10, 16, 18, 24, 26, 32, 34, 40, 46, 48, and 56. If the **mls qos trust dscp** command is entered and a packet with an untrusted DSCP value is at an ingress port, the packet CoS value is set to 0.

Examples

The following example shows how to define the DSCP-to-CoS map. DSCP values 16, 18, 24, and 26 are mapped to CoS 1. DSCP values 0, 8, and 10 are mapped to CoS 0:

```
Switch# configure terminal
Switch(config)# mls qos map dscp-cos 16 18 24 26 to 1
Switch(config)# mls qos map dscp-cos 0 8 10 to 0
```

The following example shows how to define the CoS-to-DSCP map. CoS values 0 to 7 are mapped to DSCP values 8, 8, 8, 8, 24, 32, 56, and 56:

```
Switch# configure terminal
Switch(config)# mls qos map cos-dscp 8 8 8 8 24 32 56 56
```

You can verify your settings by entering the **show mls qos maps** privileged EXEC command.

Related Commands

Command	Description
mls qos cos	Defines the default CoS value of a port or assigns the default CoS to all incoming packets on the port.
mls qos trust	Configures the port trust state.
show mls qos maps	Displays QoS mapping information.

mls qos trust

To configure the port trust state and classify traffic by examining the class of service (CoS) or Differentiated Services Code Point (DSCP) value, use the **mls qos trust** command in interface configuration mode. To return a port to its untrusted state, use the **no** form of this command.

mls qos trust [**cos** | **dscp**]

no mls qos trust [**cos** | **dscp**]

Syntax Description

cos	(Optional) Classifies ingress packets with packet CoS values. For untagged packets, use the port default CoS value.
dscp	(Optional) Classifies ingress packets with packet DSCP values (most significant 6 bits of 8-bit service-type field). For non-IP packets, the packet CoS value is 0.

Defaults

The port is not trusted. If no keyword is specified, the default is **dscp**.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Packets entering a quality of service (QoS) domain are classified at the edge of the QoS domain. Because the packets are classified at the edge, the switch port within the QoS domain can be configured to one of the trusted states; there is no need to classify the packets at every switch within the domain. Use this command to specify whether the port is trusted and which fields of the packet to use to classify traffic.

When a port is configured with trust DSCP and the incoming packet is a non-IP packet, the CoS value for the packet is set to 0, and the DSCP-to-CoS map is not applied.

If DSCP is trusted, the DSCP field of the IP packet is not modified. However, it is still possible that the CoS value of the packet is modified (according to the DSCP-to-CoS map).

If CoS is trusted, CoS of the packet is not modified, but DSCP can be modified (according to the CoS-to-DSCP map) if it is an IP packet.

Examples

The following example shows how to configure a port to be a DSCP-trusted port:

```
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# mls qos trust dscp
```

The following example shows how to configure a VLAN interface to be a DSCP-trusted port. DSCP-to-COS mapping occurs for all packets with the configured VLAN ID of 60 egressing from the CPU to the physical port.

```
Switch(config)# interface vlan 60
Switch(config-if)# mls qos trust dscp
```

You can verify your settings by entering the **show mls qos interface** privileged EXEC command.

Related Commands

Command	Description
mls qos cos	Defines the default CoS value of a port or assigns the default CoS to all incoming packets on the port.
mls qos map	Defines the CoS-to-DSCP map or the DSCP-to-CoS map.
show mls qos interface	Displays QoS information.

permit (access-list configuration)

To configure conditions for a named or numbered IP access control list (ACL), use the **permit** command in access-list configuration mode. To remove a permit condition from the IP ACL, use the **no** form of the command.

Use these commands with standard IP ACLs:

permit {*source source-wildcard* | **host** *source* | **any**}

no permit {*source source-wildcard* | **host** *source* | **any**}

Use these commands with extended IP ACLs:

permit *protocol* {*source source-wildcard* | **host** *source* | **any**} [*operator port*] {*destination destination-wildcard* | **host** *source* | **any**} [*operator port*]

no permit *protocol* {*source source-wildcard* | **host** *source* | **any**} [*operator port*] {*destination destination-wildcard* | **host** *source* | **any**} [*operator port*]

Syntax Description

source source-wildcard | **host** *source* | **any**	Defines a source IP address and wildcard. The *source* is the source address of the network or host from which the packet is being sent, specified in one of these ways: • The 32-bit quantity in dotted-decimal format. The *source-wildcard* applies wildcard bits to the source. • The keyword **host**, followed by the 32-bit quantity in dotted-decimal format, as an abbreviation for source and source-wildcard of *source* 0.0.0.0. • The keyword **any** as an abbreviation for *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. You do not need to enter a source-wildcard.
protocol	Name of an IP protocol. *protocol* can be **ip**, **tcp**, or **udp**.
destination destination-wildcard | **host** *source* | **any**	Defines a destination IP address and wildcard. The *destination* is the destination address of the network or host to which the packet is being sent, specified in one of these ways: • The 32-bit quantity in dotted-decimal format. The *destination-wildcard* applies wildcard bits to the destination. • The keyword **host**, followed by the 32-bit quantity in dotted-decimal format, as an abbreviation for source and source-wildcard of *source* 0.0.0.0. • The keyword **any** as an abbreviation for *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. You do not need to enter a destination-wildcard.

operator port	(Optional) Defines a source or destination port. The *operator* can be only **eq** (equal). If *operator* is after the source IP address and wildcard, conditions match when the source port matches the defined port. If *operator* is after the destination IP address and wildcard, conditions match when the destination port matches the defined port. The *port* is a decimal number or name of a Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port. The number can be from 0 to 65535. Use TCP port names only for TCP traffic. Use UDP port names only for UDP traffic.

Defaults

There are no specific conditions that permit packets in a named or numbered IP ACL.

The default ACL is always terminated by an implicit deny statement for all packets.

Command Modes

Access-list configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use this command after the **ip access-list** global configuration command to specify permit conditions for a named or numbered IP ACL. You can specify a source IP address, destination IP address, IP protocol, TCP port, or UDP port. Specify the TCP and UDP port numbers only if *protocol* is **tcp** or **udp** and *operator* is **eq**.

Note For more information about configuring IP ACLs, refer to the “Configuring IP Services” chapter in the *Cisco IOS IP Configuration Guide*, Release 12.2.

Examples

The following example shows how to create an extended IP ACL and configure permit conditions for it:

```
Switch(config)# ip access-list extended Internetfilter2
Switch(config-ext-nacl)# permit host 36.10.10.5 any
Switch(config-ext-nacl)# permit host 192.1.10.8 any
```

The following is an example of a standard ACL that sets permit conditions:

```
ip access-list standard Acclist1
 permit 192.5.34.0  0.0.0.255
 permit 128.88.10.0  0.0.0.255
 permit 36.1.1.0  0.0.0.255
```

Note In these examples, all other IP access is implicitly denied.

You can verify your settings by entering the **show ip access-lists** or **show access-lists** privileged EXEC command.

Related Commands

Command	Description
deny (access-list configuration)	Sets deny conditions for an IP ACL.
dot1x re-authenticate	Controls access to an interface.
ip access-list	Defines an IP ACL.
show access-lists	Displays ACLs configured on a switch.
show ip access-lists	Displays IP ACLs configured on the switch.

police

To define a policer for classified traffic, use the **police** command in policy-map class configuration mode. To remove an existing policer, use the **no** form of this command.

police {*bps* | **cir** *bps*} [*burst-byte* | **bc** *burst-byte*] **conform-action transmit** [**exceed-action** {**drop** | **dscp** *dscp-value*}]

police {*bps* | **cir** *bps*} [*burst-byte* | **bc** *burst-byte*] **conform-action transmit** [**exceed-action** {**drop** | **dscp** *dscp-value*}]

Syntax Description

bps | **cir** *bps*	Average traffic rate or committed information rate in bits per second (bps). For 10/100 ports, the range is 1000000 to 100000000, and the granularity is 1 Mbps. For Gigabit-capable Ethernet ports, the range is 8000000 to 1016000000, and the granularity is 8 Mbps.
burst-byte | **bc** *burst-byte*	(Optional) Normal burst size or burst count in bytes.
conform-action transmit	Sends packets that conform to the rate limit.
exceed-action drop	(Optional) When the specified rate is exceeded, specifies that the switch drop the packet.
exceed-action dscp *dscp-value*	(Optional) When the specified rate is exceeded, specifies that the switch changes the Differentiated Services Code Point (DSCP) of the packet to the specified *dscp-value* and then sends the packet.

Defaults

No policers are defined.

Command Modes

Policy-map class configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

You can configure up to six policers on ingress Fast Ethernet ports.

You can configure up to 60 policers on ingress Gigabit-capable Ethernet ports.

Policers cannot be configured on egress Fast Ethernet and Gigabit-capable Ethernet ports.

To return to policy-map configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.

Note For more information about configuring access control lists (ACLs), refer to the “Configuring Network Security with ACLs” chapter in the *Catalyst 2950 Desktop Switch Software Configuration Guide* for this release.

Examples

The following example shows how to configure a policer that sets the DSCP value to 46 if traffic does not exceed a 1-Mbps average rate with a burst size of 65536 bytes and drops packets if traffic exceeds these conditions:

```
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# set ip dscp 46
Switch(config-pmap-c)# police 1000000 65536 exceed-action drop
Switch(config-pmap-c)# exit
```

You can verify your settings by entering the **show policy-map** privileged EXEC command.

Related Commands

Command	Description
policy-map	Creates or modifies a policy map that can be attached to multiple interfaces, and enters policy-map configuration mode.
show policy-map	Displays QoS policy maps.

policy-map

To create or modify a policy map that can be attached to multiple interfaces and to enter policy-map configuration mode, use the **policy-map** command in global configuration mode. To delete an existing policy map and return to global configuration mode, use the **no** form of this command.

policy-map *policy-map-name*

no policy-map *policy-map-name*

Syntax Description

policy-map-name	Name of the policy map.

Defaults

No policy maps are defined.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Entering the **policy-map** command enables the policy-map configuration mode. These configuration commands are available:

- **class**: defines the classification match criteria for the specified class map. For more information, see the **class** command.
- **description**: describes the policy map (up to 200 characters).
- **exit**: exits policy-map configuration mode and returns to global configuration mode.
- **no**: removes a previously defined policy map.
- **rename**: renames the policy map.

Note In a policy map, the class named class-default is not supported. The switch does not filter traffic based on the policy map defined by the **class class-default** policy-map configuration command.

To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command.

Before you can configure policies for classes whose match criteria are defined in a class map, use the **policy-map** command to specify the name of the policy map to be created or modified. Entering this command also enables the policy-map configuration mode in which you can configure or modify the class policies for that policy map.

You can configure class policies in a policy map only if the classes have match criteria defined for them. Use the **class-map** and **match** commands to configure the match criteria for a class. Only one **match** command per class map is supported.

Only one policy map per interface per direction is supported. You can apply the same policy map to multiple interfaces but only in the ingress direction.

Note For more information about configuring access control lists (ACLs), refer to the "Configuring Network Security with ACLs" chapter in the *Catalyst 2950 Desktop Switch Software Configuration Guide* for this release.

Examples

The following example shows how to create a policy map called policy1. When attached to the ingress direction, it matches all the incoming traffic defined in class1 and polices the traffic at an average rate of 1 Mbps and bursts at 65536 bytes. Traffic exceeding the profile is dropped:

```
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# police 1000000 65536 exceed-action drop
Switch(config-pmap-c)# exit
Switch(config-pmap)#
```

The following example shows how to delete policymap2:

```
Switch(config)# no policy-map policymap2
```

You can verify your settings by entering the **show policy-map** privileged EXEC command.

Related Commands

Command	Description
class	Defines the default CoS value of a port or assigns the default CoS to all incoming packets on the port.
class-map	Creates a class map to be used for matching packets to the class whose name you specify.
police	Defines a policer for classified traffic.
show policy-map	Displays QoS policy maps.

service-policy

To apply a policy map defined by the **policy-map** command to the input of a particular interface, use the **service-policy** command in interface configuration mode. To remove the policy map and interface association, use the **no** form of this command.

service-policy input *policy-map-name*

no service-policy input *policy-map-name*

Syntax Description

input *policy-map-name*	Applies the specified policy-map to the input of an interface.

Defaults

No policy maps are attached to the interface.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Only one policy map per ingress interface is supported.

Service policy maps cannot be defined on egress interfaces.

Note For more information about configuring access control lists (ACLs), refer to the “Configuring Network Security with ACLs” chapter in the *Catalyst 2950 Desktop Switch Software Configuration Guide* for this release.

Examples

The following example shows how to apply plcmap1 to an ingress interface:

```
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy input plcmap1
```

You can verify your settings by entering the **show policy-map** privileged EXEC command.

Related Commands

Command	Description
policy-map	Creates or modifies a policy map that can be attached to multiple interfaces to specify a service policy.
show policy-map	Displays QoS policy maps.

show access-lists

To display access control lists (ACLs) configured on the switch, use the **show access-lists** command in privileged EXEC mode.

show access-lists [*name* | *number*]

Syntax Description

name	(Optional) Name of the ACL.
number	(Optional) ACL number. The range is from 1 to 2699.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Expressions are case sensitive. For example, if you enter **| exclude output**, the lines that contain *output* do not appear, but the lines that contain *Output* appear.

Examples

The following is sample output from the **show access-lists** command:

```
Switch# show access-lists
Standard IP access list testingacl
    permit 10.10.10.2
Standard IP access list wizard_1-1-1-2
    permit 1.1.1.2
Extended IP access list 103
    permit tcp any any eq www
Extended IP access list CMP-NAT-ACL
    Dynamic Cluster-HSRP deny   ip any any
    Dynamic Cluster-NAT permit ip any any
      permit ip host 10.123.222.192 any
      permit ip host 10.228.215.0 any
      permit ip host 10.245.137.0 any
      permit ip host 10.245.155.128 any
      permit ip host 10.221.111.64 any
      permit ip host 10.216.25.128 any
      permit ip host 10.186.122.64 any
      permit ip host 10.169.110.128 any
      permit ip host 10.146.106.192 any
```

Related Commands

Command	Description
ip access-list	Configures an IP ACL on the switch.
show ip access-lists	Displays the IP ACLs configured on a switch.

show class-map

To display quality of service (QoS) class maps, which define the match criteria to classify traffic, use the **show class-map** command in privileged EXEC mode.

show class-map [*class-map-name*]

Syntax Description

class-map-name	(Optional) Display the contents of the specified class map.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

If you do not specify a *class-map-name*, all class maps appear.

Expressions are case sensitive. For example, if you enter **| exclude output**, the lines that contain *output* do not appear, but the lines that contain *Output* appear.

Examples

The following is sample output from the **show class-map test** command:

```
Switch# show class-map test
 Class Map match-all test (id 2)
   Match access-group name testingacl
```

The following is sample output from the **show class-map** command:

```
Switch# show class-map
 Class Map match-all wizard_1-1-1-2 (id 3)
   Match access-group name videowizard_1-1-1-2

 Class Map match-all test (id 2)
   Match access-group name testingacl

 Class Map match-any class-default (id 0)
   Match any

 Class Map match-all class1 (id 5)
   Match access-group  103

 Class Map match-all classtest (id 4)
  Description: This is a test.
   Match access-group name testingacl
```

Related Commands

Command	Description
class-map	Creates a class map to be used for matching packets to the class whose name you specify.
match (class-map configuration)	Defines the match criteria to classify traffic.

show dot1x

To display the 802.1x statistics, administrative status, and operational status for the switch or for the specified interface, use the **show dot1x** command in privileged EXEC mode.

show dot1x [**statistics**] [**interface** *interface-id*]

Syntax Description

statistics	(Optional) Displays 802.1x statistics.
interface *interface-id*	(Optional) Slot and port number of the interface to reauthenticate.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

If you do not specify an interface, global parameters and a summary appear. If you specify an interface, details for that interface appear.

If you specify an interface with the **statistics** keyword, statistics appear for all physical ports.

Examples

The following is sample output from the **show dot1x** command:

```
Switch# show dot1x

Global 802.1X Parameters
    reauth-enabled                no
    reauth-period               3600
    quiet-period                  60
    tx-period                     30
    supp-timeout                  30
    server-timeout                30
    reauth-max                     2
    max-req                        2

802.1X Port Summary
    Port Name                 Status      Mode                 Authorized
    Gi0/1                     disabled    n/a                  n/a
    Gi0/2                     enabled     Auto (negotiate)     no

    802.1X Port Details
    802.1X is disabled on GigabitEthernet0/1
802.1X is enabled on GigabitEthernet0/2
      Status                 Unauthorized
      Port-control           Auto
      Supplicant             0060.b0f8.fbfb
      Multiple Hosts         Disallowed
      Current Identifier     2
      Authenticator State Machine
        State               AUTHENTICATING
        Reauth Count        1

      Backend State Machine
        State               RESPONSE
        Request Count       0
        Identifier (Server) 2

      Reauthentication State Machine
        State               INITIALIZE
```

Note In the previous example, the supp-timeout, server-timeout, and reauth-max values in the Global 802.1x Parameters section are not configurable.When relaying a request from the Remote Authentication Dial-In User Service (RADIUS) authentication server to the client, the supp-timeout is the amount of time the switch waits for a response before it resends the request. When relaying a response from the client to the RADIUS authentication server, the server-timeout is the amount of time the switch waits for a reply before it resends the response. The reauth-max parameter is the maximum number of times that the switch tries to authenticate the client without receiving any response before the switch resets the port and restarts the authentication process.

In the 802.1x Port Summary section of the example, the Status column shows whether the port is enabled for 802.1x (the **dot1x port-control** interface configuration command is set to **auto** or **force-unauthorized**). The Mode column shows the operational status of the port; for example, if you configure the **dot1x port-control** interface configuration command to **force-unauthorized**, but the port has not changed to that state, the Mode column displays *auto*. If you disable 802.1x, the Mode column displays *n/a*.

The Authorized column shows the authorization state of the port. For information about port states, refer to the "Configuring 802.1x Port-Based Authentication" chapter in the *Catalyst 2950 Desktop Switch Software Configuration Guide*.

The following is sample output from the **show dot1x interface gigabitethernet0/2** privileged EXEC command. Table 20 describes the fields in the output.

```
Switch# show dot1x interface gigabitethernet0/2

802.1X is enabled on GigabitEthernet0/2
  Status                Authorized
  Port-control          Auto
  Supplicant            0060.b0f8.fbfb
  Multiple Hosts        Disallowed
  Current Identifier    3

  Authenticator State Machine
    State               AUTHENTICATED
    Reauth Count        0

  Backend State Machine
    State               IDLE
    Request Count       0
    Identifier (Server) 2

  Reauthentication State Machine
    State               INITIALIZE
```

Table 20 show dot1x interface Field Descriptions

Field	Description
Status	Status of the port (authorized or unauthorized). The status of a port appears as authorized if the **dot1x port-control** interface configuration command is set to **auto**, and authentication was successful.
Port-control	Setting of the **dot1x port-control** interface configuration command.
Supplicant	Ethernet MAC address of the client, if one exists. If the switch has not discovered the client, this field displays *Not set*.
Multiple Hosts	Setting of the **dot1x multiple-hosts** interface configuration command (allowed or disallowed).
Current Identifier[1]	Each exchange between the switch and the client includes an identifier, which matches requests with responses. This number is incremented with each exchange and can be reset by the authentication server.

[1] This field and the remaining fields in the output show internal state information. For a detailed description of these state machines and their settings, refer to the IEEE 802.1x standard.

The following is sample output from the **show dot1x statistics interface gigiabitethernet0/1** command. Table 21 describes the fields in the example.

```
Switch# show dot1x statistics interface gigabitethernet0/1

GigabitEthernet0/1

    Rx: EAPOL     EAPOL     EAPOL     EAPOL     EAP       EAP       EAP
        Start     Logoff    Invalid   Total     Resp/Id   Resp/Oth  LenError
        0         0         0         21        0         0         0

        Last      Last
        EAPOLVer  EAPOLSrc
        1         0002.4b29.2a03

    Tx: EAPOL     EAP       EAP
        Total     Req/Id    Req/Oth
        622       445       0
```

Table 21 show dot1x statistics Field Descriptions

Field	Description
RX EAPOL[1] Start	Number of valid EAPOL-start frames that have been received.
RX EAPOL Logoff	Number of EAPOL-logoff frames that have been received.
RX EAPOL Invalid	Number of EAPOL frames that have been received and have an unrecognized frame type.
RX EAPOL Total	Number of valid EAPOL frames of any type that have been received.
RX EAP[2] Resp/ID	Number of EAP-response/identity frames that have been received.

Table 21 show dot1x statistics Field Descriptions (continued)

Field	Description
RX EAP Resp/Oth	Number of valid EAP-response frames (other than response/identity frames) that have been received.
RX EAP LenError	Number of EAPOL frames that have been received in which the packet body length field is invalid.
Last EAPOLVer	Protocol version number carried in the most recently received EAPOL frame.
LAST EAPOLSrc	Source MAC address carried in the most recently received EAPOL frame.
TX EAPOL Total	Number of EAPOL frames of any type that have been sent.
TX EAP Req/Id	Number of EAP-request/identity frames that have been sent.
TX EAP Req/Oth	Number of EAP-request frames (other than request/identity frames) that have been sent.

[1] EAPOL = Extensible Authentication Protocol over LAN

[2] EAP = Extensible Authentication Protocol

Related Commands

Command	Description
dot1x default	Resets the global 802.1x parameters to their default values.

show ip access-lists

To display IP access control lists (ACLs) configured on the switch, use the **show ip access-lists** command in privileged EXEC mode.

show ip access-lists [*name* | *number*]

Syntax Description

name	(Optional) ACL name.
number	(Optional) ACL number. The range is from 1 to 199 and from 1300 to 2699.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Expressions are case sensitive. For example, if you enter **| exclude output**, the lines that contain *output* do not appear, but the lines that contain *Output* appear.

Examples

The following is sample output from the **show ip access-lists** command:

```
Switch# show ip access-lists
Standard IP access list testingacl
    permit 10.10.10.2
Standard IP access list wizard_1-1-1-2
    permit 1.1.1.2
Extended IP access list 103
    permit tcp any any eq www
Extended IP access list CMP-NAT-ACL
    Dynamic Cluster-HSRP deny   ip any any
    Dynamic Cluster-NAT permit ip any any
      permit ip host 10.245.155.128 any
      permit ip host 10.245.137.0 any
      permit ip host 10.146.106.192 any
      permit ip host 10.216.25.128 any
      permit ip host 10.228.215.0 any
      permit ip host 10.221.111.64 any
      permit ip host 10.123.222.192 any
      permit ip host 10.169.110.128 any
      permit ip host 10.186.122.64 any
```

The following is sample output from the **show ip access-lists 103** command:

```
Switch# show ip access-lists 103
Extended IP access list 103
    permit tcp any any eq www
```

Related Commands

Command	Description
access-list (IP extended)	Configures an extended ACL on the switch.
access-list (IP standard)	Configures a standard ACL on the switch.
ip access-list	Configures an IP ACL on the switch.
show access-lists	Displays ACLs configured on a switch.

show ip igmp snooping

To display the Internet Group Management Protocol (IGMP) snooping configuration of the switch or the VLAN, use the **show ip igmp snooping** command in privileged EXEC mode.

show ip igmp snooping [**vlan** *vlan-id*]

Syntax Description

vlan *vlan-id*	(Optional) Specifies a VLAN. Valid values are 1 to 1001.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5.2)WC(1)	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use this command to display snooping characteristics for the switch or for a specific VLAN.

Examples

The following is sample output from the **show ip igmp snooping** command:

```
Switch# show ip igmp snooping

vlan 1
----------
  IGMP snooping is globally enabled
  IGMP snooping is enabled on this Vlan
  IGMP snooping immediate-leave is enabled on this Vlan
  IGMP snooping mrouter learn mode is pim-dvmrp on this Vlan
vlan 2
----------
  IGMP snooping is globally enabled
  IGMP snooping is enabled on this Vlan
  IGMP snooping immediate-leave is enabled on this Vlan
  IGMP snooping mrouter learn mode is pim-dvmrp on this Vlan
vlan 3
----------
  IGMP snooping is globally enabled
  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
vlan 4
----------
  IGMP snooping is globally enabled
  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
vlan 5
----------
  IGMP snooping is globally enabled
  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
vlan 33
----------
  IGMP snooping is globally enabled
  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
```

The following is sample output from the **show ip igmp snooping vlan 1** command:

```
Switch# show ip igmp snooping vlan 1

vlan 1
----------
  IGMP snooping is globally enabled
  IGMP snooping is enabled on this Vlan
  IGMP snooping immediate-leave is enabled on this Vlan
  IGMP snooping mrouter learn mode is pim-dvmrp on this Vlan
```

Related Commands

Command	Description
ip igmp snooping	Enables IGMP snooping.
ip igmp snooping vlan	Enables IGMP snooping on the VLAN interface.
ip igmp snooping vlan immediate-leave	Configures IGMP Immediate-Leave processing.
ip igmp snooping vlan mrouter	Configures a Layer 2 port as a multicast router port.
show mac-address-table multicast	Displays the Layer 2 multicast entries for a VLAN.

show ip igmp snooping mrouter

To display information on dynamically learned and manually configured multicast router ports, use the **show ip igmp snooping mrouter** command in privileged EXEC mode.

show ip igmp snooping mrouter [**vlan** *vlan-id*]

Syntax Description

vlan *vlan-id*	(Optional) Specifies a VLAN. Valid values are 1 to 1001.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5.2)WC(1)	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

You can also use the **show mac-address-table multicast** command to display entries in the MAC address table for a VLAN that has Internet Group Management Protocol (IGMP) snooping enabled.

Examples

The following is sample output from the **show ip igmp snooping mrouter vlan 1** command:

Note In this example, Fa0/3 is a dynamically learned router port, and Fa0/2 is a configured static router port.

```
Switch# show ip igmp snooping mrouter vlan 1

Vlan    ports
----    -----
   1    Fa0/2(static), Fa0/3(dynamic)
```

Related Commands

Command	Description
ip igmp snooping	Enables IGMP snooping.
ip igmp snooping vlan	Enables IGMP snooping on the VLAN interface.
ip igmp snooping vlan immediate-leave	Configures IGMP Immediate-Leave processing.
ip igmp snooping vlan mrouter	Configures a Layer 2 port as a multicast router port.
show mac-address-table multicast	Displays the Layer 2 multicast entries for a VLAN.

show mls masks

To display the details of the Access Control Parameters (ACPs) used for quality of service (QoS) and security access control lists (ACLs), use the **show mls masks** command in privileged EXEC mode.

show mls masks [**qos** | **security**]

Syntax Description

qos	(Optional) Displays ACPs used for QoS ACLs.
security	(Optional) Displays ACPs used for security ACLs.

Note ACPs are called masks in the command-line interface (CLI) commands and output.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use the **show mls mask** command without keywords to display all ACPs configured on the switch.

Use this command with the **qos** keyword to display the ACPs used for QoS ACLs.

Use this command with the **security** keyword to display the ACPs used for security ACLs.

Note You can configure up to four ACPs (QoS and security) on a switch.

Examples

The following is sample output from the **show mls masks** command:

```
Switch# show mls masks

Mask1
        Type : qos
        Fields : ip-sa(0.0.0.255), ip-da(host), dest-port
        Policymap: pmap1
            Interfaces: Fa0/9, Gi0/1
        Policymap: pmap2
            Interfaces: Fa0/1, Fa0/5, Fa0/13
```

In this example, Mask 1 is a QoS ACP consisting an IP source address (with wildcard bits 0.0.0.255), an IP destination address, and Layer 4 destination port fields. This ACP is used by the QoS policy maps pmap1 and pmap2.

Related Commands

Command	Description
ip access-group	Applies an IP ACL to an interface.
policy-map	Creates or modifies a policy map that can be attached to multiple interfaces and enters policy-map configuration mode.

show mls qos interface

To display quality of service (QoS) information at the interface level, use the **show mls qos interface** command in privileged EXEC mode.

show mls qos interface [*interface-id*] [**policers**]

Syntax Description

interface-id	(Optional) Displays QoS information for the specified interface.
policers	(Optional) Displays all the policers configured on the interface, their settings, and the number of policers unassigned.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use the **show mls qos interface** command without keywords to display parameters for all interfaces.

Use the **show mls qos interface** *interface-id* command to display the parameters for a specific interface.

Examples

The following is sample output from the **show mls qos interface fastethernet0/1** command:

```
Switch# show mls qos interface fastethernet0/1
FastEthernet0/1
trust state: trust cos
COS override: dis
default COS: 0
```

Related Commands

Command	Description
mls qos cos	Defines the default CoS value of a port or assigns the default CoS to all incoming packets on the port.
mls qos map	Defines the CoS-to-DSCP map and DSCP-to-CoS map.
mls qos trust	Configures the port trust state. Ingress traffic can be trusted and classification is performed by examining the CoS or DSCP value.

show mls qos maps

To display quality of service (QoS) mapping information, use the **show mls qos maps** command in privileged EXEC mode.

show mls qos maps [**cos-dscp** | **dscp-cos**]

Syntax Description

cos-dscp	(Optional) Displays the class of service (CoS)-to-DSCP map.
dscp-cos	(Optional) Displays the DSCP-to-CoS map.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Maps are used to generate an internal Differentiated Services Code Point (DSCP) value, which represents the priority of the traffic. Use the **show mls qos maps** command without keywords to display all maps.

Use this command with the **cos-dscp** keyword to display the CoS-to-DSCP map.

Use this command with the **dscp-cos** keyword to display the DSCP-to-CoS map.

Examples

The following is sample output from the **show mls qos maps cos-dscp** command:

```
Switch# show mls qos maps cos-dscp

Cos-dscp map:
        cos:  0  1  2  3  4  5  6  7
     --------------------------------
       dscp:  8  8  8  8 24 32 56 56
```

The following is sample output from the **show mls qos maps dscp-cos** command:

```
Switch# show mls qos maps dscp-cos

Dscp-cos map:
       dscp:  0  8 10 16 18 24 26 32 34 40 46 48 56
     -----------------------------------------------
        cos:  0  1  1  1  2  2  3  3  4  4  5  6  7
```

The following is sample output from the **show mls qos maps** command:

```
Switch# show mls qos maps

Dscp-cos map:
       dscp:  0  8 10 16 18 24 26 32 34 40 46 48 56
     -----------------------------------------------
        cos:  0  1  1  2  2  3  7  4  4  5  5  7  7

   Cos-dscp map:
        cos:  0  1  2  3  4  5  6  7
     --------------------------------
       dscp:  0  8 16 24 32 40 48 56
```

Related Commands

Command	Description
mls qos map	Defines the CoS-to-DSCP map and DSCP-to-CoS map.

show policy-map

To display quality of service (QoS) policy maps, which define classification criteria for incoming traffic, use the **show policy-map** command in privileged EXEC mode. Policy maps can include policers that specify the bandwidth limitations and the action to take if the limits are exceeded.

show policy-map [*policy-map-name* [**class** *class-name*]]

Syntax Description

policy-map-name	(Optional) Displays the specified policy-map name.
class *class-name*	(Optional) Displays QoS policy actions for a individual class.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the following platforms: Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use the **show policy-map** command without keywords to display all policy maps configured on the switch.

Note In a policy map, the class named class-default is not supported. The switch does not filter traffic based on the policy map defined by the **class class-default** policy-map configuration command.

Examples

The following is sample output from the **show policy-map** command:

```
Switch# show policy-map
 Policy Map wand
    Description: this is a description.

 Policy Map wizard_policy3
  class wizard_1-1-1-2

 Policy Map test

 Policy Map policytest
  class  classtest
police 10000000 8192 exceed-action drop
```

The following is sample output from the **show policy-map policytest** command:

```
Switch# show policy-map policytest
 Policy Map policytest
  class  classtest
police 10000000 8192 exceed-action drop
```

The following is sample output from the **show policy-map policytest class classtest** command:

```
Switch# show policy-map policytest class classtest
police 10000000 8192 exceed-action drop
```

Related Commands

Command	Description
policy-map	Creates or modifies a policy map that can be attached to multiple interfaces to specify a service policy.

show spanning-tree

To display spanning-tree information for the specified spanning-tree instances, use the **show spanning-tree** command in privileged EXEC mode.

show spanning-tree [*bridge-group*] [**active** | **backbonefast** | **blockedports** | **bridge** | **brief** | **inconsistentports** | **interface** *interface-id* | **pathcost method** | **root** | **summary** [**totals**] | **uplinkfast** | **vlan** *vlan-id*]

Syntax Description

bridge-group	(Optional) Specifies the bridge group number. The range is 1 to 255.
active	(Optional) Displays spanning-tree information on active interfaces only.
backbonefast	(Optional) Displays spanning-tree BackboneFast status.
blockedports	(Optional) Displays blocked port information.
bridge	(Optional) Displays status and configuration of this switch.
brief	(Optional) Specifies a brief summary of interface information.
inconsistentports	(Optional) Displays inconsistent port information.
interface *interface-id*	(Optional) Specifies a list of interfaces for which spanning-tree information appears. Enter each interface separated by a space. Ranges are not supported. Valid interfaces include physical ports and VLANs.
pathcost method	(Optional) Displays the default path cost method.
root	(Optional) Displays root-switch status and configuration.
summary	(Optional) Specifies a summary of port states.
total	(Optional) Displays the total lines of the spanning-tree state section.
uplinkfast	(Optional) Displays spanning-tree UplinkFast status.
vlan *vlan-id*	(Optional) Specifies the VLAN ID. The range is 1 to 1005.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5.2)WC(1)	This command was introduced.
12.1(6)EA2	The *bridge-group* argument and the **active**, **backbonefast**, **blockedports**, **bridge**, **inconsistentports**, **pathcost method**, **root**, **total**, and **uplinkfast** keywords were added.
12.2(15)ZJ	This command was implemented on the Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

If the variable *vlan-id* is omitted, the command applies to the spanning-tree instance for all VLANs.

Examples

The following is sample output from the **show spanning-tree summary** command:

```
Switch# show spanning-tree summary

UplinkFast is disabled

Name                 Blocking Listening Learning Forwarding STP Active
-------------------- -------- --------- -------- ---------- ----------
VLAN1                23       0         0        1          24
-------------------- -------- --------- -------- ---------- ----------
              1 VLAN 23       0         0        1          24
```

The following is sample output from the **show spanning-tree brief** command:

```
Switch# show spanning-tree brief
VLAN1
  Spanning tree enabled protocol IEEE
  ROOT ID    Priority 32768
             Address 0030.7172.66c4
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
VLAN1
  Spanning tree enabled protocol IEEE
  ROOT ID    Priority 32768
             Address 0030.7172.66c4
Port                           Designated
Name    Port ID Prio Cost Sts  Cost  Bridge ID      Port ID
------- ------- ---- ---- ---  ----  -------------- -------
Fa0/11  128.17  128  100  BLK  38    0404.0400.0001 128.17
Fa0/12  128.18  128  100  BLK  38    0404.0400.0001 128.18
Fa0/13  128.19  128  100  BLK  38    0404.0400.0001 128.19
Fa0/14  128.20  128  100  BLK  38    0404.0400.0001 128.20
Fa0/15  128.21  128  100  BLK  38    0404.0400.0001 128.21
Fa0/16  128.22  128  100  BLK  38    0404.0400.0001 128.22
Fa0/17  128.23  128  100  BLK  38    0404.0400.0001 128.23
Fa0/18  128.24  128  100  BLK  38    0404.0400.0001 128.24
Fa0/19  128.25  128  100  BLK  38    0404.0400.0001 128.25
Fa0/20  128.26  128  100  BLK  38    0404.0400.0001 128.26
Fa0/21  128.27  128  100  BLK  38    0404.0400.0001 128.27

Port                           Designated
Name    Port ID Prio Cost Sts  Cost  Bridge ID      Port ID
------- ------- ---- ---- ---  ----  -------------- -------
Fa0/22  128.28  128  100  BLK  38    0404.0400.0001 128.28
Fa0/23  128.29  128  100  BLK  38    0404.0400.0001 128.29
Fa0/24  128.30  128  100  BLK  38    0404.0400.0001 128.30 Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
```

The following is sample output from the **show spanning-tree vlan 1** command:

```
Switch# show spanning-tree vlan 1

Spanning tree 1 is executing the IEEE compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 00e0.1eb2.ddc0
  Configured hello time 2, max age 20, forward delay 15
  Current root has priority 32768, address 0010.0b3f.ac80
  Root port is 5, cost of root path is 10
  Topology change flag not set, detected flag not set, changes 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

Interface Fa0/1  in Spanning tree 1 is down
   Port path cost 100, Port priority 128
   Designated root has priority 32768, address 0010.0b3f.ac80
   Designated bridge has priority 32768, address 00e0.1eb2.ddc0
   Designated port is 1, path cost 10
   Timers: message age 0, forward delay 0, hold 0
   BPDU: sent 0, received 0
   .
   .
   .
```

The following is sample output from the **show spanning-tree interface fastethernet0/3** command:

```
Switch# show spanning-tree interface fastethernet0/3

Interface Fa0/3 (port 3) in Spanning tree 1 is down
   Port path cost 100, Port priority 128
   Designated root has priority 6000, address 0090.2bba.7a40
   Designated bridge has priority 32768, address 00e0.1e9f.4abf
   Designated port is 3, path cost 410
   Timers: message age 0, forward delay 0, hold 0
   BPDU: sent 0, received 0
```

Related Commands

Command	Description
spanning-tree vlan	Enables STP on a VLAN.

show storm-control

To display the packet-storm control information, use the **show storm-control** command in privileged EXEC mode. This command also displays the action that the switch takes when the thresholds are reached.

show storm-control [*interface-id*] [**broadcast** | **multicast** | **unicast** | **history**]

Syntax Description

interface-id	(Optional) Port for which information is to be displayed.
broadcast	(Optional) Displays broadcast storm information.
multicast	(Optional) Displays multicast storm information.
unicast	(Optional) Displays unicast storm information.
history	(Optional) Displays storm history on a per-port basis.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(6)EA2	This command was introduced. It replaced the **show port storm-control** command.
12.2(15)ZJ	This command was implemented on the Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

If the *interface-id* value is omitted, the **show storm-control** command displays storm-control settings for all ports on the switch.

You can display broadcast, multicast, or unicast packet-storm information by using the corresponding keyword. When no option is specified, the default is to display broadcast storm-control information.

Examples

The following is sample output from the **show storm-control broadcast** command:

```
Switch# show storm-control broadcast

Interface  Filter State   Upper    Lower    Current
---------  -------------  -------  -------  -------
Fa0/1      <inactive>     100.00%  100.00%    0.00%
Fa0/2      <inactive>     100.00%  100.00%    0.00%
Fa0/3      <inactive>     100.00%  100.00%    0.00%
Fa0/4      Forwarding      30.00%   20.00%   20.32%
.
.
.
```

Table 22 describes the fields shown in the display.

Table 22 show storm-control Field Descriptions

Field	Description
Interface	Displays the ID of the interface.
Filter State	Displays the status of the filter: • Blocking—Storm control is enabled, action is filter, and a storm has occurred. • Forwarding—Storm control is enabled, and a storm has not occurred. • Inactive—Storm control is disabled. • Shutdown—Storm control is enabled, the action is to shut down, and a storm has occurred. **Note** If an interface is disabled by a broadcast, multicast, or unicast storm, the filter state for all traffic types is *shutdown*.
Upper	Displays the rising suppression level as a percentage of total available bandwidth.
Lower	Displays the falling suppression level as a percentage of total available bandwidth.
Current	Displays the bandwidth utilization of a specific traffic type as a percentage of total available bandwidth. This field is valid only when storm control is enabled.

The following is sample output from the **show storm-control fastethernet0/4 history** command, which displays the ten most recent storm events for an interface:

```
Switch# show storm-control fastethernet0/4 history

 Interface Fa0/4 Storm Event History

 Event Type          Event Start Time  Duration (seconds)
 ------------------  ----------------  ------------------
 Unicast             04:58:18          206
 Broadcast           05:01:54          n/a
 Multicast           05:01:54          n/a
 Unicast             05:01:54          108
 Broadcast           05:05:00          n/a
 Multicast           05:05:00          n/a
 Unicast             05:06:00          n/a
 Broadcast           05:09:39          n/a
 Multicast           05:09:39          n/a
 Broadcast           05:11:32          172
```

Note The duration field could be *n/a* when a storm is still present or when a new storm of a different type occurs before the current storm ends.

Related Commands

Command	Description
storm-control	Enables broadcast, multicast, or unicast storm control on a port.

spanning-tree backbonefast

To enable the BackboneFast feature, use the **spanning-tree backbonefast** command in global configuration mode. To return to the default setting, use the **no** form of the command.

spanning-tree backbonefast

no spanning-tree backbonefast

Syntax Description

This command has no arguments or keywords.

Defaults

BackboneFast is disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced.
12.2(15)ZJ	This command was implemented on the Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

BackboneFast should be enabled on all of the Catalyst 2950 switches to allow for the detection of indirect link failures and to start the spanning-tree reconfiguration sooner.

Examples

The following example shows how to enable BackboneFast on the switch:

```
Switch(config)# spanning-tree backbonefast
```

You can verify your settings by entering the **show spanning-tree** privileged EXEC command.

Related Commands

Command	Description
show spanning-tree	Displays spanning-tree information for the specified spanning-tree instances.

storm-control

To enable broadcast, multicast, or unicast storm control on a port and to specify the action taken when a storm occurs on a port, use the **storm-control** command in interface configuration mode. To disable storm control for broadcast, multicast, or unicast traffic and disable the specified storm-control action, use the **no** form of this command.

storm-control {{{**broadcast** | **multicast** | **unicast**} **level** *level* [*lower-level*]} | **action shutdown**}

no storm-control {{**broadcast** | **multicast** | **unicast**} **level**} | **action**}

Syntax Description

{**broadcast** | **multicast** | **unicast**}	Determines the type of packet-storm suppression. • **broadcast**—Enable broadcast storm control on the port. • **multicast**—Enable multicast storm control on the port. • **unicast**—Enable unicast storm control on the port.
level *level* [*lower-level*]	Defines the rising and falling suppression levels. • *level*—Rising suppression level as a percent of total bandwidth, up to two decimal places; valid values are from 0 to 100 percent. Block the flooding of storm packets when the value specified for *level* is reached. • *lower level*—(Optional) Falling suppression level as a percent of total bandwidth, up to two decimal places; valid values are from 0 to 100. This value must be less than the rising suppression value.
action	Action taken when a storm occurs on a port. The default action is to filter traffic.
shutdown	Disables the port during a storm.

Defaults

Broadcast, multicast, and unicast storm control are disabled.

The default action is to filter traffic.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(6)EA2	This command was introduced. It replaced the **port storm-control** command.
12.2(15)ZJ	This command was implemented on the Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Use the **storm-control** command to enable or disable broadcast, multicast, or unicast storm control on a port. After a port is disabled during a storm, use the **no shutdown** interface configuration command to enable the port.

The suppression levels are entered as a percentage of total bandwidth. A suppression value of 100 percent means that no limit is placed on the specified traffic type. This feature is enabled only when the rising suppression level is less than 100 percent. If no other storm-control configuration is specified, the default action is to filter the traffic causing the storm.

When a storm occurs and the action is to filter traffic, if the falling suppression level is not specified, the switch blocks all traffic until the traffic rate drops below the rising suppression level. If the falling suppression level is specified, the switch blocks traffic until the traffic rate drops below this level.

When a multicast or unicast storm occurs and the action is to filter traffic, the switch blocks all traffic (broadcast, multicast, and unicast traffic) and sends only Spanning Tree Protocol (STP) packets.

When a broadcast storm occurs and the action is to filter traffic, the switch blocks only broadcast traffic.

Examples

The following example shows how to enable broadcast storm control on a port with a 75.67 percent rising suppression level:

```
Switch(config-if)# storm-control broadcast level 75.67
```

The following example shows how to enable multicast storm control on a port with a 87 percent rising suppression level and a 65 percent falling suppression level:

```
Switch(config-if)# storm-control multicast level 87 65
```

The following example shows how to enable the **shutdown** action on a port:

```
Switch(config-if)# storm-control action shutdown
```

The following example shows how to disable the **shutdown** action on a port:

```
Switch(config-if)# no storm-control action shutdown
```

You can verify your settings by entering the **show storm-control** privileged EXEC command.

Related Commands

Command	Description
show storm-control	Displays the packet-storm control information.

switchport

To set an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration, use the **switchport** command in interface configuration mode. To set an interface in Layer 3 mode, use the **no** form of this command.

switchport

no switchport

Use the **no switchport** command (without parameters) to set the interface to the routed-interface status and to erase all Layer 2 configurations. You must use this command before assigning an IP address to a routed port.

Note If an interface is to be configured as a Layer 3 interface, you must first enter the **switchport** command with no keywords to configure the interface as a Layer 2 port. Then you can enter additional switchport commands with keywords.

Syntax Description

This command has no arguments or keywords.

Defaults

By default, all interfaces are in Layer 2 mode.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(4)EA1	This command was introduced.
12.2(15)ZJ	This command was implemented on the Cisco 2600 series, Cisco 3600 series, and Cisco 3700 series routers.

Usage Guidelines

Entering the **no switchport** command shuts the port down and then reenables it, which might generate messages on the device to which the port is connected.

Examples

The following example shows how to cause an interface to cease operating as a Layer 2 port and become a Cisco-routed port:

```
Switch(config-if)# no switchport
```

The following example shows how to cause the port interface to cease operating as a Cisco-routed port and convert to a Layer 2-switched interface:

```
Switch(config-if)# switchport
```

Note The **switchport** command without keywords is not used on platforms that do not support Cisco-routed ports. All physical ports on such platforms are assumed to be Layer 2-switched interfaces.

You can verify the switchport status of an interface by entering the **show running-config** privileged EXEC command.

Related Commands

Command	Description
show interfaces switchport	Displays the administrative and operational status of a switching (nonrouting) port, including port blocking and port protection settings.
show running-config	Displays the current operating configuration.

Glossary

802.1d—IEEE standard for MAC bridges.

802.1p—IEEE standard for queuing and multicast support.

802.1q—IEEE standard for VLAN frame tagging.

802.1x—IEEE standard for port-based network access control.

ACE—access control entry. Entry in an access control list.

ACL—access control list. Used for security or as a general means to classify traffic.

AgPort—aggregate port (another name for EtherChannel).

ATM—Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media such as E3, SONET, and T3.

authentication server—Entity that validates the credentials of a host trying to obtain access to the network.

authenticator—Entity that enforces authentication rules for hosts connecting to a LAN via one of its ports.

authorization state—The state of a controlled port. It can be authorized (access allowed) or unauthorized (access denied).

AVVID—Architecture for voice, video, and integrated data.

BRI—Basic Rate Interface. ISDN interface comprising two B channels and one D channel for circuit-switched communication of voice, video, and data.

CAC—connection admission control. Set of actions taken by each ATM switch during connection setup to determine whether a connection's requested QoS will violate the QoS guarantees for established connections. CAC is also used when routing a connection request through an ATM network.

candidate—Switch that is not part of a cluster, but is eligible to join a cluster because it meets the qualification criteria of the cluster.

classification—Process of sorting incoming packets by examining fields of interest in the packet header. Fields can be addresses, ports, DSCP value, and so on.

CBWFQ—class-based weighted fair queuing. Extends the standard WFQ functionality to provide support for user-defined traffic classes.

CCN—Cisco Communications Network (Cisco IP phones and IP PBX).

cluster—Group of switches that are managed as a single device. A cluster comprises one commander and multiple members.

cluster commander—Switch that provides the primary management interface to a cluster.

cluster member—Member switch that is managed through the cluster commander.

CoS—Class of Service. An indication of how an upper-layer protocol requires a lower-layer protocol to treat its messages. In SNA subarea routing, CoS definitions are used by subarea nodes to determine the optimal route to establish a session. A CoS definition comprises a virtual route number and a transmission priority field. Also called ToS.

DSCP—differentiated services code point. In QoS, a modification of the type of service byte. Six bits of this byte are being reallocated for use as the DSCP field, where each DSCP specifies a particular per-hop behavior that is applied to a packet.

DSL—digital subscriber line. Public network technology that delivers high bandwidth over conventional copper wiring at limited distances. There are four types of DSL: ADSL, HDSL, SDSL, and VDSL. All are provisioned via modem pairs, with one modem at a central office and the other at the customer site. Because most DSL technologies do not use the whole bandwidth of the twisted pair, there is room remaining for a voice channel.

EAP—Extensible Authentication Protocol. A mechanism (originally designed for PPP in RFC 2284) that provides authentication of hosts requesting access to a network.

EAPOL—EAP over LAN. EAP over LAN framing instead of PPP.

Frame Relay—The capability to carry normal telephony-style voice over an IP-based network with POTS-like functionality, reliability, and voice quality. VoIP lets a router carry voice traffic (such as telephone calls and faxes) over an IP network. In VoIP, the DSP segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

FXO—Foreign Exchange Office. An FXO interface connects to the Public Switched Telephone Network (PSTN) central office and is the interface offered on a standard telephone. Cisco's FX interface is an RJ-11 connector that allows an analog connection at the PSTN's central office or to a station interface on a PBX.

FXS—Foreign Exchange Station. An FXS interface connects directly to a standard telephone and supplies ring, voltage, and dial tone. Cisco's FXS interface is an RJ-11 connector that allows connections to basic telephone service equipment, keysets, and PBXs.

HSRP—Hot Standby Router Protocol. Provides high network availability and transparent network topology changes. HSRP creates a hot standby router group with a lead router that services all packets sent to the hot standby address. The lead router is monitored by other routers in the group, and if it fails, one of these standby routers inherits the lead position and the hot standby group address.

IGMP—Internet Group Management Protocol. Used by IP hosts to report their multicast group memberships to an adjacent multicast router.

ISL—InterSwitch Link, which is used to carry traffic for multiple VLANs. A method of encapsulating tagged LAN frames and transporting them over a full-duplex, point-to-point Ethernet link. The encapsulated frames can be token-ring or Fast Ethernet and are carried unchanged from transmitter to receiver.

MIB—Management Information Base. Database of network management information that is used and maintained by a network management protocol, such as SNMP or CMIP. The value of a MIB object can be changed or retrieved using SNMP or CMIP commands, usually through a GUI network management system. MIB objects are organized in a tree structure that includes public (standard) and private (proprietary) branches.

policing—Process of ensuring whether a stream of classified incoming packets conforms to a particular traffic profile. An action (drop or remark) is taken based on the rate of arrival of packets.

PRI—primary rate interface. ISDN interface to primary rate access. Primary rate access consists of one 64-kbps D channel and 23 (T1) or 30 (E1) B channels for voice or data. Compare with BRI.

PSTN—public switched telephone network. General term referring to the variety of telephone networks and services in place worldwide. Also called POTS.

PVC—permanent virtual circuit. Virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and tear down in situations where certain virtual circuits must exist all the time. In ATM terminology, called a permanent virtual connection. Compare with SVC.

PVST—Per-VLAN spanning tree. Support for dot1q trunks to map multiple spanning trees to a single spanning tree.

QoS—quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

RADIUS—Remote Access Dial-In User Service. A service used to authenticate and authorize clients.

RMON—remote monitoring. MIB agent specification described in RFC 1271 that defines functions for the remote monitoring of networked devices. The RMON specification provides numerous monitoring, problem detection, and reporting capabilities.

RSVP—Resource Reservation Protocol. Protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive. RSVP depends on IPv6. Also known as Resource Reservation Setup Protocol.

SIP—Session Initiation Protocol. Protocol developed by the IETF MMUSIC Working Group as an alternative to H.323. SIP features are compliant with IETF RFC 2543, which was published in March 1999. SIP equips platforms to signal the setup of voice and multimedia calls over IP networks.

SNMP—Simple Network Management Protocol. Network management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices and to manage configurations, statistics collection, performance, and security.

stacking—Connecting two switches so they behave as one entity for management purposes. Regarding an Ethernet switch network module, stacking means connecting two Ethernet switch network modules inside a chassis so that they behave as one switch.

STP—Spanning Tree Protocol. Bridge protocol that uses the spanning-tree algorithm, which enables a learning bridge to dynamically work around loops in a network topology by creating a spanning tree. Bridges exchange BPDU messages with other bridges to detect loops and then remove the loops by shutting down selected bridge interfaces. Refers to both the IEEE 802.1 Spanning-Tree Protocol standard and the earlier Digital Equipment Corporation Spanning-Tree Protocol upon which it is based. The IEEE version supports bridge domains and allows the bridge to construct a loop-free topology across an extended LAN. The IEEE version generally is preferred over the Digital version.

supplicant—Entity requesting access to the network via the authenticator.

SVI—Switch Virtual Interface. Represents a VLAN of switch ports as one interface to the routing or bridging function in a system.

VBR—variable bit rate. QoS class defined by the ATM Forum for ATM networks. VBR is subdivided into a real time (RT) class and non-real time (NRT) class. VBR (RT) is used for connections in which there is a fixed timing relationship between samples. VBR (NRT) is used for connections in which there is no fixed timing relationship between samples but that still need a guaranteed QoS. Compare with ABR, CBR, and UBR.

VLAN—virtual LAN. Group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same wire, when in fact they are on separate LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible.

VoIP—Voice over IP. Ability to carry normal telephony-style voice over an IP-based internet with POTS-like functionality, reliability, and voice quality. VoIP enables a router to carry voice traffic (such as telephone calls and faxes) over an IP network. In VoIP, the DSP segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

VoIPoFR—Voice-over-IP over Frame-Relay.

VPN—Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses “tunneling” to encrypt all information at the IP level.

VQP—VLAN Query Protocol.

VTP—VLAN Trunking Protocol.

WAN—wide area network. A communications network that covers a wide geographic area such as state or country. A LAN (local area network) is within a building or complex, and a MAN (metropolitan area network) generally covers a city or suburb.

WFQ—weighted fair queuing. In QoS, a flow-based queuing algorithm that schedules low-volume traffic first while letting high-volume traffic share the remaining bandwidth. This is handled by assigning a weight to each flow, where lower weights are the first to be serviced.

WRR—Weighted Round-Robin. Type of round-robin scheduling that prevents low-priority queues from being completely neglected during periods of high-priority traffic. The WRR scheduler transmits some packets from each queue in turn. The number of packets it transmits corresponds to the relative importance of the queue.