6500

Cisco 6500

Student Guide

Version 1.0

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the Cisco Web site at www.cisco.com/go/offices.

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Hardware

Overview

This module introduces many of the hardware components and specifications that make up the Cisco Catalyst 6500 Series of switches.

Upon completing this module, you will be able to:

- List the functions of the Catalyst 6500's system modules
- Describe Ethernet connectivity options
- List WAN connectivity options
- Describe Voice options
- List Web and Security services hardware

Outline

The module contains these lessons:

- Chassis and System Modules
- Ethernet Modules
- WAN Options
- Voice Services Options
- Web Hardware
- Security Hardware

Chassis and System Modules

Overview

The Catalyst 6500 switch is available in three different models: 6506, 6509, and 6513. The number after 65xx indicates the number of slots that are available for modules. One of the main characteristics of the Catalyst 6500 family is the wide variety of modules supported. This lesson will cover the basics of the system chassis and then detail the various modules available for the Catalyst 6500.

Importance

This lesson is a fundamental building block for the rest of the course. Before you can configure the Catalyst 6500, you must understand the hardware involved.

Objectives

Upon completing this lesson, you will be able to:

- Describe the specifications of the Catalyst 6500
- Distinguish between Native Mode and Hybrid Mode
- Identify the functions of the MSFC and PFC modules
- Describe the Catalyst 6500 architecture

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Have a basic knowledge of Layer 2 networking devices, such as hubs and switches
- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have the equivalent knowledge

Outline

This lesson includes these sections:

- Overview
- The Catalyst 6500 Series Switch Family
- Supervisor Engines
- Native Mode -vs- Hybrid Mode
- Supervisor Daughter Cards
- Switch Fabric Module
- Backplane Speed Comparison
- Summary
- Lesson Assessment (Quiz)

The **Catalyst 6500 Series Switch Family**

The Catalyst 6500 switch is available in different models, depending on your port density and performance requirements.

- **Scalable, intelligent multilayer switching for enterprise and ISP infrastructures**
- **Scalable, intelligent multilayer switching for enterprise and ISP infrastructures**
- **Integrated voice solutions**
- **LAN/WAN integration**

The Catalyst 6500 Series switch family was originally targeted for the distribution layer, in Cisco's three-layered design model (i.e., Access, Distribution, and Core). However, with its performance, port density, and flexibility, the Catalyst 6500 may be a candidate for deployment in any of these layers.

Unlike traditional Layer 2 switches, the Catalyst 6500 supports routing functions, advanced quality of service (QoS) features, multicast services, voice services, and WAN connectivity. Also, with the addition of add-on modules such as the Switch Fabric Module, the Policy Feature Card, and the Multilayer Switch Feature Card (all of which will be discussed in this module) the Catalyst 6500 is among the fastest Layer 3 switches on the market.

Currently the Catalyst 6500 is available in four models:

Table 1-1 Catalyst 6500 Models

Model	Description
Catalyst 6506	Contains six modular slots, scalable to 240 10/100 Ethernet ports
Catalyst 6509	Contains nine modular slots, scalable to 384 10/100 Ethernet ports
Catalyst 6509-NEBS	Contains nine modular slots in a vertical chassis, with front-to-back air flow, and is compliant with NEBS Level 3 standards
Catalyst 6513	Contains thirteen modular slots, scalable to 576 10/100 Ethernet ports or 192 Gigabit Ethernet ports

Supervisor Engines

Currently there are two options available for the Supervisor Engine on the Catalyst 6500. As a network engineer supporting the Catalyst 6500, you will need to understand the differences between the two.

- Fast EtherChannel and Gigabit EtherChannel
- Dynamic VLANs
- ISL and IEEE 802.1q trunking
- VTP
- Jumbo frame support for Gigabit Ethernet
- Protocol filtering
- Link load balancing
- Multicast support
- IP permit lists
- UplinkFast and BackboneFast

The Catalyst 6500 Series includes two versions of the Supervisor Engine. The Supervisor Engine is required for system operations; a chassis without a Supervisor will not operate. The Supervisor Engine uses slot 1 in the chassis. The second slot in the system, slot 2, can be used for a secondary redundant supervisor engine. Note that because of the switching implementation of the Catalyst 6000 and 6500 only one Supervisor Engine needs to be active at any given time. However, with the High Availability feature enabled, both supervisors maintain the same state information, including Spanning-Tree topology, forwarding tables, and management information, so that if the primary supervisor fails, the redundant engine can take over within two seconds.

To address the needs of different customers who deploy the Catalyst 6500 in varying applications, Cisco provides two models of the Supervisor Engine.

Supervisor Engine 1A

The Supervisor 1A, which was the first switching engine for the Catalyst 6000 Family, provides performance levels of up to 15 million pps (Packets per second) using a cache-based switching scheme. The Supervisor 1A card has three main components: the Network Management Processor (NMP), the Multilayer Switch Feature Card (MSFC) and the Policy Feature Card (PFC). Each component provides a critical function to the network.

The Supervisor 1A card is available in three flavors:

- Basic Layer 2 switching with no Layer 3-based QoS or security access control lists (ACLs), although port-based class of service (CoS) and destination MAC address-based CoS is supported. Basic switching based on the MAC address is supported.

- Supervisor 1A with the PFC, which provides Layer 2 switching with Layer 3 services (including QoS and security ACLs). QoS classification and queuing, as well as security filtering, is supported at data rates of up to 15 million pps. This functionality is supported at Layer 2 and 3, even though Layer 3 switching and routing are not performed.

- Supervisor 1A with PFC/MSFC-1 (or MSFC-2), which provides full Layer 3 switching and routing. This combination enables the Catalyst 6500 to route IP and Internet Packet Exchange (IPX) traffic at rates of up to 15 million pps.

Supervisor Engine 2

The Supervisor Engine 2 is designed specifically for service provider and high-end enterprise core applications. This supervisor engine provides forwarding capability of up to 30 million pps when using fabric-enabled line cards and the Switch Fabric Module (SFM). Both the Supervisor 2 and the SFM must be used simultaneously to achieve this type of throughput. An important difference between the Supervisor 1A and Supervisor 2 is that Supervisor 2 supports Cisco Express Forwarding (CEF) in hardware. CEF is a switching implementation that is based on the topology of the network rather than the traffic flow. This causes the control plane of the Catalyst 6500 to converge faster in the event of a change in the network topology and perform lookups for millions of flows (which occurs in Internet Service Provider [ISP] deployments).

The Supervisor 2 card comes in two flavors:

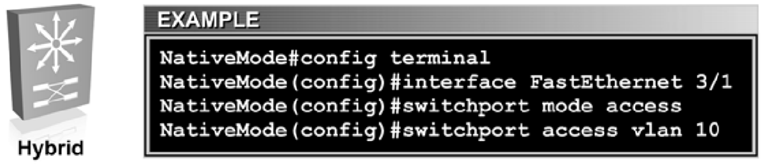
- Supervisor 2 with PFC-2, which provides QoS, Private Virtual LAN (PVLAN), and ACL functionality at rates of up to 30 million pps with no performance penalty.

- Supervisor 2 with PFC-2/MSFC-2, which enables full routing on the Catalyst 6500. This supervisor enables the Catalyst 6500 to provide Internet-class routing and high performance.

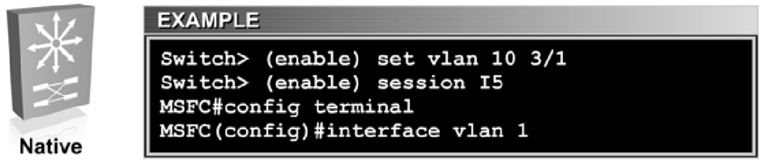
The Supervisor Engine 2 can be used in the Catalyst 6000 and Catalyst 6500 chassis. The SFM requires the use of a Supervisor 2 card; however the Supervisor 2 card can operate without the SFM.

Native Mode -vs- Hybrid Mode

Another one of the Catalyst 6500's main characteristics is flexibility. This is displayed by its support of two different operating systems



```
NativeMode#config terminal
NativeMode(config)#interface FastEthernet 3/1
NativeMode(config)#switchport mode access
NativeMode(config)#switchport access vlan 10
```



```
Switch> (enable) set vlan 10 3/1
Switch> (enable) session I5
MSFC#config terminal
MSFC(config)#interface vlan 1
```

Catalyst 6000 switches have the option of running in one of two operating system environments.

CatOS (Hybrid): This implementation is logically equivalent to a Catalyst 5000 series switch with a Route Switch Module (RSM). When running in CatOS mode, there are two separate software images. The MSFC runs a traditional Native IOS image, and the supervisor engine runs a traditional CatOS. Each device has its own configuration file.

Native IOS: This implementation provides a single, "router-like" interface. The division between the router (referred to as the Route Processor or RP) and switch supervisor (referred to as the Switch Processor or SP) is transparent to the user, with a single console connection, configuration file, and software image. (Note: An MSFC1 boot image is always required to allow the MSFC1 to load properly. The boot image is required for hardware support and provides a backup for emergency recovery situations. The software image actually loads the necessary software for the full functionality of the router.). A Policy Feature Card (PFC) is required in addition to an MSFC.

The following table outlines the primary system differences between these two modes of operation.

Table 1-2 Primary Differences between CatOS and Native IOS modes

Features	CatOS	Native IOS
Configuration File	Two configuration files: one for the supervisor <NMP> and one for the MSFC	One configuration file
Software image	Two images: one for the supervisor engine and one for the MSFC	One software image A MSFC boot image is also required to allow the MSFC to load properly
Default Port Mode	Every port is a Layer 2 switched port	Every port is a Layer 3 routed port (interface)
Default Port Status	Every port is enabled	Every port (interface) is in the shutdown state
Configuration Commands Format	The command keyword set precedes each configuration command.	Native IOS command structure with global and interface level commands
Configuration mode	No configuration mode (set, clear, and show commands)	The commands configure terminal and VLAN database activate configuration modes
Removing/Changing the Configuration	Done via the use of the clear, and disable commands	Same as Native IOS command structure, keyword no negates a command

In this course, where appropriate, the syntax for both the Native and Hybrid mode configuration will be shown. For comparisons of what features are supported in each mode, refer to:

http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/prodlit/supe_ds.htm

Supervisor Daughter Cards

The Supervisor Engine (1A and 2) supports daughter cards to provide features, such as Layer 3 routing, VLAN Access Control Lists, and Advanced QoS.

Policy Feature Card (PFC)

- Accelerated SLB
- Advanced Multiprotocol Packet Classification
- Multiprotocol QoS
- Traffic management
- Multiprotocol Multilayer Switching (with MSFC)

The Policy Feature Card (PFC) is a daughter board that mounts directly on the Supervisor Engine (Version 1A or higher). The following are some of the enhancements that the PFC provides to the Catalyst 6500:

- The PFC enables MultiLayer Switching (MLS) for the Catalyst 6500 routing module, the Multilayer Switch Feature Card (MSFC), for IP and IPX protocols.

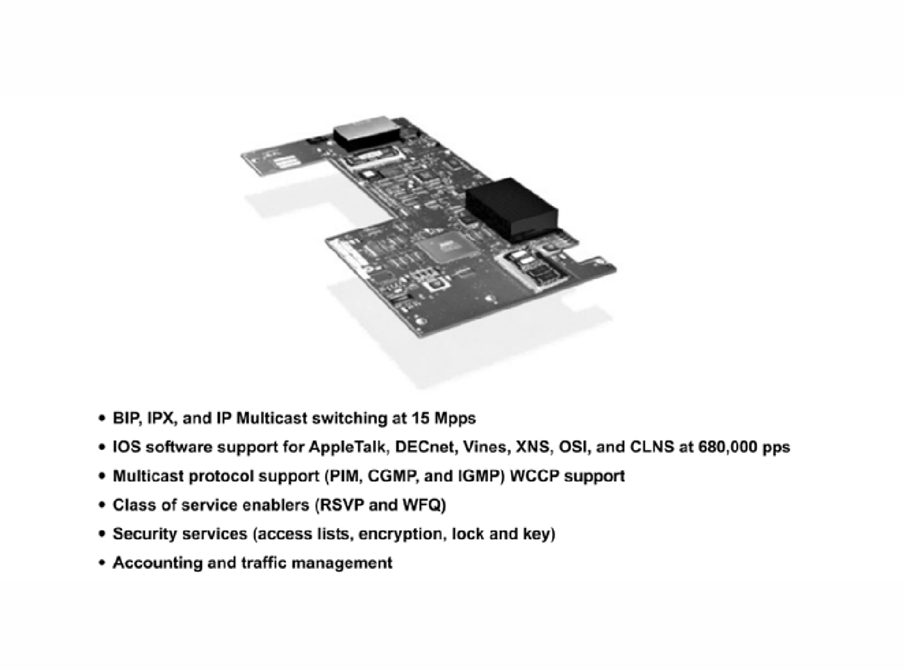
- In a multicast environment, the PFC supports IGMP Snooping, which allows the switch to dynamically learn which of its ports are connected to multicast clients for specific multicast group addresses.

- The PFC supports VACLs (VLAN Access Control Lists), which allow the filtering, capture, and redirection of traffic within a VLAN.

- While the Catalyst supports a range of Quality of Service features (e.g., LLQ) that are seen on IOS-based routers, the PFC supports QoS ACLs, which allow the switch to perform policing functions for traffic identified at Layer 2 and at Layer 3.

- When used in conjunction with Cisco’s LocalDirector, the PFC supports Server Load Balancing (SLB) at wire speed.

Multilayer Switch Feature Card (MSFC)



- BIP, IPX, and IP Multicast switching at 15 Mpps
- IOS software support for AppleTalk, DECnet, Vines, XNS, OSI, and CLNS at 680,000 pps
- Multicast protocol support (PIM, CGMP, and IGMP) WCCP support
- Class of service enablers (RSVP and WFQ)
- Security services (access lists, encryption, lock and key)
- Accounting and traffic management

The addition of the MSFC daughter board on the Catalyst 6500 Supervisor Engine transforms the Catalyst from a Layer 2 switch into a Layer 3 switch. Just as router capabilities can be added to the Catalyst 5000 and 5500 Series of switches with the addition of an RSM (Route Switch Module), the addition of an MSFC provides routing capabilities to the Catalyst 6500.

A major advantage of the MSFC over the RSM is that it has a full-duplex 1 Gbps connection to the backplane, while the RSM only offers a half-duplex 400 Mbps connection to the switch's backplane. Additionally, while the RSM consumes valuable real estate by taking up a slot in the Catalyst 5000/5500 switches, the MSFC is a daughter board that mounts directly onto the Supervisor Engine.

There are two versions of the MSFC, both of which are supported on Supervisor Engines 1A and higher. The MSFC Version 1 can provide 15 Million pps of centralized forwarding for IP, IPX, and IP Multicast traffic, using MultiLayer Switching (MLS). However, with its support of Cisco Express Forwarding (CEF), the MSFC Version 2, when used with a Supervisor Engine 2 and a PFC Version 2, can provide up to 210 Million pps of distributed forwarding. Distributed forwarding is made possible by having distributed forwarding daughter boards mounted on selected line cards. The MSFC Version 2 also has other enhancements over the MSFC Version 1, such as support for 800 WAN interfaces and support for 1000 terminated VLANs.

While many vendors offer multilayer switching for IP, or IP/IPX, Cisco maintains support for other protocols as well, such as AppleTalk and Banyan Vines, in the Catalyst 6500. These protocols, however, do not participate in MLS. Therefore, their performance is limited to approximately 680,000 pps, because each packet has to be routed by the MSFC. Still, this performance rivals most of Cisco's other software-based routers.

Switch Fabric Module (SFM)

The addition of a Switch Fabric Module (SFM) allows the Catalyst 6500's backplane speed to scale up to 256 Gbps.

- Increases system bandwidth from 32 Gbps to 256 Gbps
- Must be installed in slot 5 or 6

The default bandwidth available on the backplane of the Catalyst 6500 Series switch is 32 Gbps. This 32 Gbps bus is shared by all slots for the serial transmission of data. Therefore, only two ports can be communicating at any given instant. Access to the bus is scheduled by the Fabric Arbitration ASIC (Application Specific Integrated Circuit).

With the addition of a Switch Fabric Module (SFM), the switch's backplane changes from a serially-accessed bus to a crossbar fabric. By using a crossbar fabric, many ports can be simultaneously transmitting and receiving data, providing a much higher aggregate throughput. Therefore, with the SFM installed in a Catalyst 6500, the backplane speed is increased from 32 Gbps to 256 Gbps.

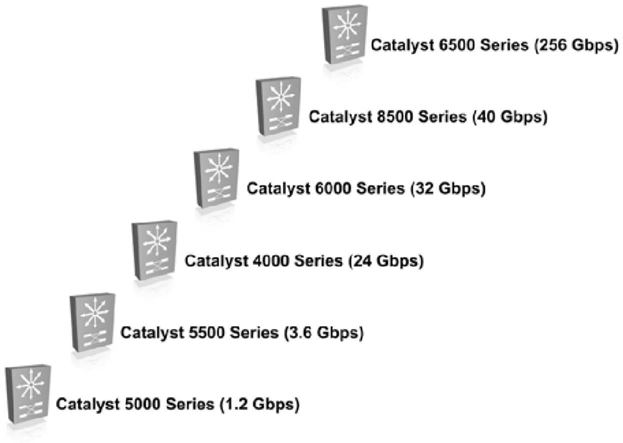
There are two versions of the SFM. While the SFM Version 1 can be used in Catalyst 6506 and 6509 models, it cannot be used in the Catalyst 6513. However, the SFM Version 2 can be used in the Catalyst 6513.

Two SFMs can be installed into the chassis for redundancy. If one of the SFMs fails, the other takes over, still providing a backplane speed of 256 Gbps. If both SFMs fail, the system reverts to its default backplane, and operates at 32 Gbps.

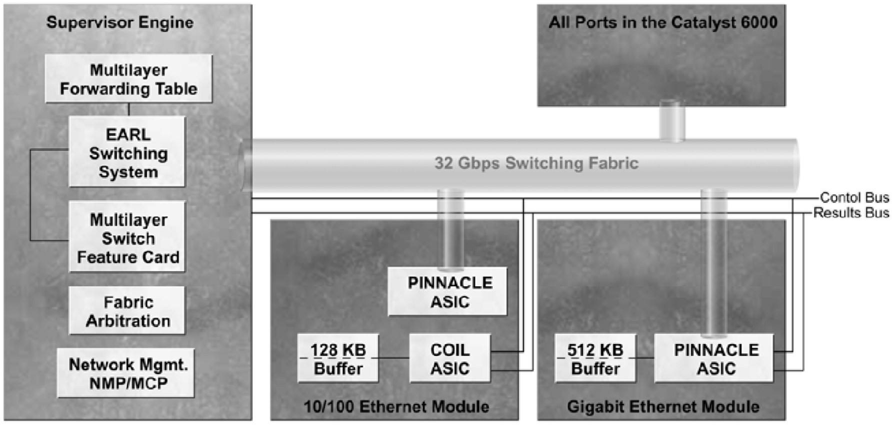
When installing SFM Version 1 modules, they should only be installed in slots 5 and/or 6. When installing SFM Version 2 modules, they should only be installed in slots 7 and/or 8.

Backplane Speed Comparison

This section compares the backplane speed of the Catalyst 6500 with the SFM installed to the capabilities of other Cisco Layer 3 switches. This section also discusses the use of the crossbar fabric in the SFM to achieve such increases in backplane speed.


Catalyst 6500 Series (256 Gbps)
Catalyst 8500 Series (40 Gbps)
Catalyst 6000 Series (32 Gbps)
Catalyst 4000 Series (24 Gbps)
Catalyst 5500 Series (3.6 Gbps)
Catalyst 5000 Series (1.2 Gbps)

With the addition of the Switch Fabric Module, the Catalyst 6500 Series switch has a backplane that far exceeds the capabilities of any other Cisco Layer 3 switch. In fact, the primary difference between the Catalyst 6000 Series and the 6500 Series is that the 6000 Series cannot accept a SFM, and is therefore limited to a backplane speed of 32 Gbps. The above diagram provides a backplane speed comparison of several of Cisco's popular Layer 3 switches.



The Catalyst 6000 system is based on a 32 Gbps advanced pipelining switching bus. The switching bus is a shared medium bus; that is, all the ports attached to the bus see all the frames that are transmitted across it. Coupled with the pipelining mechanism, this switching is very efficient because after a decision is made the switching engine orders the non-destination ports to ignore the frame.

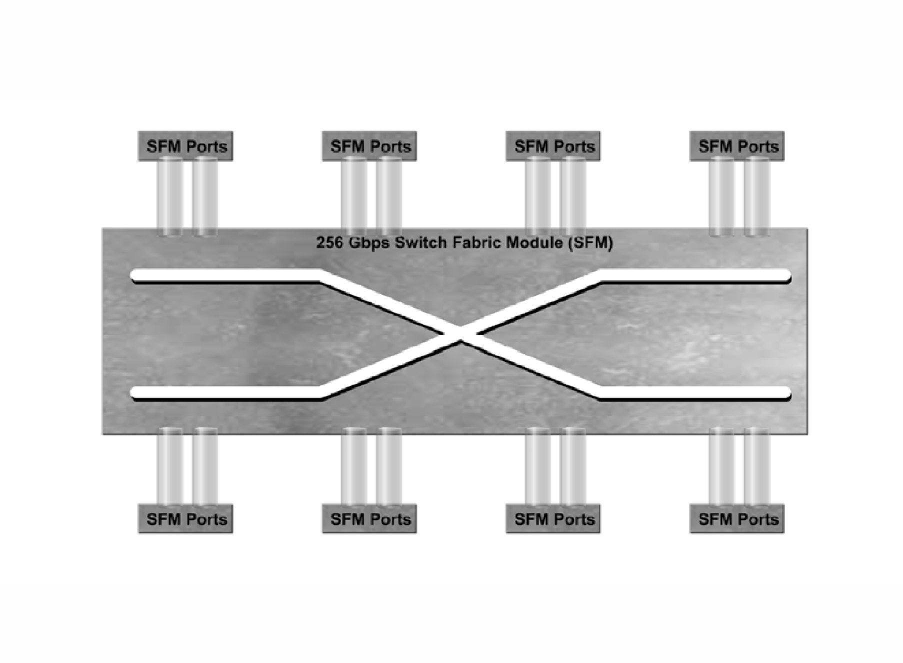
The Catalyst 6000 switching bus includes three distinct buses: the D-bus (or Data bus), the C-bus (or Control bus), and the R-bus (or Results bus). All non fabric-enabled line cards connect to the switching bus through the connectors on the right side of the chassis. The D-bus is the bus where data is forwarded from one port to another and realizes a bandwidth of 32 Gbps. The Results bus (R) takes information from the switching logic located on the Supervisor Engine back to all the ports on the switch. The control bus (C-bus) relays information between the port ASICs and the Network Management Processor (NMP).

Two notable features on the switching bus of the Catalyst 6000 are pipelining and burst mode. Pipelining enables the Catalyst 6000 Family switches to switch multiple frames onto the bus before obtaining the results of the first frame. Typically, on shared medium architectures, only a single frame or packet can reside on the bus at a time. The entire lookup process by the forwarding engine happens in parallel to the transfer of the frame across the switching bus.

If the frame is switched across the switching bus before the lookup is done, the switching bus is idle until the lookup is done. This is where pipelining comes into play. Ports are allowed to source frames on the switching bus before the results of the first frame lookup are done. The second frame (from any port) is switched across the bus and pipelined for a lookup operation at the forwarding engine. Thirty-one such frames can be switched across the switching bus before the result of the first frame is received. The 32nd frame must wait before it can be sourced on the switching bus.

The Burst-Mode feature enables the port to source multiple frames onto the switching bus. If the port sends just one frame each time, it is granted access to the data bus. There is a potentially unfair allocation of bandwidth to the bus when it is heavily loaded. For example, if two ports are trying to send data and one has 100-byte frames, while the other has 1000-byte frames, the port with the 1000-byte frames can switch 10 times as much data as the port with 100-byte frames. This is because they alternate in the arbitration and each port is allowed to send one frame at a time.

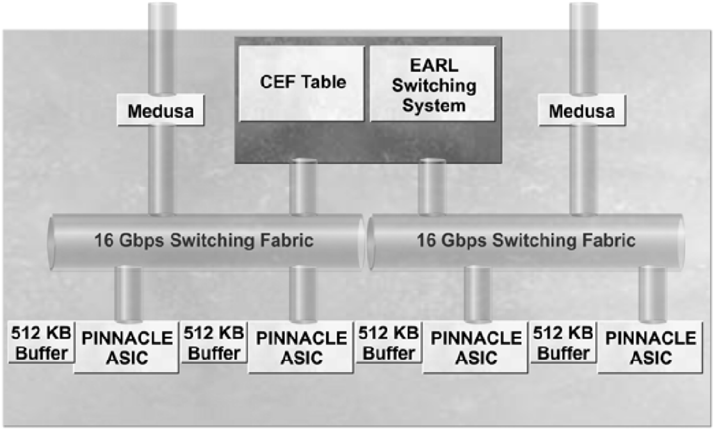
In the Catalyst 6000 Family switches, a port can send multiple frames to the switching bus in a manner that controls the amount of bandwidth it consumes regardless of the frame size. To accomplish this the port ASICs maintain a counter of the number of bytes transferred and compare it to a threshold. Provided that the count is below the threshold value, the port continues to send frames as long as it has data to send. When the count exceeds the threshold, the port stops sending data after completing the current frame and stops transmitting because the arbitration logic at this point senses the condition and removes bus access. The threshold value is a function of the number of ports in the system, their thresholds, empirical data, simulation results, and so forth. The system automatically computes the threshold value to ensure fair bandwidth distribution.



The Catalyst 6500 and the Switch Fabric Module (SFM) provide a 256-Gbps switching system with forwarding rates of over 100 million pps. The SFM uses the connectors on the left side of the Catalyst 6500 chassis. Note that because these connectors are not available in the Catalyst 6000, this chassis cannot use the SFM. The SFM uses a 256-Gbps crossbar switching fabric to interconnect the line cards on the switch. The diagram shown is a logical representation of the SFM.

The SFM can best be thought of as a 16-port "switch," with the ports actually connecting to the line cards. In the Catalyst 6500, each slot in the chassis receives two crossbar ports, and each port is clocked at 8 Gbps (the actual bandwidth is 16 Gbps; there is one 8 Gbps path for transmitting into the crossbar and another 8 Gbps path for transmitting out of the crossbar). The fabric-enabled modules connect to one of the ports on the crossbar, providing 8 Gbps access into the switching fabric. The fabric-only line cards attach to both ports per slot into the crossbar, allowing them 16 Gbps of connectivity.

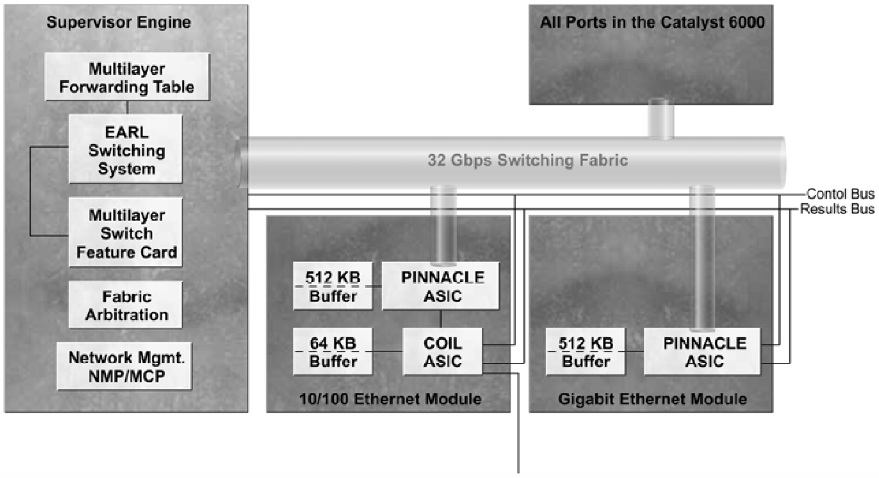
The Catalyst 6500 SFM uses overspeed to eliminate congestion and head-of-line blocking. Overspeed is a concept by which the internal "paths" *within* the crossbar fabric are clocked at a speed faster than the input rates *into* the crossbar. This allows packets to be switched out of the source module through the fabric to the output line card at high data rates. The SFM uses 3x overspeed, meaning that each internal trace is clocked at 24 Gbps relative to the input rate, which is clocked at 8 Gbps.



Each of the line cards connecting to the SFM uses a local switching fabric. The fabric-enabled cards, such as the WS-X6516, support the Distributed Forwarding Card (DFC) to enable high-speed switching. These line cards have connectivity to one channel port on the SFM and also have a connection into the 32 Gbps centralized switching bus. The fabric-only line cards, such as the WS-X6816, connect only into the SFM via dual fabric channels.

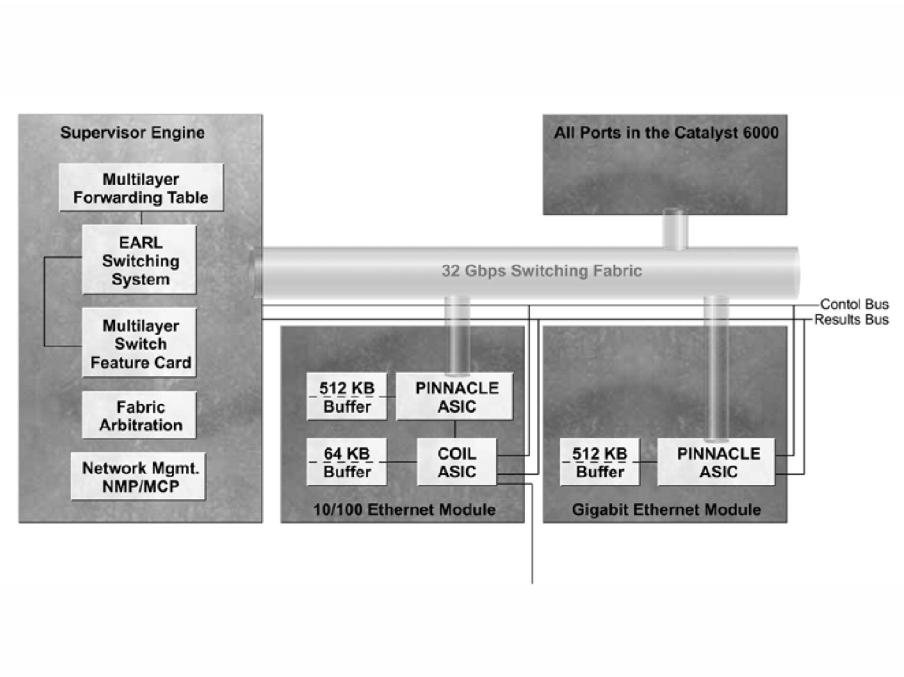
The key difference between the two line cards is that the fabric-enabled cards use a single local switching bus with a bandwidth capacity of 16 Gbps. The fabric-only line cards use two local switching buses, each clocked at 16 Gbps. Both line cards can support distributed forwarding. The DFC daughter card is available as an add-on for the fabric-enabled cards. The fabric-only line cards have the DFC embedded in the system.

A critical component of the local-switch implementation is the connection point between the local system and the SFM. In the Catalyst 6500, this function is handled by an ASIC called Medusa. This ASIC is the interface between the local bus and the crossbar. On the fabric-enabled cards (*not* the fabric-only cards), Medusa also interfaces to the main 32 Gbps switching bus.



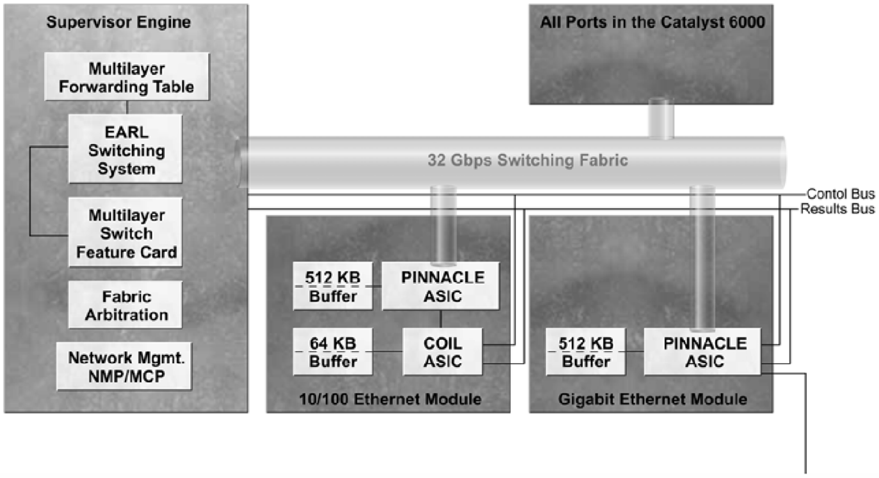
Step 1 Packet enters the switch

When a frame enters the Catalyst 6000 switch, the input port takes the frame in and places it into the input buffer. The input buffer is design to handle store-and-forward checking and hold the frame while PINNACLE arbitrates for access onto the switching bus. There is a local arbiter on each line card that is responsible for allowing each port on each PINNACLE access to the switching bus. This local arbiter signals the central arbiter (on the Supervisor Engine), which is responsible for allowing each local arbiter to allow frames onto the switching bus.



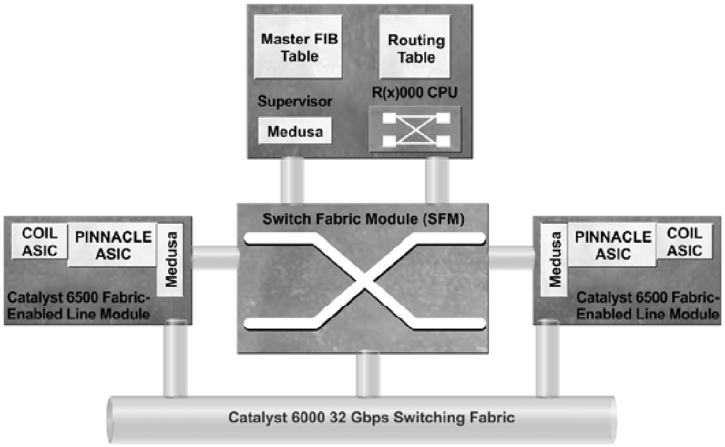
Step 2 Packet sent across switching fabric and lookup

The switching bus on the Catalyst 6000 is a shared medium, meaning that all the ports on the switch see the packet as it transverses the bus. Once the central arbiter has granted access to the switching fabric, the packet is sent across the bus. All ports begin downloading that packet into their transmit buffers. The PFC also looks at the switching bus, sees the frame and initiates a lookup. First, the Layer 2 table is consulted. If the packet is local to the VLAN, a switching decision is finished. If the Layer 2 destination is the router's MAC address, then the Layer 3 Engine examines the packet and determines if a forwarding entry exists in the hardware. If not, the packet is sent to the MSFC. If an entry does exist, the destination VLAN is identified and the Layer 2 lookup table is consulted again, this time to determine the destination MAC address within the VLAN and its associated outbound port.



Step 3 Forwarding the packet

Now that the outgoing interface has been identified, all the ports on the switch that are not the destination port are told, over the Results Bus, to flush their buffers of that packet. The Results Bus also carries to the destination port the MAC re-write information and the appropriate QoS parameters to use (so the packet can be queued correctly on the outgoing port). Once the destination port has received the packet, PINNACLE queues the packet in the correct queue and then uses the SP/WRR scheduler to switch the frame out of memory and to the destination device external to the switch.


Master FIB Table
Routing Table
Supervisor
R(x)000 CPU
Medusa
Switch Fabric Module (SFM)
COIL ASIC
PINNACLE ASIC
Medusa
Catalyst 6500 Fabric-Enabled Line Module
PINNACLE ASIC
COIL ASIC
Medusa
Catalyst 6500 Fabric-Enabled Line Module
Catalyst 6000 32 Gbps Switching Fabric

Step 1a Handling the frame on the local line card

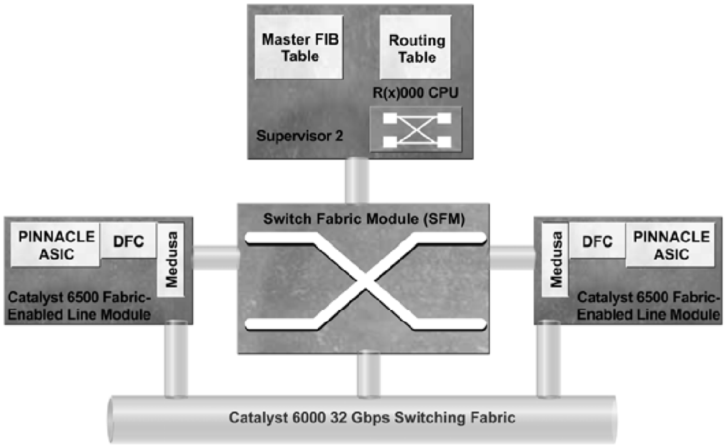
A fabric-enabled system is different from the Catalyst 6000 bus-based system; however, there are remarkable similarities. Each SFM-enabled line card can be thought of conceptually as a Catalyst 6000 on a line card. The switching functionality, therefore, is fairly similar. The packet first enters the switch and is handled by PINNACLE the same way as in the Catalyst 6000 system. Arbitration is required on the local 16 Gbps bus and, while the packet is on the local bus, the header information required for look-up is parsed by the Medusa ASIC, compressed, and sent across the 32 Gbps bus to the Supervisor Engine. All of the Medusa ASICs on all of the other line cards see that frame and download the information.

Step 1b Packet Lookup

The packet is received by the Supervisor Engine 2 and is presented to the PFC-2, which maintains both the Layer 2 and 3 forwarding tables. Similar to when using the PFC-1, a Layer 2 lookup is performed to determine whether a Layer 3 switching decision needs to be made. If a Layer 3 decision is needed, the header information is looked up by the Layer 3 engine, which is utilizing the CEF table. Once a destination VLAN is identified, a second Layer 2 lookup is performed to determine the correct destination MAC address. This information is then sent across the results bus, which again, the Medusa ASICs on all of the other line cards sees. All Medusas, except for the destination one, drop the frame from their buffers.

Step 2 Forwarding the packet

The source line card now knows where the destination is. The line card, via the crossbar interface, adds a tag, which identifies the destination line card and sends the packet into the SFM. The SFM switches the frame to the appropriate destination line card. The information on the Results bus informs the destination line card what the destination port is and what the port of exit is. The packet is queued, for QoS, the same way it is in the Catalyst 6000 system. The WRR scheduler then sends the frame out to the network.

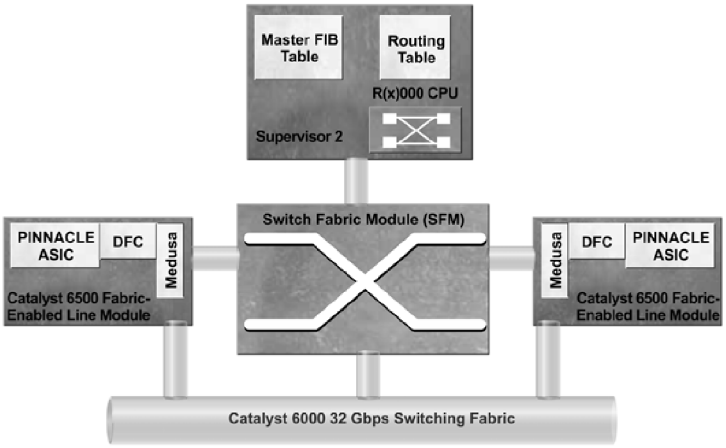

Master FIB Table
Routing Table
R(x)000 CPU
Supervisor 2
Switch Fabric Module (SFM)
PINNACLE ASIC
DFC
Medusa
Catalyst 6500 Fabric-Enabled Line Module
Medusa
DFC
PINNACLE ASIC
Catalyst 6500 Fabric-Enabled Line Module
Catalyst 6000 32 Gbps Switching Fabric

Step 1 Downloading the CEF table

The first step in the distributed switching model of the Catalyst 6500 is to calculate the CEF table and download that table to the line cards. As stated earlier in the course, the CEF table is calculated based on the entries in the routing table. This table is computed centrally at the MSFC-2 on the Supervisor Engine and downloaded to the PFC-2 and DFC (or integrated CEF table). Therefore, the local and central CEF tables contain the same information.

Step 2 Packet lookups

When a packet enters the switch, it is handled by PINNACLE and arbitration is requested for the local switching bus. All ports on the local bus see that frame, including the DFC. The DFC performs a lookup in the local table and identifies whether the destination is local to the line card or across the switching fabric.


Master FIB Table
Routing Table
R(x)000 CPU
Supervisor 2
Switch Fabric Module (SFM)
PINNACLE ASIC
DFC
Medusa
Catalyst 6500 Fabric-Enabled Line Module
Medusa
DFC
PINNACLE ASIC
Catalyst 6500 Fabric-Enabled Line Module
Catalyst 6000 32 Gbps Switching Fabric

Step 3 Switching the packet to the SFM

If the destination is across the SFM, the DFC tells the SFM interface controller (called Medusa) to prepend a tag onto the packet identifying the exit "port" on the SFM.

Step 4 Packet switching in the SFM

Once the packet is received by the SFM, the SFM examines the tag prepended to the packet and makes its own switching decision. Remember that the fabric uses 3x overspeed, so although the inputs to the SFM are 8 Gbps, internal switching takes place at 24 Gbps. The SFM identifies the outgoing port and switches the frame to the Medusa ASIC on the outgoing line card.

Step 5 Switching the frame to the outbound port

The Medusa ASIC on the outgoing port takes the frame out of the SFM and places it onto the switching bus. Since a switching decision has already been made, the local Data bus and Results bus are driven with data at the same time. The Data bus broadcasts the frame, and the Results bus indicates what the destination port is. The information on the Results bus informs the destination line card what the destination port is and what the port of exit is. The packet is queued, for QoS, the same way it is in the Catalyst 6000 system. The WRR scheduler then sends the frame out to the network.

Summary

This lesson accomplished the following:

- Described the specifications of the Catalyst 6500
- Distinguished between Native Mode and Hybrid Mode
- Identified the functions of the MSFC and PFC modules

Described the Catalyst 6500's architecture

Next Steps

After completing this lesson, go to:

- Ethernet Modules

References

For additional information, refer to these resources:

- Catalyst 6500 Family Hardware Documentation - http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/index.htm

Ethernet Modules

Overview

The Catalyst 6500 switch is a LAN Switch. Therefore, the majority of its ports will be Ethernet. This lesson will discuss the different options that are available for Ethernet connectivity on the Catalyst 6500. The Catalyst 6500 supports Ethernet speeds from 10 Mb all the way up to 10 GB.

Importance

The main responsibility of the Catalyst 6500 is Ethernet connectivity. This lesson focuses on the different options available for Ethernet connectivity on the Catalyst 6500.

Objectives

Upon completing this lesson, you will be able to:

- Describe the 10/100 Ethernet modules supported by the Catalyst 6500 series
- Describe the 1 Gigabit Ethernet modules supported by the Catalyst 6500 series
- Describe the 10 Gigabit Ethernet modules supported by the Catalyst 6500 series

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Have a basic knowledge of Ethernet concepts such as Half-Duplex and Full-Duplex.
- Know the differences between Ethernet, Fast Ethernet, and Gigabit Ethernet
- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have the equivalent knowledge

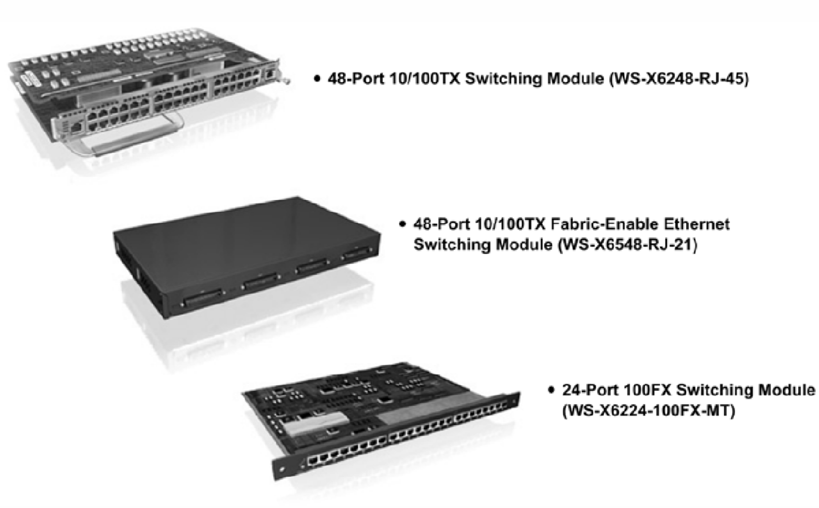
Outline

This lesson includes these sections:

- Overview
- 10/100 Ethernet Modules
- Gigabit Ethernet Modules
- Summary
- Lesson Assessment (Quiz)

10/100 Ethernet Modules

This section details the available 10/100 modules for the Catalyst 6500.

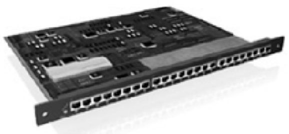

• 48-Port 10/100TX Switching Module (WS-X6248-RJ-45)
• 48-Port 10/100TX Fabric-Enable Ethernet Switching Module (WS-X6548-RJ-21)
• 24-Port 100FX Switching Module (WS-X6224-100FX-MT)

There are a wide variety of options for Ethernet (10 Mbps) and FastEthernet (100 Mbps) connectivity to the Catalyst 6500, via RJ-45, MTRJ, and RJ-21 connectors. Following are descriptions and part numbers for Ethernet and FastEthernet modules supported by the Catalyst 6500:

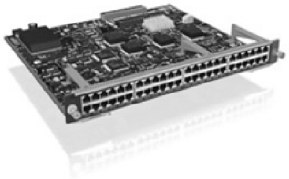
- • 24-Port 10BASE-FL Switching Module (WS-X6024-10FL-MT)
- • 24-Port 100FX Switching Module (WS-X6224-100FX-MT)
- • 48-Port 10/100TX Switching Module (WS-X6248-RJ-45)
- • 48-Port 10/100TX Switching Module (WS-X6248-TEL and WS-X6248A-TEL)
- • 24-Port 100FX Switching Module (WS-X6324-100FX-SM)
- • 24-Port 100FX Switching Module (WS-X6324-100FX-MM)
- • 48-Port 10/100TX Switching Module (WS-X6348-RJ-45)
- • 24-Port 100BASE-FX Fabric-Enabled Ethernet Switching Module (WS-X6524-100FX-MM)
- • 48-Port 10/100TX Fabric-Enabled Ethernet Switching Module (WS-X6548-RJ-21)
- • 48-Port 10/100TX Fabric-Enabled Ethernet Switching Module (WS-X6548-RJ-45)

Gigabit Ethernet Modules

The Catalyst 6500 also supports a few different Gigabit Ethernet Modules, including a 10 Gigabit Module!



8-Port Gigabit Ethernet Switching Module (WS-X6408-GBIC)



16-Port Gigabit Ethernet Switching Module (WS-X6816-GBIC)

16-Port 10/100/1000BASE-Ethernet Switching Module (WS-X6516-GE-TX)

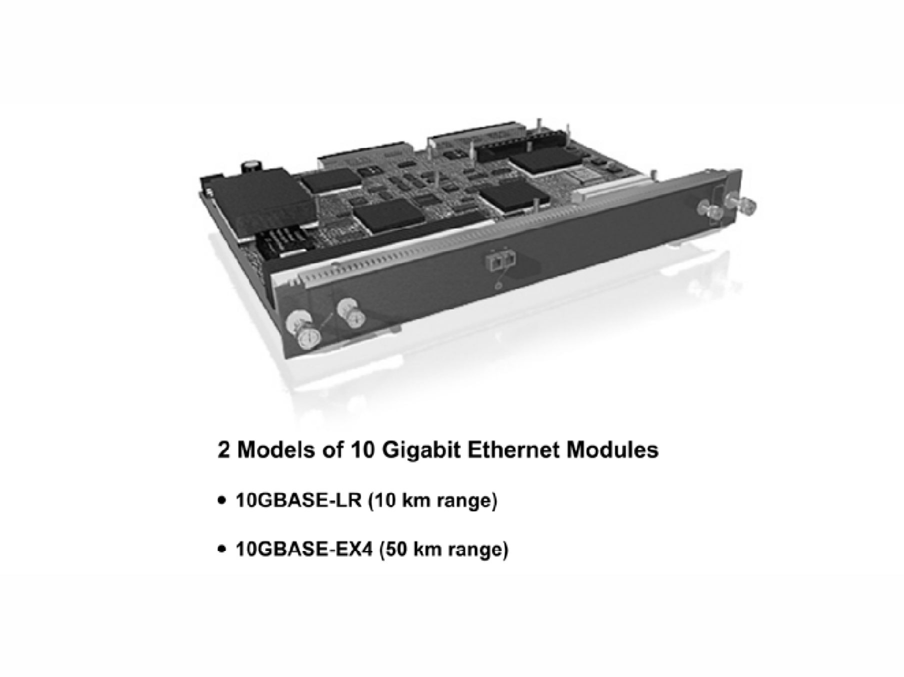
Depending on the model of the Catalyst 6500, a single chassis can accommodate as many as 192 Gigabit Ethernet ports. The following table details the features of the available Gigabit Ethernet modules:

Table 1-3: Gigabit Ethernet Module Features

	Backplane Connection	Forwarding	Number of Transmit Queues/Port	Number of Receive Queues/Port
WS-X6408-GBIC	Bus	Centralized	2	1
WS-X6408A-GBIC	Bus	Centralized	3	2
WS-X6316-GE-TX	Bus	Centralized	3	2
WS-X6416-GBIC	Bus	Centralized	3	2
WS-X6501-10GEX4	Switch fabric and bus	Centralized. Support for distributed forwarding with optional DFC[1]	3	2
WS-X6516-GBIC	Switch fabric and bus	Centralized. Support for distributed forwarding with optional DFC[1]	3	2
WS-X6516-GE-TX	Switch fabric and bus	Centralized. Support for distributed forwarding with optional DFC[1]	3	2
WS-X6816-GBIC	Switch fabric (dual channel)	Distributed forwarding with integrated DFC[1]	3	2

[1] DFC = Distributed Forwarding Card

10 Gigabit Ethernet Modules



2 Models of 10 Gigabit Ethernet Modules

- 10GBASE-LR (10 km range)
- 10GBASE-EX4 (50 km range)

Cisco Catalyst 6500 10GBASE-LR Serial 1310nm 10 Gigabit Ethernet Module (WS-X6502-10GE, WS-G6488)

The Cisco Catalyst 6500 Series and Cisco 7600 Series support two 10 Gigabit Ethernet modules:

- 10GBASE-LR Serial 1310nm Long Haul 10 Gigabit Ethernet Module (up to 10km over single-mode fiber)
- 10GBASE-EX4 Metro 1550nm Extended Reach 10 Gigabit Ethernet Module (10-50 km over single-mode fiber)

The Cisco Catalyst 6500 10GBASE-LR Serial 1310nm 10 Gigabit Ethernet module is the industry's first shipping 10 Gigabit Ethernet module that is based on the current IEEE 802.3ae draft. It can interoperate with any module that supports the same Serial 1310nm specification in the current IEEE 802.3ae 10 Gigabit Ethernet draft. Supporting distances up to 10 km over single-mode fiber, the Catalyst 6500 Serial 1310nm 10 Gigabit Ethernet module is ideal for building simple high-bandwidth inter-building connections inside a campus, between points-of-presence (POPs), or aggregating Gigabit Ethernet in the core. The base board of the Serial 1310nm 10 Gigabit Ethernet module can also be used to support other optical media in the future, giving customers of the Catalyst 6500 Series and Cisco 7600 Series yet another level of investment protection.

The Cisco Catalyst 6500 10GBASE-EX4 Metro 10 Gigabit Ethernet module supports an extended reach of 10 to 50 km over single-mode fiber. It is ideal for building inter-campus connections and MANs. By supporting high-bandwidth connections over extended distances, the Catalyst 6500 10GBASE-EX4 Metro 10 Gigabit Ethernet solution enables new applications such as server-less buildings, data center remote mirroring, and disaster recovery, which are key storage networking applications.

Summary

This lesson accomplished the following:

- Described the 10/100 Ethernet modules supported by the Catalyst 6500 series

- Described the 1 Gigabit Ethernet modules supported by the Catalyst 6500 series

- Described the 10 Gigabit Ethernet modules supported by the Catalyst 6500 series

Next Steps

After completing this lesson, go to:

- WAN Options

References

For additional information, refer to these resources:

- Catalyst 6500 Family Module Installation Guide – Product Overview: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/mod_inst/01overvw.htm

WAN Options

Overview

One of the features that truly differentiates the Catalyst 6500 from a traditional LAN switch is its ability to support WAN services. The Catalyst 6500 can terminate WAN services such as T1/E1, T3/E3, and ATM; including ATM LANE and MPOA. This support of WAN services makes the Catalyst 6500 one of the most versatile switches on the market today.

Importance

The support for WAN Services on the Catalyst 6500 is critical to Cisco's move towards unified network devices that support multiple functions, such as Layer 3 and Layer 4 switches.

Objectives

Upon completing this lesson, you will be able to:

- Describe the FlexWAN module
- Describe the ATM LANE/MPOA Module

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Have a basic knowledge of WAN Technologies such as T1, Frame Relay, and ATM

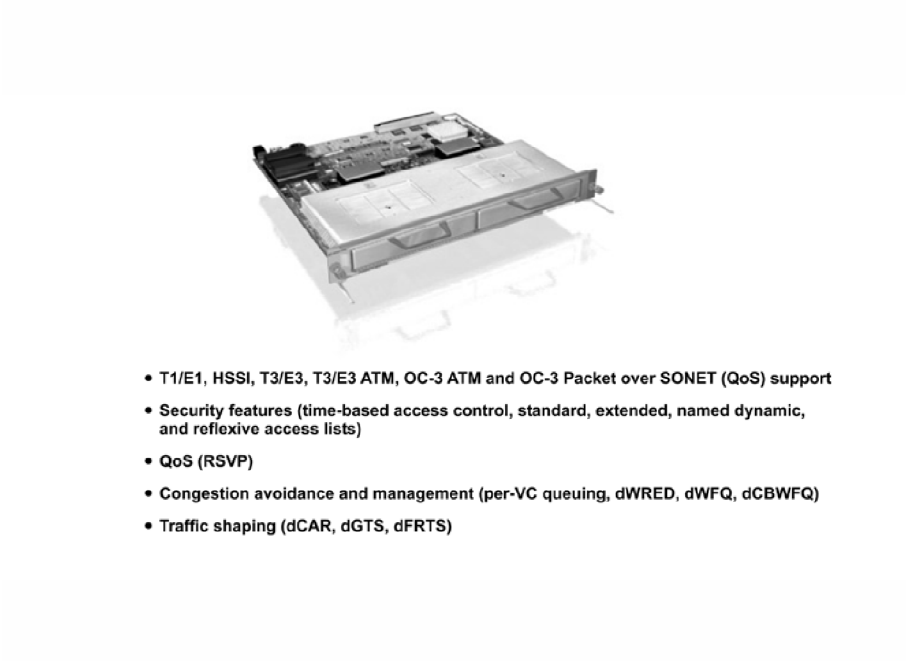
Outline

This lesson includes these sections:

- Overview
- FlexWan Module
- ATM LANE/MPOA Module
- Summary

FlexWan Module

The FlexWan Module allows you to terminate WAN circuits on your Catalyst 6500.



- T1/E1, HSSI, T3/E3, T3/E3 ATM, OC-3 ATM and OC-3 Packet over SONET (QoS) support
- Security features (time-based access control, standard, extended, named dynamic, and reflexive access lists)
- QoS (RSVP)
- Congestion avoidance and management (per-VC queuing, dWRED, dWFQ, dCBWFQ)
- Traffic shaping (dCAR, dGTS, dFRTS)

The FlexWAN module accepts up to two Cisco 7200/7500 series WAN port adapters. As shown later in this course, the FlexWAN module is configured from the MSFC and can provide WAN connectivity through circuits such as T1s, E1s, T3s, E3s, and OC-3s. However, not all 7200/7500 port adapters are supported in the FlexWAN module. Some of the unsupported media types include Token Ring, FDDI, and Channel Port Adapters.

ATM LANE/MPOA Module

The ATM LANE/MPOA Module can be added to the Catalyst 6500 to support ATM LANE and MPOA.

ATM Support for:

- **LANE (LAN Emulation)**
- **MPOA (Multiprotocol over ATM)**

Support for both ATM Forum standard LAN Emulation (LANE) and Multiprotocol over ATM (MPOA) enable the Catalyst 6500 Family of switches to connect to ATM backbones, providing scalable, intelligent switching with value-added high availability services. Service Provider and Enterprise customers alike can take advantage of support for RFC 1483 Permanent Virtual Circuits (PVC's) with traffic shaping to extend transparent LAN services across wide area ATM networks. In addition, future support for end-to-end Quality of Service (QoS) will enable enterprise voice solutions and services to seamlessly coexist between Ethernet and ATM backbones.

LANE

The Catalyst 6000 Family enhances ATM LANE network performance by delivering support for the entire range of LANE v1.0 servers. By using these features, network administrators can easily distribute LANE services across each Catalyst ATM dual-PHY uplink module. For example, this feature allows scalable distribution of bandwidth-intensive LANE BUS services for multimedia applications. Operating as a standalone LANE v1.0 BUS, each Catalyst 6000 Family OC-12 dual-PHY uplink module can forward more than 400,000 pps.

MPOA

As an integral part of the Cisco intelligent switching strategy, MPOA provides a standards-based Layer 3 switching solution for ATM networks. An MPOA network comprises the following: an MPOA server (MPS) and an MPOA client (MPC). Routers such as the Cisco 7500 and Cisco 7200 support the MPS function that supplies all of the Layer 3 forwarding information used by MPCs. The Catalyst 6000 Family OC-12 LANE/MPOA modules support

MPC through dedicated onboard hardware. With a hardware implementation of MPOA, Cisco provides unmatched performance for MPOA and delivers distributed, high-speed Layer 3 switching throughout the ATM backbone.

The ATM OC-12 modules are available for both multimode and singlemode fiber optic cables, as shown in the following table:

Table 1-4: ATM LANE/MPOA Modules

Part Number	Description
WS-X6101-OC12-MMF	OC-12 ATM LANE/MPOA Uplink for the Catalyst 6000 Family switches, multimode fiber, SC
WS-X6101-OC12-SMF	OC-12 ATM LANE/MPOA Uplink for the Catalyst 6000 Family switches, singlemode fiber, SC

Summary

This lesson accomplished the following:

- Described the FlexWAN module
- Described ATM LANE/MPOA Module

Next Steps

After completing this lesson, go to:

- Voice Services Options

References

For additional information, refer to these resources:

- FlexWan Module - http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/mod_inst/01overvw.htm#xtocid2525835
- ATM Modules - http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/mod_inst/01overvw.htm#xtocid2525831

Voice Services Options

Objectives

Upon completing this lesson, you will be able to:

- Describe the Catalyst 6500’s Inline Power Ethernet module
- Explain the advantages of Auxiliary VLANs
- Describe the Catalyst 6500’s 24-port FXS module
- Describe the Catalyst 6500’s 6608 Voice T1 and Services module

Inline Power FastEthernet Module (WS-X6348-RJ45V)

Some Ethernet modules support a feature called inline power, which is 48-volt DC power provided over standard Category 5 unshielded twisted-pair (UTP) cable up to 100 meters. Instead of using wall power, terminal devices such as IP telephones can utilize power provided from the Catalyst 6500 switch. This capability gives the network administrator centralized power control, which translates into greater network availability. By deploying the Catalyst 6500 family of switches with uninterruptible power supply (UPS) systems in secured wiring closets, network administrators can ensure that building power outages will not affect network telephony connections.

The inline power feature will work over customers' existing Category 5 UTP installations. The Catalyst inline power implementation passes the required domestic and international safety regulations and compliance measures. These modules are fully compliant with the 802.3 standard when no inline power is supplied. The 802.3 standard does not include specifications for providing power over Ethernet (PoE); this omission will be amended by the 802.3af task force that is currently under way in the IEEE. Cisco is committed to standards-based operation and will support the IEEE in its efforts to add Ethernet power specifications to the 802.3 Ethernet standard.

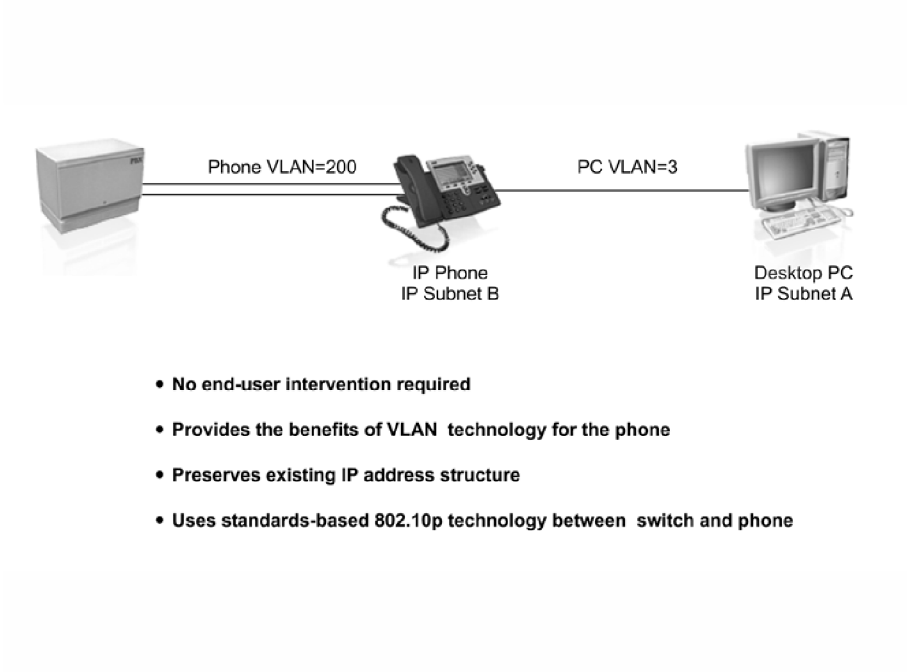
Phone Discovery

The Cisco phone discovery feature eases the network management burden by automating the inline power feature. With phone discovery, the Catalyst switch automatically detects the presence of an IP phone and supplies inline power. This means that network administrators can maintain centralized control without the need to manually enable each port to supply inline power. The phone discovery mechanism is intelligent enough to differentiate between an IP phone and a network interface card, and will not supply inline power to a network interface card or other device not designed to use inline power. Therefore, network administrators can depend upon automatic and centralized control of inline power that is safe to deploy and maintain.

To support the new demand for phone power provided with the inline power feature, Cisco has developed a 2500-watt power supply for the Catalyst 6500 family. This power supply has been designed to work in Catalyst 6500 family switches that will be loaded with inline power line cards and IP phones. For fault tolerance, two power supplies can be deployed in a single chassis to guard against a single power supply failure.

The 5.5(1) release of Catalyst 6500 software also supports new power management features. Network administrators can query the switch for available power resources. Furthermore, users can configure two power supplies to operate in non-redundant mode, increasing the overall power provided within a system. For example, a user can configure two 1300-watt power supplies to act together as one 2600-watt supply, enabling a system to support more IP telephones.

Auxiliary VLANs

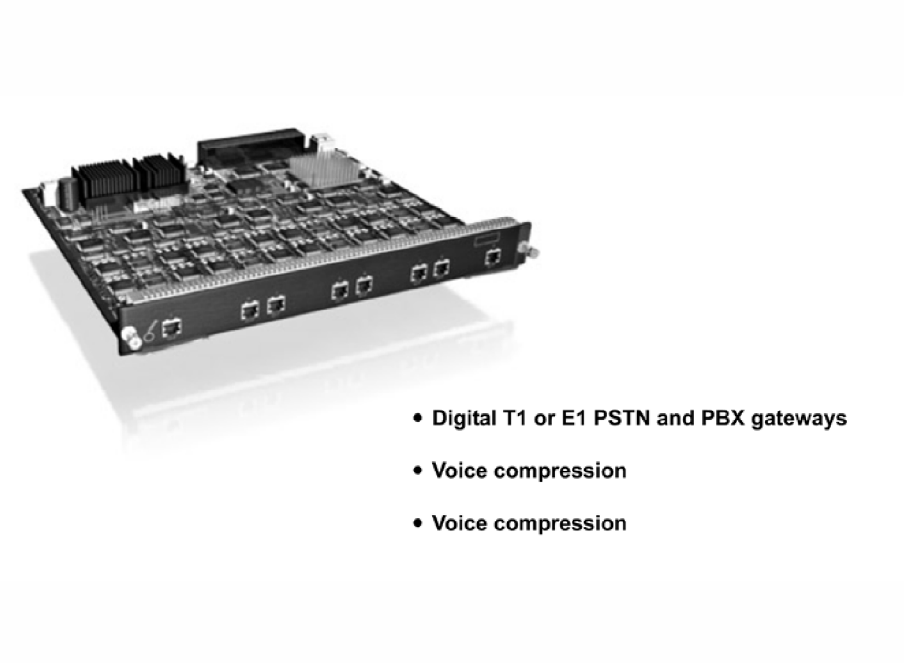


- **No end-user intervention required**
- **Provides the benefits of VLAN technology for the phone**
- **Preserves existing IP address structure**
- **Uses standards-based 802.10p technology between switch and phone**

The auxiliary VLAN feature places the phones into their own VLANs without any end-user intervention. Furthermore, these VLAN assignments can be seamlessly maintained, even if the phone is moved to a new location. The user simply plugs the phone into the switch, and the switch will provide the phone with the necessary VLAN information. By placing phones into their own VLANs, network administrators gain the advantages of network segmentation and control.

Furthermore, network administrators can preserve their existing IP topology for the data end stations. IP phones can be easily assigned to different IP subnets using standards-based Dynamic Host Configuration Protocol (DHCP) operation. With the phones in their own IP subnets and VLANs, network administrators can more easily identify and troubleshoot network problems. Additionally, network administrators can create and enforce QoS or security policies. With the auxiliary VLAN feature, Cisco enables network administrators to gain all the advantages of physical infrastructure convergence while maintaining separate logical topologies for voice and data terminals, creating the most effective way to manage a multiservice network.

Voice T1 and Services Module (WS-X6608-T1)



- Digital T1 or E1 PSTN and PBX gateways
- Voice compression
- Voice compression

The voice T1 and services module provides 8 T1 ports (192 channels or DS0 voice trunks) for connections to the Public Switched Telephone Network (PSTN) or PBX. As a gateway to the PSTN or legacy PBX, this module provides voice packetization services for delivery to/from the IP network.

Furthermore, the voice T1 and services module can be used as a shared network resource. Users have the flexibility to use each port for T1 connections or as a network resource that provides voice services such as transcoding (converting from one CODEC to another) or conferencing. For example, a network administrator can configure two ports as T1 connections to the PSTN. The remaining six ports can be used to provide voice-conferencing services.

FXS Analog Interface Module (WS-X6624-FXS)

Insert Slide here.

The Catalyst 6000 FXS Analog Interface Module provides 24 foreign exchange station (FXS) ports for analog phones, conference room speakerphones, and fax machines. The FXS module provides legacy analog devices with connectivity into the IP network, enabling them to utilize the IP network infrastructure for toll-bypass applications and to communicate with devices such as IP phones and H.323 end stations. This module also supports fax relay, which enables compressed fax transmission over the IP WAN, preserving valuable WAN bandwidth for other data applications.

Summary

This lesson accomplished the following:

- Described the Catalyst 6500’s Inline Power Ethernet module
- Explained the advantages of Auxiliary VLANs
- Described the Catalyst 6500’s 24-port FXS module
- Described the Catalyst 6500’s 6608 Voice T1 and Service module

Next Steps

After completing this lesson, go to:

- Web Hardware

References

For additional information, refer to these resources:

- FlexWan Module - http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/mod_inst/01overvw.htm#xtocid2525835
- ATM Modules - http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/mod_inst/01overvw.htm#xtocid2525831

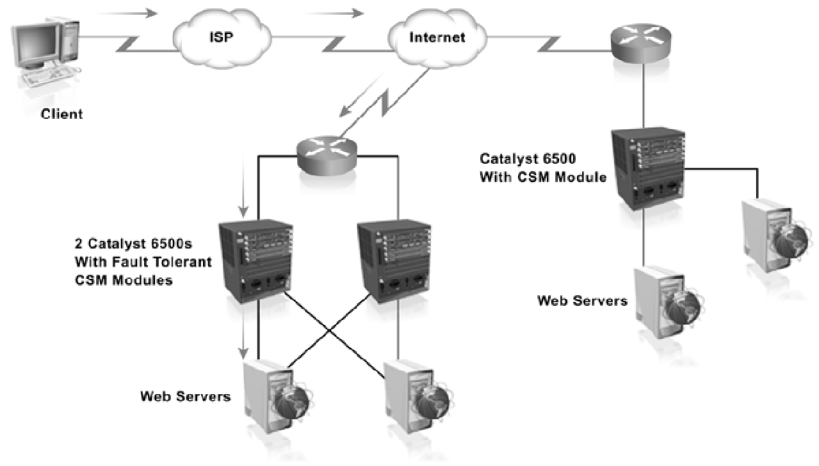
Web Hardware

Objectives

Upon completing this lesson, you will be able to:

- Describe the Content Switching Module

Content Switching Module (WS-X6066-SLB-APC)



The Cisco Content Switching Module (CSM) is a Catalyst 6500 line card that balances client traffic to farms of servers, firewalls, SSL devices, or VPN termination devices. Fault tolerant CSM configurations maintain full state information and provide true hitless failover required for mission-critical functions.

The CSM provides the following key benefits:

- Performance—Establishes up to 200,000 Layer 4 connections per second and provides high-speed content switching, while maintaining 1 million concurrent connections.
- Price/performance value for large data centers and ISPs—Features a low connection cost and occupies a small footprint. The CSM slides into a slot in a new or existing Catalyst 6500 and enables all ports in the Catalyst 6500 for layer 4 through layer 7 content switching. Multiple CSMs can be installed in the same Catalyst 6500.
- Ease of configuration—Uses the same Cisco IOS Command Line Interface (CLI) that is used to configure the Catalyst 6500 Switch.

Summary

This lesson accomplished the following:

- Described the Content Switching Module

Next Steps

After completing this lesson, go to:

- Security Hardware

References

For additional information, refer to these resources:

- FlexWan Module - http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/mod_inst/01overvw.htm#xtocid2525835

- ATM Modules - http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/mod_inst/01overvw.htm#xtocid2525831

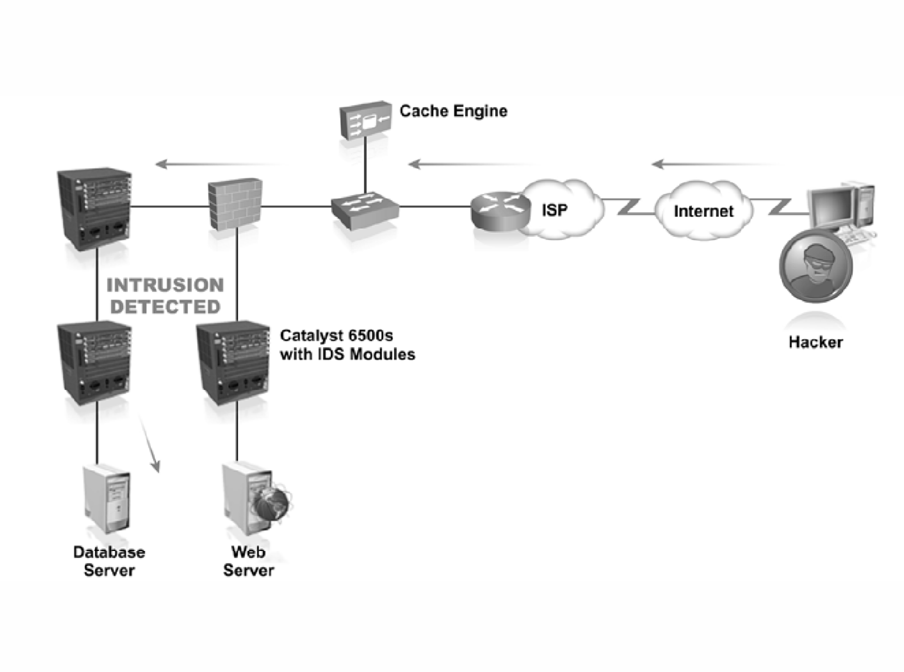
Security Hardware

Objectives

Upon completing this lesson, you will be able to:

- Describe the Intrusion Detection System (IDS) Module

Intrusion Detection System (IDS) Module (WS-X6381-IDS)



The IDS module allows security and network administrators to monitor network traffic right off the switch backplane rather than using external IDS sensors connected to a switch's destination SPAN port. This allows more granular access to the network traffic and overcomes some of the limitations that external IDS sensors connected to destination SPAN ports have.

The IDS module detects unauthorized activity traversing the network, such as attacks by hackers, and will send alarms to a management console with the details of the detected event. The security or network administrator specifies the network traffic that should be inspected by the IDS module using the Catalyst OS VLAN access control list (VACL) capture feature or SPAN functionality, allowing for very granular traffic monitoring. In addition, the IDS module can be managed and monitored by the same management console as the Cisco Secure IDS sensors, allowing customers to deploy both appliance sensors and the IDS module to monitor critical subnets throughout their enterprise network. The Catalyst 6500 IDS Module, which has the widest range of attack recognition, provides the best real time intrusion detection solution available in the industry today.

Due to the type and volume of traffic at the network core, the IDS module is most effective in the distribution and access layers of the network.

Summary

Insert Slide here.

Review Questions

Q1) Which of the following is true of the “hybrid” mode?

A) There is a single OS controlling both switching and routing functions

B) IOS commands cannot be used

C) Only “set-based” commands can be used

D) The “Cat OS” is used for switch configuration, and the IOS is used for MSFC configuration

Answer: D

Q2) Which of the following is NOT made possible by the PFC?

A) Accelerated SLB

B) IGMP Snooping

C) CGMP

D) QoS ACLs

Answer: C

Q3) What is the maximum distance for a 10 Gbps interface on the Catalyst 6500?

A) 100m

B) 200m

C) 2 km

D) 10 km

E) 50 km

F) 100 km

Answer: F

Which of the following port adapter types is supported by the FlexWAN module?

G) Token Ring Adapters

H) E3 Adapters

I) FDDI Adapters

J) Channel Port Adapters

Answer: B

Q4) How many Layer 4 sessions can the CSM (Content Switching Module) establish per second?

A) 200,000

B) 400,000

C) 2,000,000

D) 4,000,000

Answer: A

Layer 2 Services

Overview

This module details the configuration of the Catalyst 6500's Layer 2 Services.

Upon completing this module, you will be able to:

- Configure an IP address for a Catalyst 6500's management interface
- Enable the High Availability feature
- Create an EtherChannel (in both Hybrid and Native modes)
- Configure VLANs and trunks (in both Hybrid and Native modes)
- Optimize Spanning Tree Performance using PortFast, UplinkFast, and BackboneFast

Outline

The module contains these lessons:

- Switching Overview and Basic Switch Configuration
- Layer 2 Redundancy
- EtherChannel
- Spanning Tree Protocol
- VLANs and Trunking

Switching Overview and Basic Switch Configuration

Overview

When a Catalyst 6500 is first deployed, certain decisions, such as the VLAN of the management interface, must be made. Additionally, Ethernet ports should be configured with the correct duplex and speed settings. This lesson presents a five-step process for initial switch configuration.

Importance

The initial configuration settings of the Catalyst 6500 are critical to a successful rollout.

Objectives

Upon completing this lesson, you will be able to:

- Explain the basic functions of Layer 2 switches
- Configure the Catalyst 6500's management interface
- Configure basic system settings

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have equivalent knowledge

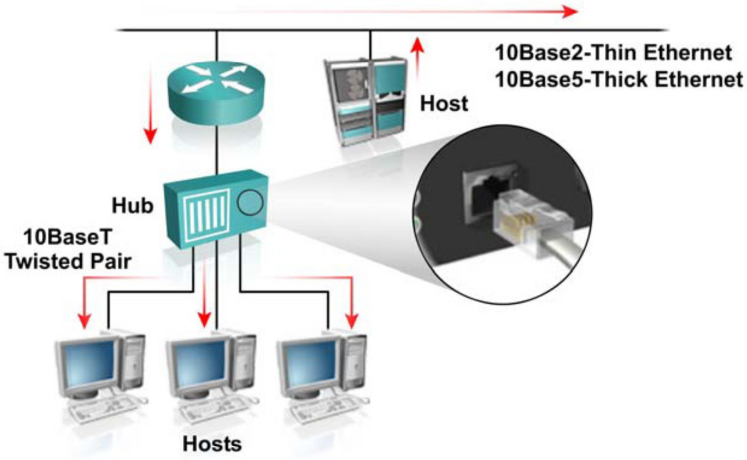
Outline

This lesson includes these sections:

- Overview
- Understanding How Ethernet Works
- Basic Software Configuration
- Summary

Understanding How Ethernet Works

This section reviews the principles of Ethernet hub and switch operations.



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

The Catalyst 6500 family switches solve 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. Since each Ethernet port on the switch represents a separate Ethernet segment, servers in a properly configured switched environment achieve full access to the bandwidth available on their segment.

Collisions are a major bottleneck in Ethernet networks. An effective solution to collisions is full-duplex communication. Full-duplex is an option for any 10- or 100-Mbps port on a Catalyst 6500 family switch. Normally, Ethernet operates in half-duplex mode, which means that stations can either receive only or transmit only during a certain timeframe. In full-duplex mode, a station can transmit and receive at the same time. When packets can flow in both directions simultaneously, effective Ethernet bandwidth doubles to 20 Mbps for Ethernet ports and 200 Mbps for Fast Ethernet ports. Gigabit Ethernet ports on Catalyst 6500 family switches are always in full-duplex mode, thus providing 2 Gbps of effective bandwidth.

Each Ethernet port on a Catalyst 6500 family switch can connect to a single workstation or server, or to another switch or hub through which multiple workstations or servers connect to the network.

Ports on a typical Ethernet hub all connect to a common backplane within the hub, and the bandwidth of the hub's uplink is shared by all devices attached to the hub. If two stations establish a session that uses a significant amount of bandwidth, the network performance of all other stations attached to the hub is degraded.

To reduce degradation, the switch treats each of its ports as individual segments. When stations on different ports need to communicate, the switch forwards frames from one port to the other at wire speed to ensure that each session receives the full amount of bandwidth available, based on the media being used.

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

Basic Software Configuration

This section covers the five-step process for performing an initial configuration on a Catalyst 6500.

Basic software configuration

1. Prepare to configure the switch

2. Establish a console port connection

3. Set the switch IP address

4. Configure the Ethernet ports

5. Configure global system settings

Very little configuration is required to establish basic connectivity to your Catalyst 6500 family switch. This section describes the basic tasks needed to get your switch up and running:

Step 1 Preparing to Configure the Switch—Gather the information you need to configure the switch.

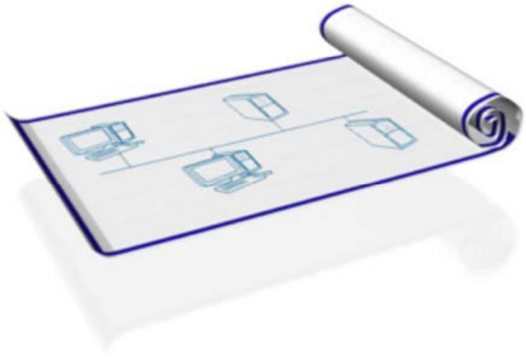
Step 2 Establishing a Console Port Connection—Connect to the switch via the console port to access the switch's command-line interface (CLI).

Step 3 Setting the Switch's IP Address—Assign an IP address, subnet mask, and default gateway to the switch's management interface.

Step 4 Configuring Ethernet Ports—Make sure the Ethernet, Fast Ethernet, and Gigabit Ethernet ports are properly configured to communicate with connected devices.

Step 5 Configuring Global System Settings—Configure global settings such as system name, date and time, prompt, and passwords.

Basic Software Configuration (Step 1)



- **Obtain a network map or diagram**
- **Obtain switch IP address and mask**

Preparing to Configure the Switch

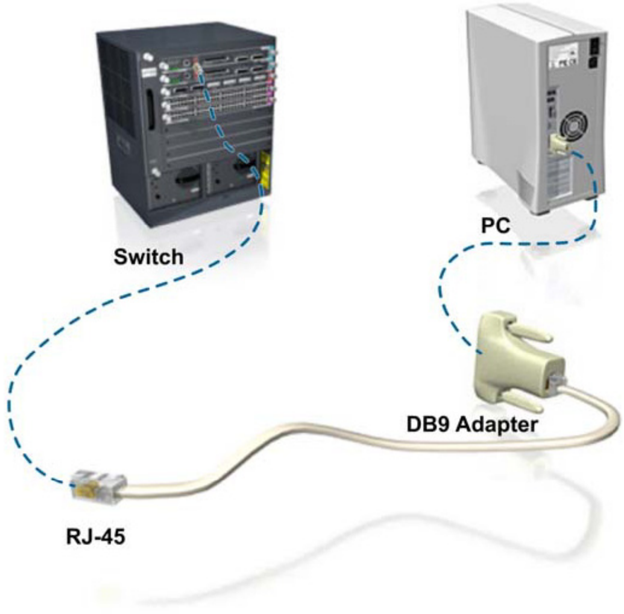
Before you configure the switch, make sure the switch, modules, and power supplies are installed and cabled as described in the *Catalyst 6000 Family Installation Guide* and *Catalyst 6000 Family Module Installation Guide* publications.

Before you begin configuring the switch, you should collect the following information:

A map or diagram of your network topology showing how the Catalyst 6500 switch will be used in the network.

The IP address and netmask for the switch. You assign this address to the in-band (sc0) interface on the switch *for remote management purposes, such as Telnet and SNMP*.

Basic Software Configuration (Step 2)



Establishing a Console Port Connection

Connecting a terminal to the supervisor engine's console port allows you to access the switch's CLI before the switch is configured and connected to the network.

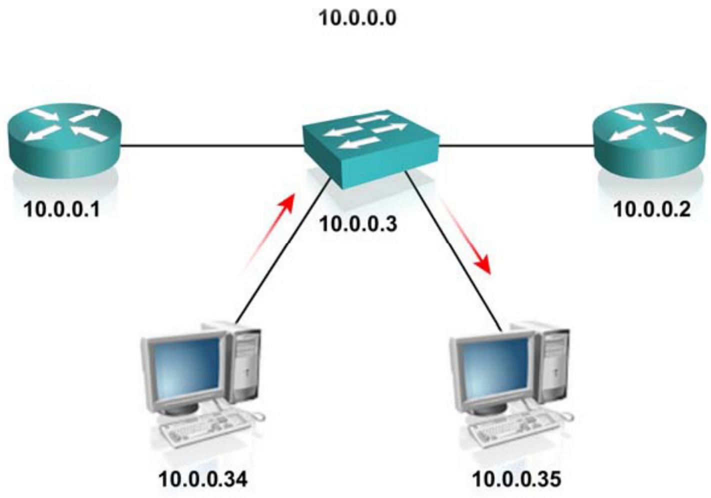
You must enter privileged mode to perform most of the tasks described in this course. Enter the enable command to access privileged mode.

To connect to the switch via the console port and enter privileged mode, perform these steps:

Table 2-1: Entering Privileged Mode

	Task	Command
Step 1	Make sure the terminal connected to the console port is configured as follows: 9600 baud, 8 data bits, no parity, 2 stop bits.	-
Step 2	Power up the switch. Output from the bootup script appears on the terminal screen.	-
Step 3	At the Enter Password prompt, press Return.	-
Step 4	Enter privileged mode.	`enable`
Step 5	At the Enter Password prompt, press Return.	-

Basic Software Configuration (Step 3)



Setting the Switch's IP Address

Before you can Telnet to your Catalyst 6500 switch, you need to assign the switch an IP address, subnet mask, and a default gateway to allow it to communicate with remote devices via TCP/IP.

When configuring the switch for the first time, assign the in-band (sc0) interface to the default VLAN, VLAN 1. After you have configured additional VLANs, you can assign the interface to any VLAN you would like. Make sure the IP address you specified belongs to the subnet associated with that VLAN. To assign an IP address, subnet mask, and default gateway, perform these steps in privileged mode:

Table 2-2: Assigning an IP Address to the Management Interface

	Task	Command
Step 1	Assign an IP address and subnet mask to the switch.	**set interface sc0** *ip_addr/netmask* or *ip_addr subnet mask*
Step 2	(Optional) Assign the switch's sc0 interface to a VLAN. (If you do not specify a VLAN, VLAN 1 is used.)	**Set interface sc0** *vlan_num*
Step 3	Assign a default gateway to the switch. The primary keyword is used to specify the primary gateway if multiple default gateways are defined for redundacy purposes.	**set ip route default** *gateway* **primary**
Step 4	Verify the in-band (sc0) interface configuration.	**show interface**
Step 5	Verify the default gateway assignment.	**show ip route**
Step 6	Test connectivity to a remote host on the network.	**ping [-s]** *host [packet_size]* [*packet_count*]

This example shows how to assign an IP address and default gateway to the switch, verify the switch's sc0 interface's configuration, and check connectivity to a remote host:

```
Console> (enable) set interface sc0 10.1.1.50/8
Interface sc0 IP address and netmask set.
Console> (enable) set interface sc0 100
Interface sc0 vlan set.
Console> (enable) set ip route default 10.1.1.1
Route added.
Console> (enable) show interface
sl0: flags=51<UP,POINTOPOINT,RUNNING>
        slip 0.0.0.0 dest 0.0.0.0
sc0: flags=63<UP,BROADCAST,RUNNING>
        vlan 100 inet 10.1.1.50 netmask 255.0.0.0 broadcast 10.255.255.255
Console> (enable) show ip route
Fragmentation   Redirect   Unreachable
-------------   --------   -----------
enabled         enabled    enabled
Destination             Gateway                 Flags   Use          Interface
----------------------- ----------------------- ------  ----------   ---------
default                 10.1.1.1                UG               0   sc0
10.0.0.0                10.1.1.50               U                0   sc0
default                 default                 UH               0   sl0
Console> (enable) ping 10.1.1.100
10.1.1.100 is alive
```

```
Console> (enable)
```

Basic Software Configuration (Step 4)

Configuring Ethernet Ports

Ethernet, Fast Ethernet, and Gigabit Ethernet ports on both ends of a link must use the same port speed and duplex setting. Ethernet and Fast Ethernet ports can autonegotiate duplex mode. In addition, 10/100-Mbps Fast Ethernet ports can autonegotiate port speed. Gigabit Ethernet ports are always full duplex.

Setting the Port Name

You can configure port names on Ethernet, Fast Ethernet, and Gigabit Ethernet ports to simplify switch administration.

To assign a name to a port, you must be in privileged mode:

Table 2-3: Specify a Port Name

	Task	Command
Step 1	Set a port name.	**set port name** *mod_num/port_num* [*name_string*]
Step 2	Verify the port name configuration.	**show port** [*mod_num*[/*port_num*]]

This example shows how to assign a name to ports 1/1 and 1/2 and how to verify that the port names are configured correctly:

```
Console> (enable) set port name 1/1 Router Connection
Port 1/1 name set.
Console> (enable) set port name 1/2 Server Link
Port 1/2 name set.
Console> (enable) show port 1
Port  Name               Status     Vlan         Duplex Speed Type
----- ------------------ ---------- ----------   ------ ----- ------------
 1/1  Router Connection  connected  trunk          full  1000 1000BaseSX
 1/2  Server Link        connected  trunk          full  1000 1000BaseSX

<...output truncated...>

Last-Time-Cleared
--------------------------
Wed Jun 16 1999, 16:25:57
Console> (enable)
```

Setting the Port Speed

You can configure the port speed on 10/100-Mbps Ethernet switching modules. Use the auto keyword to autonegotiate the port's speed and duplex mode with the neighboring device.

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

To manually set the port speed for a 10/100-Mbps port, follow these steps from privileged mode:

Table 2-4: Setting the Port Speed

	Task	Command
Step 1	Set the port speed of a 10/100-Mbps Fast Ethernet port.	**set port speed** *mod num/port num* {10 | 100 | auto}
Step 2	Verify that the speed of the port is configured correctly.	**show port** [*mod_num*[/*port_num*]]

This example shows how to set the port speed to 100 Mbps on port 2/2:

```
Console> (enable) set port speed 2/2 100
Port 2/2 speed set to 100 Mbps.
Console> (enable)
```

This example shows how to make port 2/1 autonegotiate speed and duplex with a neighboring device:

```
Console> (enable) set port speed 2/1 auto
Port 2/1 speed set to auto-sensing mode.
Console> (enable)
```

Setting the Port Duplex Mode

You can set the port duplex mode to either half- or full-duplex for Ethernet and Fast Ethernet ports.

Note Gigabit Ethernet is full duplex only. You cannot change the duplex mode on Gigabit Ethernet ports.

If the port speed is set to auto on a 10/100-Mbps Ethernet port, both speed and duplex will be autonegotiated. You cannot change the duplex mode of ports that are set for autonegotiation.

To set the duplex mode of a port, perform this task in privileged mode:

Table 2-5: Setting the Port Duplex

	Task	Command
Step 1	Set the duplex mode of a port.	**`set port duplex`** *`mod num/port num`* `{half	full}`
Step 2	Verify that the duplex mode of the port is configured correctly.	**`show port`** `[`*`mod_num`*`[/`*`port_num`*`]]`

This example shows how to set the duplex mode to half duplex on port 2/1:

```
Console> (enable) set port duplex 2/1 half
Port 2/1 set to half-duplex.
Console> (enable)
```

Configuring IEEE 802.3Z Flow Control

Gigabit Ethernet ports on the Catalyst 6000 family switches use flow control to inhibit the transmission of packets to the port for a period of time; other Ethernet ports use flow control to respond to flow-control requests.

If a Gigabit Ethernet port’s receive buffer becomes full, the port transmits a "pause" packet that tells remote devices to delay the sending of more packets for a specified period of time. All Ethernet ports (1000 Mbps, 100 Mbps, and 10 Mbps) can receive and act upon "pause" packets from other devices.

Enter the set portflow control command to configure flow control on ports:

Table 2-6: Configuring Flow Control

Keywords	Function
receive on	The port uses flow control dictated by the neighboring device.
receive desired	The port uses flow control if the neighboring device uses it and does not use flow control if the neighbor device does not use it.
receive off	The port does not use flow control, regardless of whether flow control is requested by the neighboring device.
send on[1]	The port sends flow-control frames to the neighboring device.
send desired[1]	The port sends flow-control frames to the port if the neighboring device asks to use flow control.
send off[1]	The port does not send flow-control frames to the neighboring device.

[1] Supported only on Gigabit Ethernet ports.

To configure flow control, perform this task in privileged mode:

Table 2-7: Disabling Flow Control

	Task	Command
Step 1	Set the flow-control parameters.	**set port flowcontrol** *mod_num/port_num* **{receive** | **send}** **{off** | **on** | **desired}**
Step 2	Verify the flow-control configuration.	**show port flowcontrol**

This example shows how to turn transmit and receive flow control on, and how to verify the flow-control configuration:

```
Console> (enable) set port flowcontrol 3/1 send on
Port 3/1 will send flowcontrol to far end.
Console> (enable) set port flowcontrol 3/1 receive on
Port 3/1 will require far end to send flow control
Console> (enable) show port flowcontrol
Port  Send-Flowcontrol  Receive-Flowcntl  RxPause  TxPause
      Admin   Oper      Admin   Oper
----- ----------------  ----------------  -------  -------
 3/1  on      disagree  on      disagree  0        0
 3/2  off     off       off     off       0        0
 3/3  desired on        desired off       10       10
Console> (enable)
```

Enabling and Disabling Port Negotiation

To enable port negotiation, perform this task in privileged mode:

Table 2-8: Enabling Port Negotiation

	Command	Description
Step 1	Enable port negotiation.	**set port negotiation** *mod_num/port_num* **enable**
Step 2	Verify the port negotiation configuration.	**show port negotiation** [*mod_num/port_num*]

This example shows how to enable port negotiation and verify the configuration:

```
Console> (enable) set port negotiation 2/1 enable
Port 2/1 negotiation enabled
Console> (enable) show port negotiation 2/1
Port   Link Negotiation
-----  ----------------
 2/1   enabled
Console> (enable)
```

To disable port negotiation, perform this task in privileged mode:

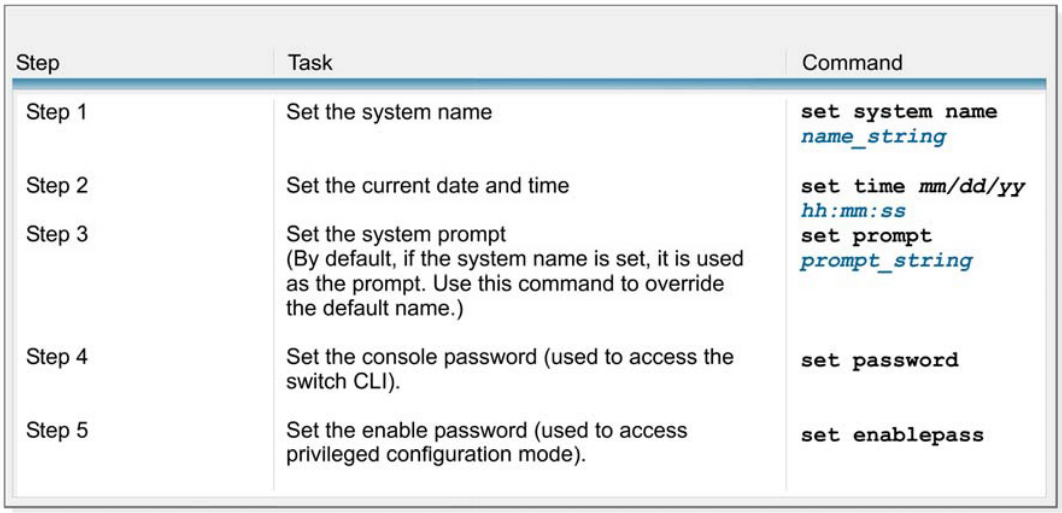
Table 2-9: Disabling Port Negotiation

	Command	Description
Step 1	Disable port negotiation.	**set port negotiation** *mod_num/port_num* **disable**
Step 2	Verify the port negotiation configuration.	**show port negotiation** [*mod_num/port_num*]

This example shows how to disable port negotiation and verify the configuration:

```
Console> (enable) set port negotiation 2/1 disable
Port 2/1 negotiation disabled
Console> (enable) show port negotiation 2/1
Port   Link Negotiation
-----  ----------------
 2/1   disabled
Console> (enable)
```

Basic Software Configuration (Step 5)



Step	Task	Command
Step 1	Set the system name	**set system name** *name_string*
Step 2	Set the current date and time	**set time** ***mm/dd/yy*** ***hh:mm:ss***
Step 3	Set the system prompt (By default, if the system name is set, it is used as the prompt. Use this command to override the default name.)	**set prompt** *prompt_string*
Step 4	Set the console password (used to access the switch CLI).	**set password**
Step 5	Set the enable password (used to access privileged configuration mode).	**set enablepass**

Configuring the Global System Settings

You can specify a variety of useful global system settings for your switch, such as system name, current date and time, system prompt, and passwords.

To configure the global system settings, perform these steps in privileged mode:

Table 2-10: Configuring Global System Settings

	Task	Command
Step 1	Set the system name.	**set system name** *name_string*
Step 2	Set the current date and time.	**set time** *mm/dd/yy hh:mm:ss*
Step 3	Set the system prompt. (By default, if the system name is set, it is used as the prompt. Use this command to override the default behavior.)	**set prompt** *prompt_string*
Step 4	Set the console password (used to access the switch's CLI).	**set password**
Step 5	Set the enable password (used to access privileged configuration mode).	**set enablepass**

This example shows how to configure the global system settings:

```
Console> (enable) set system name Catalyst 6000
System name set.
Catalyst 6000> (enable) set time 08/18/98 10:08:00
Sat Apr 18 1998, 10:08:00
Catalyst 6000> (enable) set password
Enter old password:
Enter new password:
Retype new password:
Password changed.
Catalyst 6000> (enable) set enablepass
Enter old password:
Enter new password:
Retype new password:
Password changed.
Catalyst 6000> (enable)
```

Lesson Summary

This lesson accomplished the following:

- Explained the basic functions of Layer 2 switches
- Explained the configuration of the Catalyst 6500's management interface
- Explained the configuration of basic system settings on the Catalyst 6500

Next Steps

After completing this lesson, go to:

- Layer 2 Redundancy

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/cfgnotes/

Layer 2 Redundancy

Overview

The Catalyst 6500 can support two Supervisor Engines, for redundancy. This lesson details the configuration of Supervisor Engine redundancy and discusses compatibility issues with some Cat OS features.

Importance

The Catalyst 6500 typically plays a mission-critical role, and switch redundancy is critical to maintaining uptime.

Objectives

Upon completing this lesson, you will be able to:

- Describe how Supervisor Engine redundancy works
- Explain the High Availability feature
- Configure the High Availability feature on the Catalyst 6500

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have equivalent knowledge

Outline

This lesson includes these sections:

- Overview
- How Supervisor Engine Redundancy Works
- Configuring High Availability
- Summary

How Supervisor Engine Redundancy Works

This section describes how dual Supervisor Engines can be used to provide redundancy for the Catalyst 6500.

Note	Redundant Supervisor Engines must be of the same type with the same model feature card.

When you install two Supervisor Engines, the first Supervisor Engine to come online becomes the active module; the second Supervisor Engine goes into standby mode. All administrative and network management functions, such as SNMP, command-line interface (CLI) access via the console, Telnet, Spanning-Tree Protocol (STP), Cisco Discovery Protocol (CDP), and VLAN Trunk Protocol (VTP) are processed on the active Supervisor Engine.

The console port on the standby Supervisor Engine is inactive and the module status for the standby Supervisor Engine appears as “standby.” However, the actual status for the uplink ports on the standby Supervisor Engine is up. This is because those ports can be used even when the Supervisor Engine in slot 2 is in standby mode.

Redundant Supervisor Engines must be installed in slots 1 and 2 of the chassis. Redundant Supervisor Engines are hot swappable. The system continues to operate with the same configuration after switching over to the redundant Supervisor Engine. For more information on this refer to the *Catalyst 6000 Family Module Installation Guid*e.

Note	The switchover time from the active to standby Supervisor Engine does not include spanning-tree convergence time.

At power-up, both Supervisor Engines run initial module-level diagnostics. Assuming both Supervisor Engines pass this level of diagnostics, the two Supervisor Engines communicate over the backplane, allowing them to cooperate during switching-bus diagnostics. The Supervisor Engine in slot 1 becomes active, and the Supervisor Engine in slot 2 enters standby mode. At this point, if the software versions of the two Supervisor Engines are different, or if the NVRAM configuration of the two Supervisor Engines is different, the active Supervisor Engine automatically downloads its software image and configuration to the standby Supervisor Engine.

If the background diagnostics on the active Supervisor Engine detect a major problem, or an exception occurs, the active Supervisor Engine resets. The standby Supervisor Engine detects that the active Supervisor Engine is no longer running and becomes active. The standby Supervisor Engine can detect if the active Supervisor Engine is not functioning, and can force a reset if necessary. If the Supervisor Engine that was reset comes online again it enters standby mode.

If you hot-insert a second Supervisor Engine, the second module communicates with the active Supervisor Engine after completing its initial module-level diagnostics. Since the active Supervisor Engine is already switching traffic on the backplane, no switching-bus diagnostics are run for the second Supervisor Engine because running diagnostics can disrupt normal traffic. The second Supervisor Engine immediately enters standby mode. If necessary, the active Supervisor Engine downloads the software image and configuration to the standby Supervisor Engine.

The Supervisor Engines on the Catalyst 6500 use two different Flash images, the *boot image* and the *runtime imag*e. The boot image filename is specified in the BOOT environment variable, which is stored in NVRAM. The runtime image is the boot image that the ROM monitor uses to boot the Supervisor Engine. After the system boots, the runtime image resides in dynamic RAM (DRAM).

When you power up or reset a switch with redundant Supervisor Engines, synchronization occurs to ensure that the runtime and boot images on the standby Supervisor Engine are the same as the images on the active Supervisor Engine.

The Supervisor Engines can have different runtime and boot images. If you change the BOOT environment variable or overwrite or destroy the current boot image on the Flash device that was used to boot the system, the runtime and boot images will differ. Whenever you reconfigure the boot image the active Supervisor Engine synchronizes its current boot image with the standby Supervisor Engine.

The boot image is read directly into the Flash file system. You can perform operations (such as **copy**, **delete**, **undelete**, and so on) on files stored on Flash memory devices, and you can store the boot image of the active Supervisor Engine in the standby Supervisor Engine's bootflash.

The Supervisor Engine has a Flash PC card (PCMCIA) slot (slot0:) in addition to the onboard Flash memory. This slot can hold a Flash PC card that can store additional boot images.

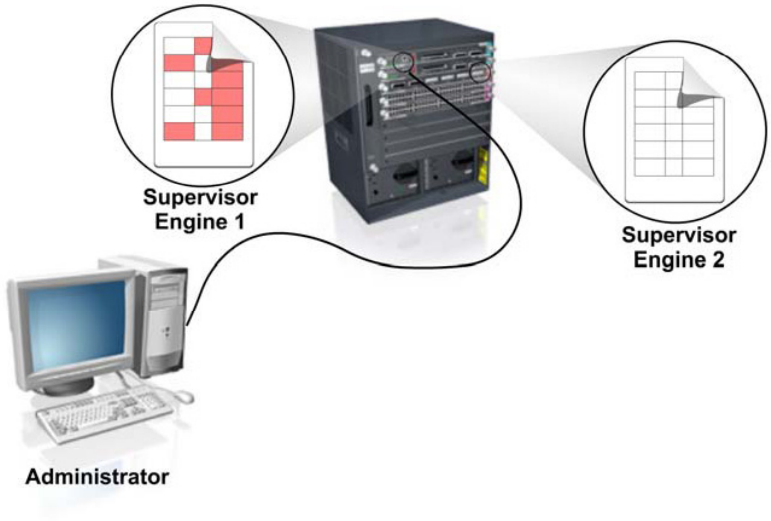
Note Throughout this course, the term *Flash PC card* is used in place of the term *PCMCIA car*d.

Since you can store multiple boot images, you must specify the name of the boot file image and the location of the image in Flash in order for the switch to boot and synchronize properly.

During the synchronization process, the active Supervisor Engine checks the standby Supervisor Engine's runtime image to make sure that it matches its own runtime image. The active Supervisor Engine checks the following three conditions:

- If it needs to copy its boot image to the standby Supervisor Engine.
- If the standby Supervisor Engine's bootstring needs to be changed.
- If the standby Supervisor Engine needs to be reset.

The High Availability Feature



This section describes the High Availability feature that is used to minimize the switchover time from the active Supervisor Engine to the standby Supervisor Engine. The High Availability feature ensures that there will be minimal disturbance to the steady-state operation of the network when the standby Supervisor Engine takes control of the switch.

Prior to the High Availability feature, the fast switchover feature was available to ensure that a switchover to the standby Supervisor Engine happened quickly. However, with fast switchover, all of the switch's features needed to be reinitialized and restarted when the standby Supervisor Engine assumed the active role because their state before the switchover was unknown.

The High Availability feature removes this limitation; the active Supervisor Engine communicates with the standby Supervisor Engine, keeping feature protocol states synchronized. When the standby Supervisor Engine is synchronized with the active Supervisor Engine, the standby can take over in the event of a failure and continue exactly where the failed Supervisor Engine left off.

The High Availability feature also provides a versioning option that allows you to run different software images on the active and standby Supervisor Engines.

With the High Availability feature, a system database is maintained on the active Supervisor Engine and updates are sent to the standby Supervisor Engine for any change of data in the system database. The active Supervisor Engine communicates and updates the standby Supervisor Engine when any state changes occur, ensuring that the standby Supervisor Engine knows the current protocol state of all supported features. The standby Supervisor Engine knows the current protocol states for all modules, ports, and VLANs. The protocols can initialize with this state information and start running immediately upon switchover.

The active Supervisor Engine controls the system bus (backplane), sends and receives packets to and from the network, and controls all modules. Protocols run on the active Supervisor Engine only. The standby Supervisor Engine is isolated from the system bus and does not switch packets. However, it *does* receive packets from the switching bus to learn and populate its Layer 2 forwarding table for Layer 2-switched flows. The standby Supervisor Engine also receives packets from the switching bus to learn and populate the Multilayer Switching (MLS) table for Layer 3-switched flows. The standby Supervisor Engine does not participate in the forwarding of any packets and does not communicate with any of the switch's modules.

Note Routing table entries in the active MSFC are not preserved with the High Availability feature, as this feature runs on the Supervisor Engines and not the MSFCs . However, you can configure both MSFCs on the active and standby Supervisor Engines with the same configuration and use Hot Standby Router Protocol (HSRP) to preserve routing table entries across the active and standby MSFCs.

When you enable High Availability, and the standby Supervisor Engine is running, image version compatibility is checked. If the versions are found to be compatible, the database synchronization is started. High Availability compatible features continue from the saved states on the standby Supervisor Engine after a switchover.

When you disable High Availability, the database synchronization is not done and all features must restart on the standby Supervisor Engine after a switchover.

If you change High Availability from enabled to disabled, synchronization from the active Supervisor Engine is stopped, and the standby Supervisor Engine discards all current synchronization data.

If you change High Availability from disabled to enabled, synchronization from the active to standby Supervisor Engine is started (provided the standby Supervisor Engine is present and its image version is compatible).

NVRAM synchronization occurs irrespective of High Availability being enabled or disabled (provided there are compatible NVRAM versions on the two Supervisor Engines).

If the standby Supervisor Engine is not installed during system bootup, the active Supervisor Engine detects this, and the database updates are not queued for synchronization. Similarly, when you reset or remove the standby Supervisor Engine, the synchronization updates are not queued and any pending updates in the synchronization queue are discarded. When you hot insert or restart a second Supervisor Engine that becomes the standby Supervisor Engine, the active Supervisor Engine downloads the entire system database to the standby Supervisor Engine. Only after this global synchronization is completed does the active Supervisor Engine queue and synchronize the individual updates to the standby Supervisor Engine.

Note When you hot insert or restart a second Supervisor Engine, it might take a few minutes for the global synchronization process to complete.

High Availability Supported Features

Note MLS flows are preserved from the active Supervisor Engine to the standby Supervisor Engine.

The Catalyst 6500 family switch features are classified into three categories with respect to High Availability support:

- Supported features—High Availability is fully supported; the feature's database is synchronized from the active Supervisor Engine to the standby Supervisor Engine.
- Compatible features—High Availability is not supported; the feature's database is not synchronized from the active Supervisor Engine to the standby Supervisor Engine. However, *the feature can be enabled* (operational) along with the High Availability feature.
- Incompatible features—High Availability is not supported; the feature's database is not synchronized from the active Supervisor Engine to the standby Supervisor Engine. Additionally, *the feature cannot be enabled* if High Availability is enabled and similarly, High Availability cannot be enabled if the feature is enabled.

Note Timers and statistics are not synchronized from the active to the standby Supervisor Engine.

Table 2-11: High Availability Feature Support

Supported Feature	Compatible Features	Incompatible Features
COPS-DS	ASLB	Dynamic VLAN
COPS-PR	CDP	GVRP
DTP	GMRP	Port security
EtherChannel	IGMP snooping	Protocol filtering
IOS ACLs	RMON	
MLS	RSVP	
PAgP	SNMP	
QoS	Telnet sessions	
SPAN		
STP		
Trunking		
UDLD		
VACLs		

Versioning

- Different but compatible images run on active and standby engines
- Release 5.4(1)CSX or later

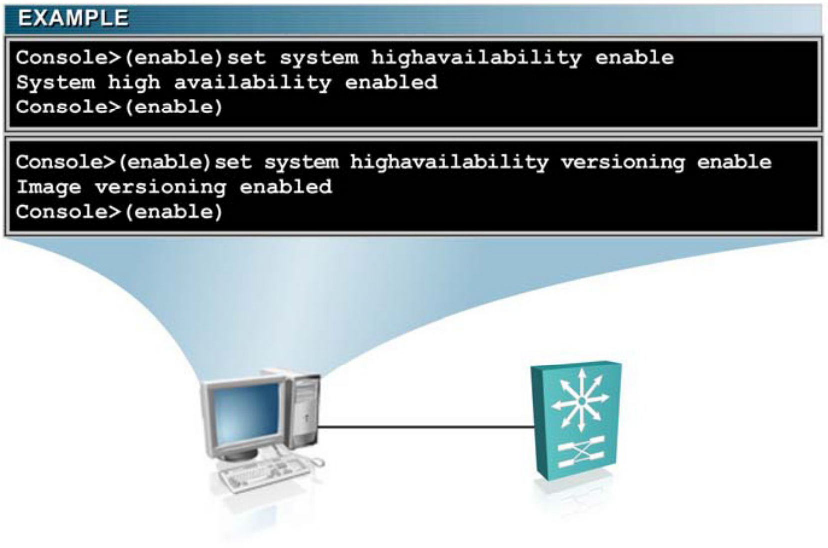
With High Availability versioning enabled, you can have two different, but compatible, images on the active and standby Supervisor Engines. The active Supervisor Engine exchanges image version information with the standby Supervisor Engine, and determines whether the images are compatible for enabling High Availability. If the active and standby Supervisor Engines are not running compatible image versions, the High Availability feature cannot be enabled.

Image versioning is supported in Supervisor Engine software releases 5.4(1)CSX and later. Image versioning is not supported with images prior to release 5.4(1)CSX. Therefore, with versioning enabled, the High Availability feature is fully supported with the active and standby Supervisor Engines running different images as long as both images are release 5.4(1)CSX or later.

Note When you install two Supervisor Engines, the first Supervisor Engine to come online becomes the active module; the second Supervisor Engine goes into standby mode. In a switch with two Supervisor Engines installed, at power up the Supervisor Engine in slot 1 becomes active, and the Supervisor Engine in slot 2 enters standby mode. At this point, if the software versions of the two Supervisor Engines are different, or if the NVRAM configuration of the two Supervisor Engines is different, and versioning is not enabled, the active Supervisor Engine automatically downloads its software image and configuration to the standby Supervisor Engine.

Configuring High Availability

This section demonstrates how to configure the High Availability feature on the Catalyst 6500.



```
Console>(enable)set system highavailability enable
System high availability enabled
Console>(enable)
```

```
Console>(enable)set system highavailability versioning enable
Image versioning enabled
Console>(enable)
```

Enabling or Disabling High Availability

High Availability is disabled by default. To enable or disable High Availability, perform these tasks in privileged mode:

Table 2-12: Enabling High Availability

Task	Command
Enable or disable High Availability.	**`set system highavailability enable	disable`**

This example shows how to enable High Availability:

```
Console> (enable) set system highavailability enable
System high availability enabled.
Console> (enable)
```

This example shows how to disable High Availability:

```
Console> (enable) set system highavailability disable
```

```
System high availability disabled.
Console> (enable)
```

Enabling or Disabling High Availability Versioning

High Availability versioning is disabled by default. To enable or disable High Availability versioning, perform these tasks in privileged mode:

Table 2-13: Enabling Versioning

Command	Description
Enable or disable High Availability versioning.	**`system highavailability versioning enable	disable`**

This example shows how to enable High Availability versioning:

```
Console> (enable) set system highavailability versioning enable
Image versioning enabled.
Console> (enable)
```

This example shows how to disable High Availability versioning:

```
Console> (enable) set system highavailability versioning disable
Image versioning disabled.
Console> (enable)
```

Showing High Availability Settings and Operational Status

The **`show system highavailability`** command displays the following:

- High Availability setting (enabled or disabled)
- Versioning setting (enabled or disabled)
- High Availability operational status (based on whether the standby Supervisor Engine is present and operational).
  The operational status field displays one of the following:
  - OFF (High-Availability-not-enabled): The High Availability option in NVRAM is disabled.
  - OFF (standby-supervisor-not-present): The standby Supervisor Engine is not installed.
  - OFF (standby-supervisor-image-incompatible): The standby Supervisor Engine is running a different image than the active Supervisor Engine and it is not version compatible (the versioning option in NVRAM is enabled). No synchronization can be performed (even a configuration change in NVRAM on the active Supervisor Engine cannot be propagated to the standby because of the version incompatibility).

- OFF (standby-supervisor-image-nvram-only-compat): The standby Supervisor Engine is running a different image than the active Supervisor Engine (versioning option in NVRAM is enabled) and the image is only NVRAM compatible (that is, a configuration change in NVRAM on the active Supervisor Engine will be propagated to the standby). However, High Availability cannot be supported.
- OFF (standby-supervisor-not-operational-yet): The standby Supervisor Engine is detected, but is not operational (not online yet).
- OFF (High-Availability-not-operational-yet): The standby Supervisor Engine is operational (online), but High Availability is not operational yet (when the system is booted from reset, it takes a few minutes before High Availability is operational).
- ON: High Availability is operational. The active Supervisor Engine's features have started queuing their state changes for synching to the standby Supervisor Engine. To show the High Availability configuration and operational states, perform this task:

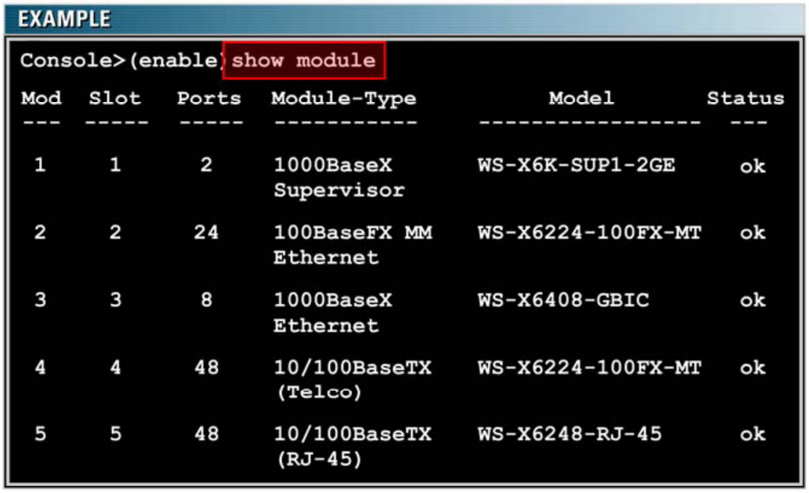
In this example, High Availability and versioning are disabled:

```
Console> (enable) show system highavailability
Highavailability: disabled
Highavailability versioning: disabled
Highavailability Operational-status: OFF (high-availability-not-enabled)
Console> (enable)
```

In this example, High Availability is enabled:

```
Console> (enable) set system highavailability enable
System high availability enabled.
Console> (enable)
Console> (enable) show system highavailability
Highavailability: enabled
Highavailability versioning: disabled
Highavailability Operational-status: ON
Console> (enable)
```

Checking Module Status



```
Console>(enable) show module
Mod  Slot  Ports  Module-Type        Model             Status
---  ----- -----  -----------     ----------------- ---

 1     1     2    1000BaseX       WS-X6K-SUP1-2GE    ok
                  Supervisor

 2     2    24    100BaseFX MM    WS-X6224-100FX-MT  ok
                  Ethernet

 3     3     8    1000BaseX       WS-X6408-GBIC      ok
                  Ethernet

 4     4    48    10/100BaseTX    WS-X6224-100FX-MT  ok
                  (Telco)

 5     5    48    10/100BaseTX    WS-X6248-RJ-45     ok
                  (RJ-45)
```

Catalyst 6500 family switches are multimodule systems. You can see what modules are installed, as well as the MAC address ranges and version numbers for each module, using the show module [mod_num] command. Specify a particular module number to see detailed information on that module.

This example shows how to check module status. The output shows that there is one Supervisor Engine and four additional modules installed in the chassis.

```
Console> (enable) show module
Mod Slot Ports Module-Type               Model               Status
--- ---- ----- ------------------------- ------------------- --------
1    1    2     1000BaseX Supervisor      WS-X6K-SUP1-2GE     ok
2    2    24    100BaseFX MM Ethernet     WS-X6224-100FX-MT   ok
3    3    8     1000BaseX Ethernet        WS-X6408-GBIC       ok
4    4    48    10/100BaseTX (Telco)      WS-X6248-TEL        ok
5    5    48    10/100BaseTX (RJ-45)      WS-X6248-RJ-45      ok
```

This example shows how to check module status on a specific module:

```
Console> (enable) show module 4
Mod Slot Ports Module-Type               Model               Status
--- ---- ----- ------------------------- ------------------- --------
4    4    48    10/100BaseTX (Telco)      WS-X6248-TEL        ok
Mod Module-Name         Serial-Num
--- ------------------- -----------
4                        SAD03140787
Mod MAC-Address(es) Hw Fw Sw
--- -------------------------------------- ------ ---------- -----------------
4    00-50-54-bf-59-64 to 00-50-54-bf-59-93 0.103 4.2(0.24)V 5.2(1)CSX Console>
Console> (enable)
```

Lesson Summary

This lesson accomplished the following:

- Described how Supervisor Engine redundancy works
- Explained the High Availability feature
- Detailed the configuration of the High Availability feature

Next Steps

After completing this lesson, go to:

- EtherChannel

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw5_1/cnfigide/supcfg.htm

EtherChannel

Overview

When interconnecting Catalyst 6500s, even a 1 Gbps interface could potentially become a bottleneck. Therefore, Cisco provides the EtherChannel feature, which can aggregate multiple physical interfaces into a single logical interface. This lesson details the configuration of EtherChannel on the Catalyst 6500.

Importance

EtherChannel is a critical feature for eliminating potential bottlenecks when interconnecting Catalyst switches.

Objectives

Upon completing this lesson, you will be able to:

- Describe how the Catalyst 6500 enhances Cisco’s EtherChannel feature
- Configure EtherChannel (for Hybrid and Native mode)

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have equivalent knowledge

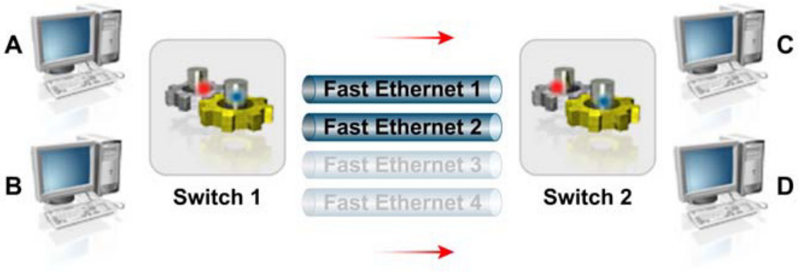
Outline

This lesson includes these sections:

- Overview
- EtherChannel
- Configuring EtherChannel
- Summary

EtherChannel

This section discussed the EtherChannel feature supported on the Catalyst 6500.



EtherChannel bundles individual Ethernet links into a single logical link that provides bandwidth up to 1.6 Gbps (8-port Fast EtherChannel full duplex) or 16 Gbps (8-port Gigabit EtherChannel) between a Catalyst 6500 family switch and another Etherchannel capable device. The Catalyst 5000 and 5500 series only support a 4-port EtherChannel with FastEthernet interfaces. On the Catalyst 5000/5500 series these interfaces had to begin with an odd-numbered port and also had to be sequential. These limitations do not exist on the Catalyst 6500 series switches.

A Catalyst 6500 switch supports a maximum of 128 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet ports on any module in a Catalyst 6500 switch. All ports in an EtherChannel must be the same speed.

Note The network device to which a Catalyst 6500 switch is connected may impose its own limits on the number of ports in an EtherChannel.

If a link within an EtherChannel fails, traffic previously carried over the failed port switches to the remaining links within the EtherChannel. A trap message is sent upon a failure identifying the switch, the EtherChannel, and the failed port. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other port of the EtherChannel.

EtherChannels can be configured as trunks. After a channel has been formed, configuring any port in the channel as a trunk applies the configuration to all ports in the channel. Identically configured trunk ports can be configured as an EtherChannel.

Understanding Administrative Groups

Configuring an EtherChannel creates an administrative group, designated by an integer between 1 and 1024, to which the EtherChannel belongs. When an administrative group is created, you can assign an administrative group number or let the next available administrative group number be assigned automatically. Forming a channel without specifying an administrative group number creates a new automatically numbered administrative group. An administrative group may contain a maximum of eight ports.

Understanding EtherChannel IDs

Each EtherChannel is automatically assigned a unique EtherChannel ID. Use the **`show`** channel group *admin_group* command to display the EtherChannel ID.

Understanding Port Aggregation Protocol

The Port Aggregation Protocol (PAgP) facilitates the automatic creation of EtherChannels by exchanging packets between Ethernet ports. PAgP packets are exchanged only between ports in **`auto`** and **`desirable`** modes. Ports configured in **`on`** or **`off`** mode do not exchange PAgP packets. The protocol learns the capabilities of port groups dynamically and informs the other ports. Once PAgP correctly identifies matched EtherChannel links, it groups the ports into an EtherChannel bundle. The EtherChannel bundle is then added to the spanning tree topology as a single bridge port.

EtherChannel includes four user-configurable modes: **`on`**, **`off`**, **`auto`**, and **`desirable`**. Only **`auto`** and **`desirable`** are PAgP modes. The auto and desirable modes can be modified with the **`silent`** and **`non-silent`** keywords. By default, ports are in **`auto silent`** mode.

Table 2-14: EtherChannel Modes

Mode	Description
`on`	Mode that forces the port to channel without PAgP. With the **`on`** mode, a usable EtherChannel exists only when a port group in **`on`** mode is connected to another port group in **`on`** mode.
`off`	Mode that prevents the port from channeling.
`auto`	PAgP mode that places a port into a passive negotiating state, in which the port responds to PAgP packets it receives, but does not initiate PAgP packet negotiation. (Default)
`desirable`	PAgP mode that places a port into an active negotiating state, in which the port initiates negotiations with other ports by sending PAgP packets.
`silent`	Keyword that is used with the **`desirable`** mode when no traffic is expected from the other device. This keyword is used to prevent the link from being reported to the Spanning-Tree Protocol as down. (Default)

Both the **`auto`** and **`desirable`** modes allow ports to negotiate with connected ports to determine if they can form an EtherChannel based on criteria such as port speed, trunking state, and VLAN numbers.

Ports can form an EtherChannel when they are in different PAgP modes as long as the modes are compatible. For example:

A port in **`desirable`** mode can form an EtherChannel successfully with another port that is in **`desirable`** or **`auto`** mode.

A port in **`auto`** mode can form an EtherChannel with another port in **`desirable`** mode.

A port in **`auto`** mode cannot form an EtherChannel with another port that is also in auto mode, since neither port will initiate negotiation.

Understanding Frame Distribution

EtherChannel distributes frames 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.

Enter a **`show module`** command for the Supervisor Engine to determine if EtherChannel frame distribution is configurable on your switch.

If the display shows the "Sub-Type" to be "L2 Switching Engine I WS-F6020," then EtherChannel frame distribution is not configurable on your switch; it uses source and destination MAC addresses.

EtherChannel frame distribution is configurable with all other switching engines. The default is to use source and destination IP addresses.

When configurable, EtherChannel frame distribution can use either MAC addresses or IP addresses, and either source or destination, or both source and destination, addresses. 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 only going 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 frame distribution.

EtherChannel Configuration Guidelines

- Assign all ports to the same VLAN,or configure as a trunk
- If you create a trunk, set all ports to the same trunking mode
- Configure all ports for the same speed and duplex
- None of the ports should be configured as a SPAN destination port
- Ports should not have port security enabled

If improperly configured, some EtherChannel ports are disabled automatically to avoid network loops and other problems. Follow these guidelines to avoid configuration problems:

- Assign all ports in an EtherChannel to the same VLAN, or configure them as trunk ports.
- If you configure the EtherChannel as a trunk, configure the same trunk mode on all the ports in the EtherChannel. Configuring ports in an EtherChannel in different trunk modes can have unexpected results.
- An EtherChannel supports the same allowed range of VLANs on all the ports in a trunking EtherChannel. If the allowed range of VLANs is not the same for a port list, the ports do not form an EtherChannel even when set to the **`auto`** or **`desirable`** mode with the **`set port channel`** command.
- Ports with different port path costs, set by the **`set spantree portcost`** command, can form an EtherChannel as long they are otherwise compatibly configured. Setting different port path costs does not, by itself, make ports incompatible for the formation of an EtherChannel.
- Do not configure the ports in an EtherChannel as dynamic VLAN ports. Doing so can adversely affect switch performance.
- An EtherChannel will not form with ports that have different GARP VLAN Registration Protocol (GVRP), GARP Multicast Registration Protocol (GMRP), and quality of service (QoS) configurations.
- Configure all ports in an EtherChannel to operate at the same speed and duplex mode.

- An EtherChannel will not form with ports where the port security feature is enabled.

- You cannot enable the port security feature for ports in an EtherChannel.

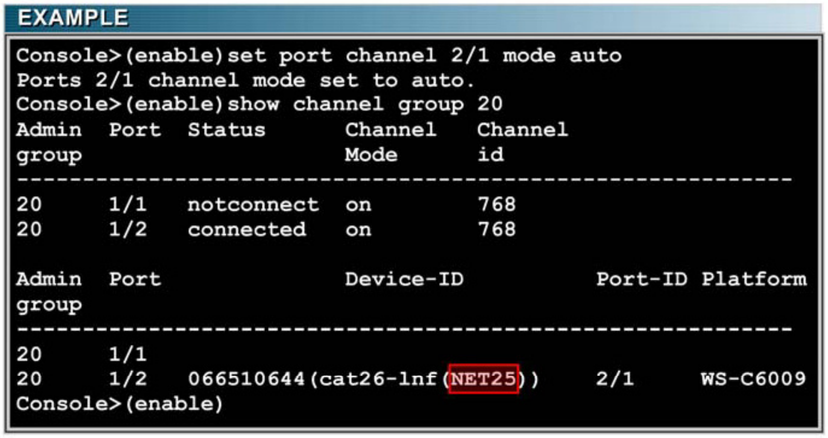
- An EtherChannel will not form if one of the ports is a SPAN destination port.

- An EtherChannel will not form if protocol filtering is set differently on the ports.

- Enable all ports in an EtherChannel. If you disable a port in an EtherChannel, it is treated as a link failure and its traffic is transferred to one of the remaining ports in the EtherChannel.

Configuring EtherChannel

This section demonstrates how to configure EtherChannel on the Catalyst 6500.



EXAMPLE

```
Console>(enable)set port channel 2/1 mode auto
Ports 2/1 channel mode set to auto.
Console>(enable)show channel group 20
Admin  Port  Status      Channel   Channel
group                    Mode      id
-----------------------------------------------------------
20     1/1   notconnect  on        768
20     1/2   connected   on        768

Admin  Port              Device-ID          Port-ID Platform
group
-----------------------------------------------------------
20     1/1
20     1/2   066510644(cat26-lnf(NET25))    2/1     WS-C6009
Console>(enable)
```

EtherChannels in Native IOS are configured much differently than in CatOS. Enabling EtherChannel on a group of ports in Native IOS requires the use of a **`port-channel interface`**. If all conditions are valid for the group of ports, they will form a port-channel. By default, all interfaces have port-channeling disabled even when an interface is configured as a **`switchport`**.

To configure a group of interfaces to be part of an EtherChannel, the command **`channel-group`** ***`<group-number>`*** **`mode`** ***`<channel-mode>`*** must be configured under each interface individually. If the **`switchport`** command is removed from the configuration, all the commands related to that switchport will no longer show in the configuration. However, reconfiguring the port as a **`switchport`** returns all the previous commands. As a result, configuring and unconfiguring a port as a **`switchport`** does not clear the port channel group information.

Once a channel-group is created, all of the configuration must be entered on the port-channel interface and not on the individual physical ports. Any commands entered on the port-channel are propagated to all the physical ports transparently. Commands configured on a channel member's physical interface may remove the interface from the channel-group.

Table 2-15: Configuring EtherChannel

Function	CatOS	Native IOS
Creating the channel	**CatOS (enable) set port channel 4/3-4 on** Port(s) 4/3-4 are assigned to admin group 613. Port(s) 4/3-4 channel mode set to on. **CatOS (enable)**	**NativeIOS#configure terminal** Enter configuration commands, one per line. End with CNTL/Z. **NativeIOS(config)#interface port-channel 1** **NativeIOS(config-if)#exit** **NativeIOS(config)#interface fast 4/3** **NativeIOS(config-if)#channel-group 1 mode on** **NativeIOS(config-if)#interface fast 4/4** **NativeIOS(config-if)#channel-group 1 mode on** **NativeIOS(config-if)#**
Setting the channel mode	**CatOS (enable) set port channel** **<mod/port> mode {on | off | desirable |** **auto} [silent | non-silent]**	**NativeIOS(config-if)#channel-group** **<channel-group number> mode {on | auto** **[non-silent] | desirable [non-silent]}**
Verifying the configuration	**Show port channel** **Show port channel** *<mod>/<port>* **Show port channel** *<channel-group>*	**show etherchannel** **show etherchannel** *<channel-group>* **show interfaces etherchannel** **show interfaces** *<interface-type>* ***<mod>/<port>* etherchannel**

Lesson Summary

This lesson accomplished the following:

- Described how the Catalyst 6500 enhances Cisco's EtherChannel feature
- Explained the Configuration of EtherChannel (for Hybrid and Native mode)

Next Steps

After completing this lesson, go to:

- Spanning Tree Protocol

References

For additional information, refer to these resources:

- http://www.cisco.com/warp/public/779/largeent/learn/technologies/fast_echannel.html

Spanning Tree Protocol

Overview

While having redundant connections in a switched topology, bridging loops can occur. Therefore, Cisco uses the Spanning Tree Protocol to provide redundancy while eliminating the bridging loops. This lesson details the configuration and optimization of the Spanning Tree Protocol on the Catalyst 6500.

Importance

By default, the Spanning Tree Protocol can take 50 seconds to converge. Therefore, the optimization techniques presented in this chapter are critical to a resilient Layer 2 design.

Objectives

Upon completing this lesson, you will be able to:

- Explain the purpose of the Spanning Tree Protocol (STP)
- Configure STP on the Catalyst 6500
- Optimize the STP configuration with PortFast, UplinkFast, and BackboneFast

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have equivalent knowledge

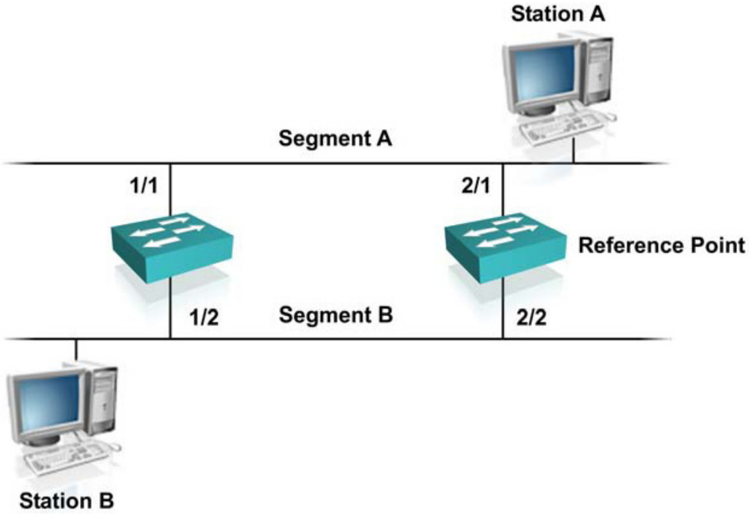
Outline

This lesson includes these sections:

- Overview
- Spanning Tree Protocol
- Configuring the Spanning Tree Protocol
- Summary

Spanning Tree Protocol (STP)

This section describes operation of the Spanning Tree Protocol.



STP is a link management protocol that provides path redundancy while preventing undesirable loops in the network. For an Ethernet network to function properly, only one active path must exist between two stations.

STP (IEEE 802.1D bridge protocol) is used on all Ethernet, Fast Ethernet, Gigabit Ethernet, and Token Ring-based VLANs. A single instance of STP runs on each configured VLAN (provided you do not manually disable STP). You can enable and disable STP on a per-VLAN basis.

Note In network environments using IEEE 802.1Q trunks, a single instance of spanning tree runs for the entire Layer 2 topology, instead of a single spanning tree for each configured VLAN.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. In STP, an algorithm calculates the best loop-free path throughout a switched network. Switches send and receive spanning tree packets at regular intervals. The switches do not forward the packets, but use the packets to identify a loop-free path. The default configuration has STP enabled for all VLANs.

Multiple active paths between stations cause loops in the network. If a loop exists in the network, you might receive duplicate messages. When loops occur, some switches see stations on both sides of the switch. This condition confuses the forwarding algorithm and allows duplicate frames to be forwarded.

To provide path redundancy, STP defines a tree that spans all switches in an extended network. STP forces certain redundant data paths into a standby (blocked) state. If one network segment in the STP becomes unreachable, or if STP costs change, the spanning tree algorithm reconfigures the spanning tree topology and reestablishes the link by activating the standby path.

STP operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

Bridge Protocol Data Units

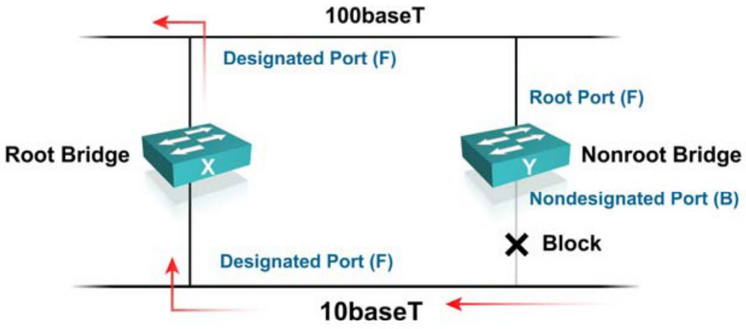
The stable active topology of a switched network is determined by the following:

- The unique switch identifier (MAC address) associated with each switch.
- The path cost to the root associated with each switch port.
- The port identifier (MAC address) associated with each switch port.

Each configuration BPDU contains the following minimal information:

- The unique identifier of the switch that the transmitting switch believes to be the root switch.
- The cost of the path to the root from the transmitting port.
- The identifier of the transmitting port.

Spanning Tree Operations

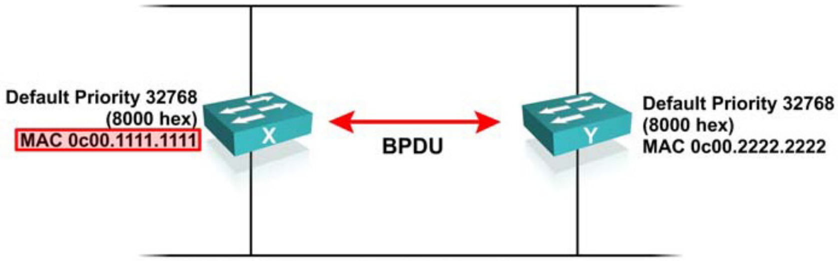


- **One root bridge per network**
- **One root port per nonroot bridge**
- **One designated port per segment**

In a Spanning Tree topology there is one, and only one, Root Bridge which acts as the “reference point” for the Layer 2 network. All non-root bridges have one, and only one, Root Port which is the port that is closest, in terms of cost, to the Root Bridge. In the above example, the top port of switch Y is the Root Port, because it only has a cost of 19 to the Root Bridge, since it is a 10/100 Mbps port. The bottom port on switch Y has a cost of 100 to the Root Bridge, since it is a 10 Mbps port.

Each segment has one, and only one, Designated Port. The Designated Port is the port on a segment that is closest, in terms of cost, to the Root Bridge. Therefore, all ports on a Root Bridge are Designated Ports. All other ports are Blocked Ports, which still listen to BPDUs, but do not transmit or receive any other traffic.

Election of the Root Switch



BPDU = Bridge protocol data unit
(default = sent every 2 seconds)
Root bridge = Bridge with the lowest bridge ID
Bridge ID = Bridge priority + bridge MAC address

All switches in an extended LAN participating in STP gather information on other switches in the network through an exchange of data messages called bridge protocol data units (BPDUs). This exchange of messages results in the following actions:

- The election of a unique root switch for the stable spanning tree network topology.
- The election of a designated switch for every switched LAN segment.
- The removal of loops in the switched network by placing redundant switch ports in a blocking state.

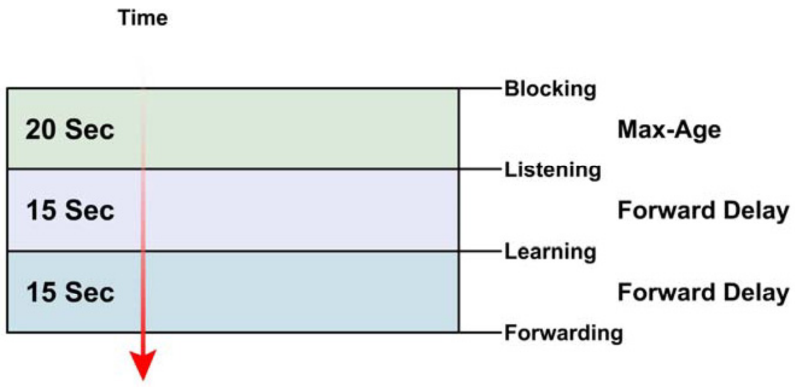
The STP root switch is the logical center of the spanning tree topology in a switched network. All paths anywhere in the switched network that are not needed to reach the root switch are placed in STP blocking mode. The following table describes the root switch variables that affect the entire spanning tree performance.

Table 2-16: STP Root Switch Parameters

Parameter	Description
`Hello Timer`	Determines how often the switch broadcasts its hello messages to other switches.
`Maximum Age Timer`	Measures the age of the received protocol information recorded for a port. This setting also ensures that STP information is discarded when its age limit exceeds the value of the maximum age parameter recorded by the switch.
`Forward Delay Timer`	Configures the amount time spent by a port in the learning and listening states.

BPDUs contain information about the transmitting switch and its ports, including switch and port MAC addresses, switch priority, port priority, and port cost. STP uses this information to elect the root switch and root port for the switched network, as well as the root port and designated port for each switched segment.

STP Port States



- Spanning Tree uses the timers as it passes through the Spaning-Tree Protocol 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 switch port transitions directly from nonparticipation in the stable topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate throughout the switched LAN before beginning to forward frames. They must allow the frame lifetime to expire for frames that have been forwarded using the old topology.

Each port on a switch using STP exists in one of the following five states:

- Blocking - 20 seconds
- Listening - 15 seconds
- Learning - 15 seconds
- Forwarding
- Disabled

A port moves through these five states as follows:

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

- From forwarding to disabled

You can modify each port state by using management software. When you enable STP, every switch in the network goes through the blocking state and the transitory states of listening and learning at power up. If properly configured, each port stabilizes to the forwarding or blocking state. Before the spanning tree algorithm places a port in the forwarding state, the following occurs:

- The port is put into the listening state while it waits for protocol information that suggests it should go to the blocking state.
- The port waits for the expiration of a protocol timer that moves the port into the learning state.
- In the learning state, the port continues to block frame forwarding as it learns station addressing and location information for the forwarding database.
- The expiration of a protocol timer moves the port into the forwarding state, where both learning and forwarding are enabled.

Blocking State

A port in the blocking state does not participate in frame forwarding. After initialization, BPDUs are sent out each port on the switch. A switch initially assumes that it is the root bridge until it is told otherwise during BPDU exchanges with other switches. These exchanges establish which switch in the network is really the root bridge. If only one switch resides in the network, no exchange occurs, the forward delay timer expires, and the ports move into the listening state. A switch always enters the blocking state following initialization.

A port in the blocking state performs as follows:

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

Listening State

The listening state is the first transitional state a port enters after the blocking state. The port enters this state when STP determines that the port should participate in frame forwarding. Learning is disabled in the listening state.

A port in the listening state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate 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.
- Processes BPDUs received from the system module.
- Receives and responds to network management messages.

Learning State

A port in the learning state prepares to participate in frame forwarding. The port enters the learning state from the listening state.

A port in the learning state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Incorporates 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 port in the forwarding state forwards frames. The port enters the forwarding state from the learning state.

A port in the forwarding state performs as follows:

- Forwards frames received from the attached segment.
- Forwards frames switched from another port for forwarding.
- Incorporates 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

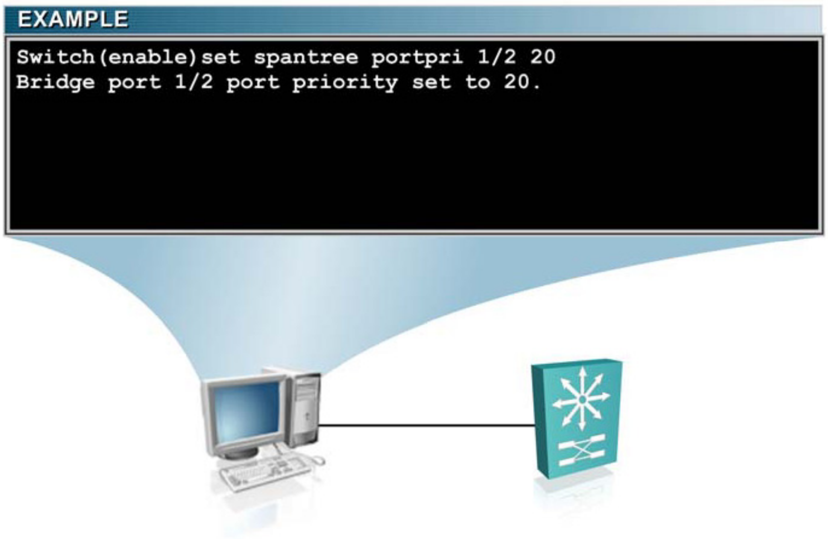
A port in the disabled state does not participate in frame forwarding or STP. A port in the disabled state is virtually non-operational.

A disabled port performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate station location into its address database (There is no learning, so there is no address database update).
- Receives BPDUs, but does not direct them to the system module.
- Does not receive BPDUs for transmission from the system module.
- Receives and responds to network management messages.

Configuring the Spanning Tree Protocol

Configuring the Port Priority



```
Switch(enable)set spantree portpri 1/2 20
Bridge port 1/2 port priority set to 20.
```

- **Path priority is a Spanning-Tree Protocol parameter that can be modified to influence the links that are forwarding or blocking**

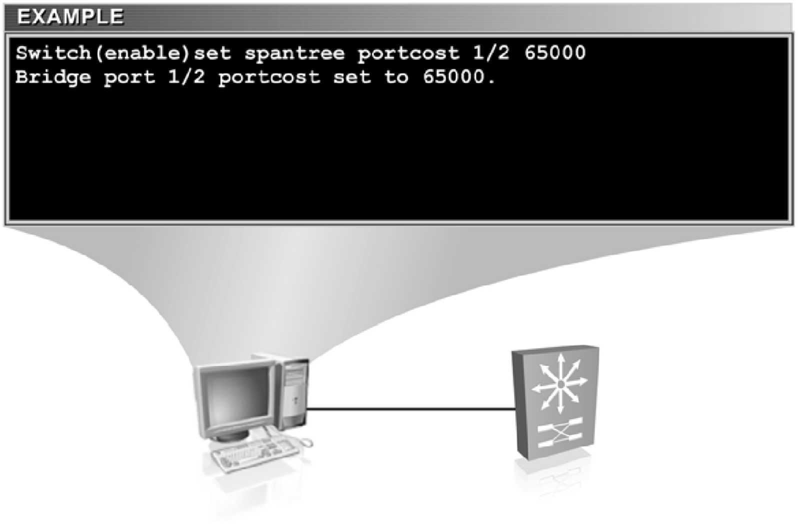
You can configure the port priority of switch ports. The port with the lowest priority value forwards frames for all VLANs. The possible port-priority range is 0 through 63. The default is 32. In the case of all ports having the same priority value, the port with the lowest port number forwards frames.

The following is the syntax for setting the port priority in both the CatOS and Native IOS environments.

Table 2-17: Setting the Port Priority

Function	CatOS Syntax	Native IOS Syntax
Configure Port Priority	**`set spantree portpri`** *mod_num/port_num priority [vlans]*	**`spanning-tree port-priority`** *`port_priority`* *This command is given in interface-configuration mode.*

Configuring the Port Cost



EXAMPLE

```
Switch(enable)set spantree portcost 1/2 65000
Bridge port 1/2 portcost set to 65000.
```

- **Path cost is used to decide which ports should forward and which ports should block**
- **Path cost is a sum of port costs from the root bridge**

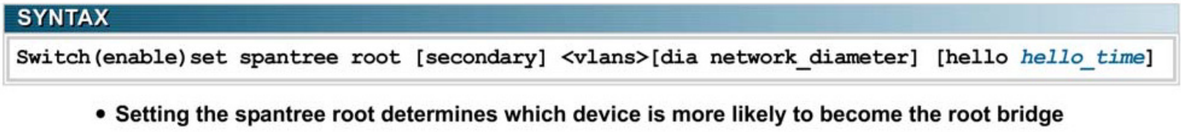
You can configure the port cost of switch ports. Ports with lower port costs are more likely to be chosen to forward frames. Assign lower numbers to ports attached to faster media (such as Fast Ethernet and Gigabit Ethernet) and higher numbers to ports attached to slower media. The possible range is 1 to 65535. The default differs for each media type.

The following is the syntax for setting the port cost in both the CatOS and Native IOS environments.

Table 2-18: Setting the Port Cost

Function	CatOS Syntax	Native IOS Syntax
Configure Port Cost	**`set spantree portcost`** *mod_num/port_num cost]*	**`spanning-tree cost`** *`cost`* *This command is given in interface-configuration mode*.

Configuring a Primary Root Switch



SYNTAX

```
Switch(enable)set spantree root [secondary] <vlans>[dia network_diameter] [hello hello_time]
```

- Setting the spantree root determines which device is more likely to become the root bridge

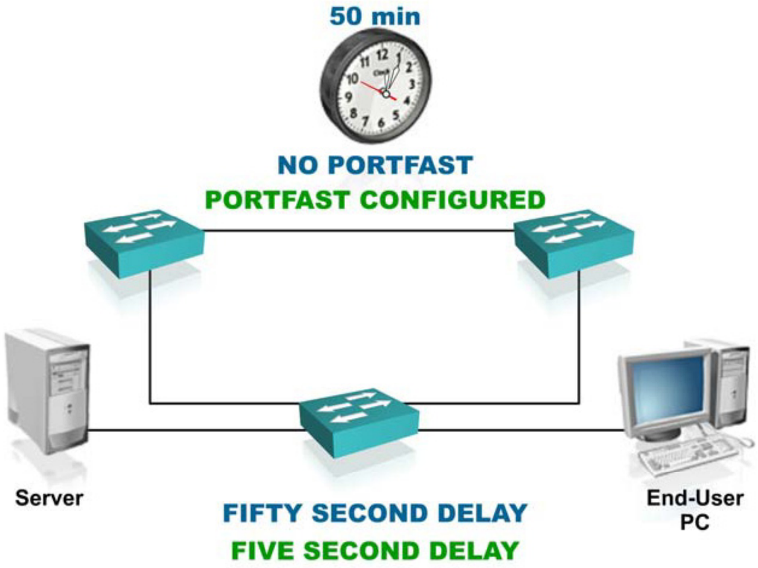
The **`set spantree root`** command reduces the bridge priority (the value associated with the switch) from the default (32,768) to a significantly lower value, which allows the switch to become the root switch. When you specify a switch as the primary root, the default bridge priority is modified to 8192 so that the switch becomes the root bridge for the specified VLANs. If this setting does not result in the switch becoming the root bridge, modify the bridge priority to be 100 less than the bridge priority of the current root switch. Since different VLANs could potentially have different root switches, the bridge VLAN-priority chosen makes this switch the root for all the VLANs specified. If reducing the bridge priority as low as 1 still does not make the switch the root switch, the system displays a message.

The following is the syntax for setting the primary root switch in both the CatOS and Native IOS environments.

Table 2-19: Designating the Root Switch

Function	CatOS Syntax	Native IOS Syntax
Configure a Primary Root Switch	**`set spantree root`** [**`secondary`**] <*`vlans`*> [*`network_diameter`*] [hello *`hello_time`*]	**`spanning-tree vlan`** *`vlan_id`* {root {primary | secondary} [diameter *`net-diameter`* [hello-time *`hello-time`*]]}] *This command is given in global configuration mode.*

PortFast



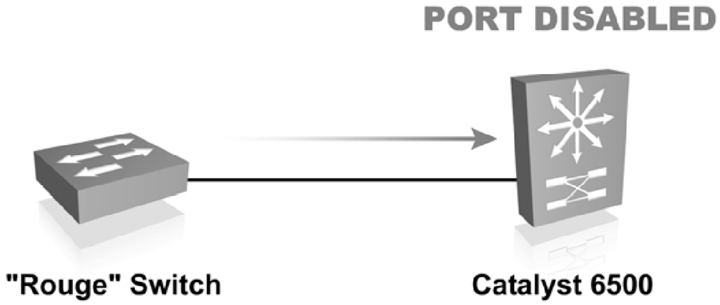
PortFast causes a spanning tree port to enter the forwarding state immediately, bypassing the listening and learning states. You can use PortFast on switch ports connected to a single workstation or server to allow those devices to connect to the network immediately, rather than waiting for spanning tree to converge.

The following is the syntax for configuring PortFast in both the CatOS and Native IOS environments.

Table 2-20: Enabling PortFast

Function	CatOS Syntax	Native IOS Syntax
Configure PortFast	**`set spantree portfast`** *`mod_num/port_num`*	**`spanning-tree portfast`** *This command is given in global configuration mode.*

BPDU Guard



To prevent loops in a network, the PortFast mode is supported on non-trunking access ports only, because these ports typically do not transmit or receive BPDUs. The most secure implementation of PortFast is to enable it only on ports that connect to end stations. However, because PortFast can be enabled on non-trunking ports interconnecting two switches, spanning tree loops can occur if BPDUs are still being transmitted and received on those ports.

PortFast BPDU guard can prevent loops by moving a non-trunking port into the errdisable state when a BPDU is received on that port. When the BPDU guard feature is enabled on the switch, spanning tree shuts down PortFast-configured interfaces that receive BPDUs, rather than putting them into the spanning tree blocking state. In a valid configuration, PortFast-configured interfaces should never receive BPDUs. Reception of a BPDU by a PortFast-configured interface signals an invalid configuration, such as the connection of an unauthorized device. The BPDU guard feature provides a secure response to invalid configurations, because the administrator must manually put the interface back into service.

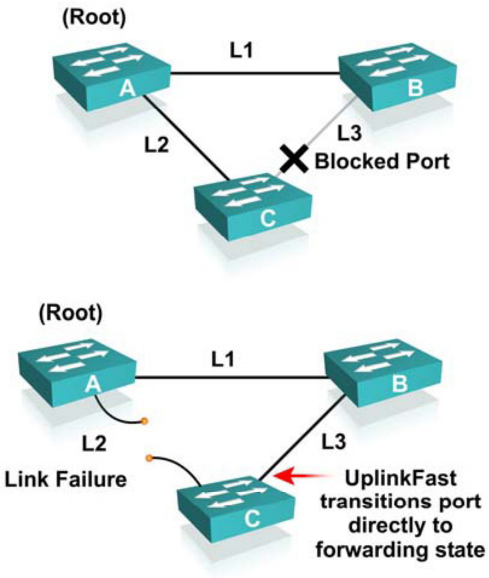
Note When enabled on the switch, spanning tree applies the PortFast BPDU guard feature to all PortFast-configured interfaces.

The following is the syntax for configuring BPDU Guard in both the CatOS and Native IOS environments.

Table 2-21: Enabling BPDU Guard

Function	CatOS Syntax	Native IOS Syntax
Configure BPDU Guard	`set spantree portfast bpdu-guard enable`	`spanning-tree portfast bpduguard` *This command is given in global configuration mode.*

UplinkFast



UplinkFast dramatically reduces the STP convergence time for wiring closet switches. UplinkFast has the ability to detect a directly attached failure, and almost immediately transition a blocked port to a forwarding state.

To inform upstream switches about the new path, the UplinkFast switch sends a series of multicast messages, each message having the source address of a MAC address in the switch's CAM table. Since upstream switches will flood these multicast messages out all ports except the port they were received on, the network can very quickly recover.

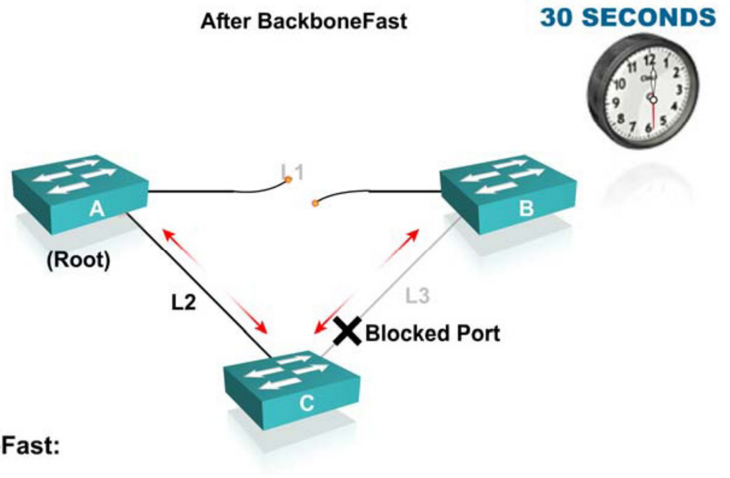
However, the UplinkFast feature should only be used on wiring closet switches. After enabling the UplinkFast feature, the bridge priority is changed to 49,152, well above the default value of 32,768. Also, the port cost of each port on the switch is increased by 3,000.

The following is the syntax for configuring UplinkFast in both the CatOS and Native IOS environments.

Table 2-22: Enabling UplinkFast

Function	CatOS Syntax	Native IOS Syntax
Enable UplinkFast	`set spantree uplinkfast enable`	`spanning-tree uplinkfast` *This command is given in global configuration mode.*

BackboneFast



BackboneFast:

- **Detects indirect failures**
- **Bypasses the Max Age timer, so that the port transitions directly to the Listening state**

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 indicates 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 bridge). Under normal spanning tree rules, the switch ignores inferior BPDUs for the configured maximum aging time, as specified by the *agingtime* variable of the **`set spantree maxage`** command.

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

If the switch has alternate paths to the root bridge, it uses these alternate paths to transmit a new kind of PDU called the Root Link Query PDU. The switch sends the Root Link Query PDU out all alternate paths to the root bridge. 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 bridge indicate that the switch has lost connectivity to the root bridge, 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 bridge, 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 blocking state), through the listening and learning states, and into the forwarding state.

The following is the syntax for configuring BackboneFast in both the CatOS and Native IOS environments.

Table 2-23: Enabling BackboneFast

Function	CatOS Syntax	Native IOS Syntax
Enable UplinkFast	**`set spantree backbonefast enable`**	**`spanning-tree backbone`** *This command is given in global configuration mode.*

Lesson Summary

This lesson accomplished the following:

- Explained the purpose of the Spanning Tree Protocol (STP)
- Detailed the configuration of STP on the Catalyst 6500
- Demonstrated how to optimize the STP configuration with PortFast, UplinkFast, and BackboneFast

Next Steps

After completing this lesson, go to:

- VLANs and Trunking

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_6_3/confg_gd/spantree.htm

VLANs and Trunking

Overview

In a switched environment with several hundred connections, the ports on a switch should be divided into multiple broadcast domains for performance purposes. This lesson details the configuration and optimization of broadcast domains (i.e., VLANs) on the Catalyst 6500 switch.

Importance

To minimize the impact of broadcast traffic proper design and configuration of VLANs is essential.

Objectives

Upon completing this lesson, you will be able to:

- Explain the purpose of VLANs
- Configure VLANs in both Hybrid and Native modes
- Explain the need for and advantages of trunking
- Configure trunking in both Hybrid and Native modes

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have equivalent knowledge

Outline

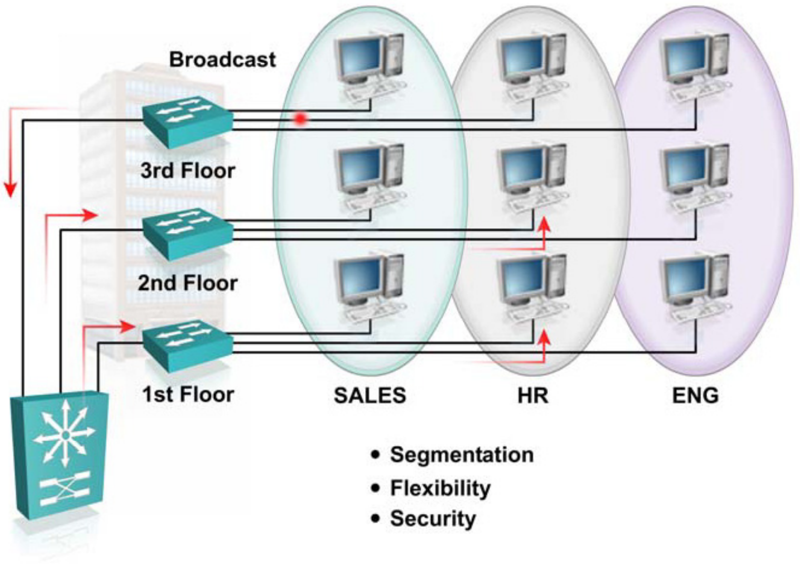
This lesson includes these sections:

- Overview
- VLANs
- Trunking
- VLAN Trunk Protocol (VTP)
- Configuring Trunking
- Configuring VTP
- Configuring VLANs
- Summary

VLANs

This section introduces the purpose of features of VLANs.

A VLAN = A Broadcast Domain = Logical Network (Subnet)



- Segmentation
- Flexibility
- Security

VLAN Overview

A typical LAN configuration is configured according to the physical infrastructure it is connecting. Users are grouped based on their location in relation to the hub they are plugged into and how the cable is run to the wiring closet. Segmentation is typically provided by the router interconnecting each shared hub.

This type of segmentation does not group users according to their workgroup association or need for bandwidth. Engineering users can be plugged into the same hub as accounting and administration users because of their respective physical locations. They all share the same segment and contend for the same bandwidth, although the bandwidth requirements may vary greatly according to workgroup or department.

Additionally, this segmentation technique requires that each hub connected to a router port have a unique subnet address. This prevents a logical assignment of network addresses across the network campus resulting in security issues.

Switched Internetworking Configuration

The problems associated with shared LANs and the emergence of LAN switches are causing traditional LAN configurations to be replaced with switched VLAN internetworking configurations. Switched VLAN configurations vary from LAN configurations in the following ways:

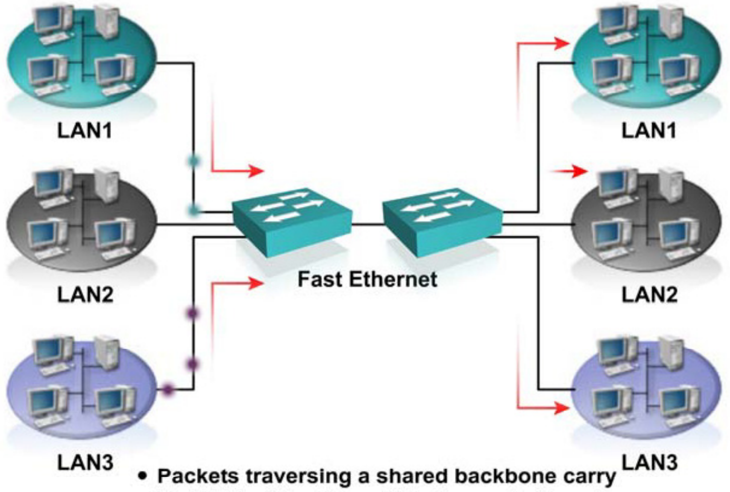
Switches replace front-end hubs in the wiring closet. Switches are easily installed with little or no cabling changes, and can completely replace a shared hub with per port service to each user.

VLANs are created to provide the segmentation services traditionally provided by routers in LAN configurations. A VLAN is a switched network that is logically segmented by functions, project teams, or applications without regard to the physical location of users. Each switch port can be assigned to a VLAN. Broadcasts are limited to the VLAN they originated in. Ports that do not belong to that VLAN do not receive these broadcasts. This improves the overall performance of the network.

Communication between VLANs is provided by Layer 3 routing.

Trunking

This section describes trunking and the trunking options supported on the Catalyst 6500.



- **Packets traversing a shared backbone carry VLAN identification within the packet header**
- **VLAN Identification Options:**
  - Cisco ISL
  - IEEE 802.1Q

A trunk is a point-to-point link between one or more Ethernet switch ports and another networking device, such as a router or a switch. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network. Two trunking encapsulations are available on all Ethernet ports on the Catalyst 6500:

- Inter-Switch Link (ISL) - Cisco-proprietary trunking encapsulation
- IEEE 802.1Q - Industry-standard trunking encapsulation

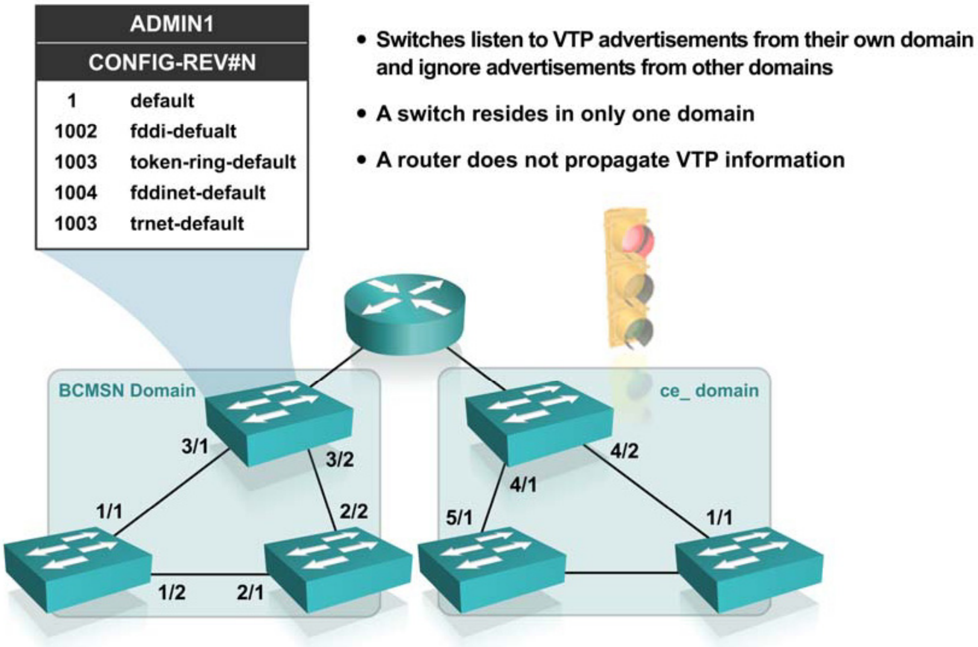
You can configure a trunk on a single Ethernet port or on an EtherChannel bundle.

Ethernet trunk ports support five different trunking modes. In addition, you can specify if the trunk will use ISL encapsulation, 802.1Q encapsulation, or if the encapsulation type will be autonegotiated. For trunking to be autonegotiated, the ports must be in the same VLAN Trunk Protocol (VTP) domain. However, you can use the **`on`** or **`nonegotiate`** mode to force a port to become a trunk, even if it is in a different domain.

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

VLAN Trunk Protocol (VTP)

This section discusses how the VLAN Trunk Protocol can be used to minimize administrative effort when configuring VLANs.



ADMIN1	
CONFIG-REV#N	
1	default
1002	fddi-defualt
1003	token-ring-default
1004	fddinet-default
1003	trnet-default

- Switches listen to VTP advertisements from their own domain and ignore advertisements from other domains
- A switch resides in only one domain
- A router does not propagate VTP information

Before you create VLANs, you must decide whether or not 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 is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs on a network-wide basis. 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.

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 the no-management domain state until the switch receives an advertisement for a domain over a trunk link or you manually configure a management domain. You cannot create or modify VLANs on a VTP server until the management domain name is specified or learned.

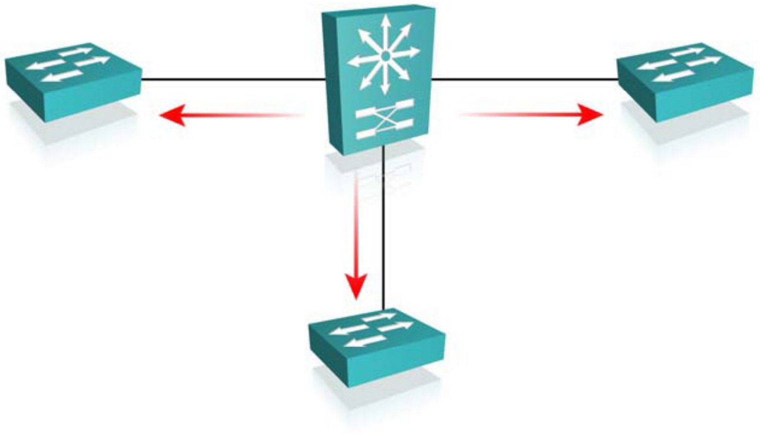
If the switch receives a VTP advertisement over a trunk link, it inherits the management domain name and the VTP configuration revision number. The switch ignores advertisements with a different management domain name or a lower configuration revision number.

If you configure the switch in VTP transparent mode, you can create and modify VLANs, but the changes will only affect 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, including Inter-Switch Link (ISL), IEEE 802.1Q, IEEE 802.10, and ATM LAN Emulation (LANE).

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



- **Server Mode = Create/delete global VLANs**
- **Client Mode = Cannot change any VLANs**
- **Transparent = Create/delete local VLANs, ignore VTP updates**

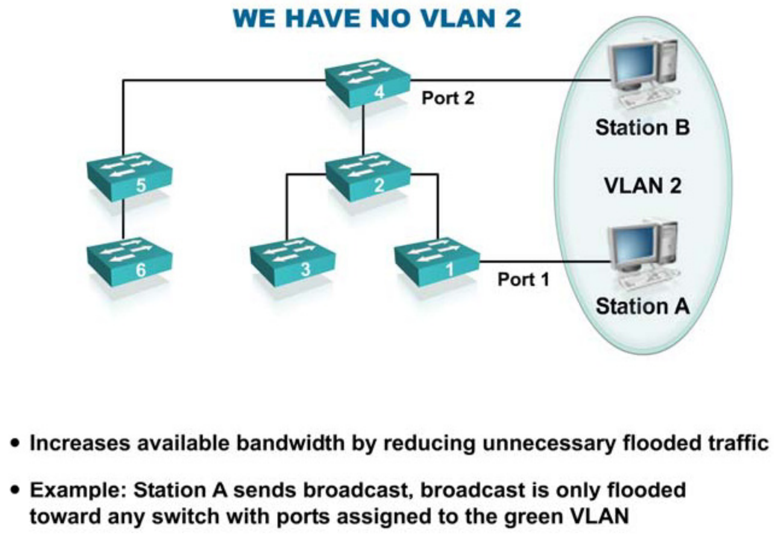
You can configure a switch to operate in any one of these VTP modes:

- Server—In VTP server mode, you can create, modify, and delete VLANs and specify other configuration parameters (such as VTP version and VTP pruning) 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 ports.

VTP Pruning



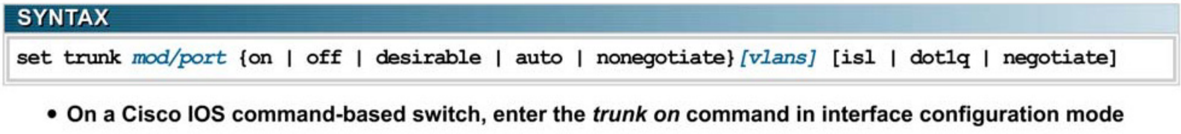
- Increases available bandwidth by reducing unnecessary flooded traffic
- Example: Station A sends broadcast, broadcast is only flooded toward any switch with ports assigned to the green VLAN

VTP pruning enhances network bandwidth use by reducing the amount of unnecessary traffic that is flooded throughout the network, such as broadcast, multicast, unknown, and flooded unicast packets. VTP pruning increases available bandwidth by restricting flooded traffic to those trunk links that the traffic must use to access the appropriate network devices. By default, VTP pruning is disabled.

Make sure that all devices in the management domain support VTP pruning before enabling it. VTP pruning is supported in Supervisor Engine software release 2.3 and later.

Configuring Trunking

This section demonstrates how to configure trunks on a Catalyst 6500.



SYNTAX

```
set trunk mod/port {on | off | desirable | auto | nonegotiate}[vlans] [isl | dot1q | negotiate]
```

- **On a Cisco IOS command-based switch, enter the *trunk on* command in interface configuration mode**

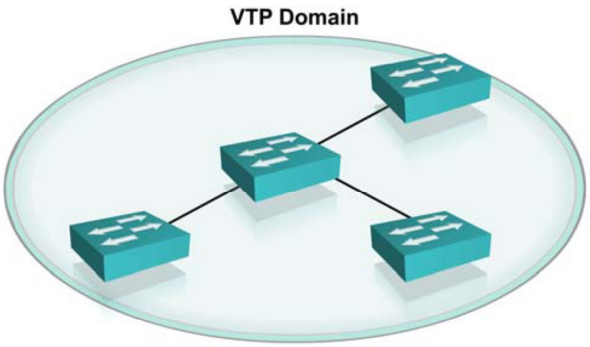
Both ISL and 802.1Q trunking modes are supported in the CatOS and the Native IOS. Trunking for the Native IOS works exactly the same way as CatOS except for the default setting in Native IOS, which is desirable rather than auto.

Table 2-24: Configuring Trunking

Function	CatOS	Native IOS
Enabling ISL trunk	CatOS (enable) **set trunk 4/1 on isl** Port(s) 4/1 trunk mode set to on. Port(s) 4/1 trunk type set to isl.	NativeIOS#**conf t** Enter configuration commands, one per line. End with CNTL/Z. NativeIOS(config)#**interface fastethernet 4/1** NativeIOS(config-if)#**switchport trunk encap isl** NativeIOS(config-if)#**switchport mode trunk** 3d22h: %DTP-SP-5-TRUNKPORTON: Port Fa4/1 has become isl NativeIOS(config-if)#**^Z** NativeIOS#
Enabling dot1q trunk	CatOS (enable) **set trunk 4/1 on dot1q** Port(s) 4/1 trunk mode set to on. Port(s) 4/1 trunk type set to dot1q CatOS (enable) **set vlan 2 4/1** VLAN 2 modified. VLAN 1 modified. VLAN Mod/Ports ---- ----------------- 2 1/1 4/1 Note: In the case of dot1q, it is very important that the Native VLAN matches across the trunk link. The **set vlan <vlan-id> <mod/port>** command is used in CatOS to set the native vlan for the trunk.	NativeIOS#**conf t** Enter configuration commands, one per line. End with CNTL/Z. NativeIOS(config)#**interface fastethernet 4/1** NativeIOS(config-if)#**switchport trunk encap dot1q** NativeIOS(config-if)#**switchport mode trunk** **3d22h: %DTP-SP-5-TRUNKPORTON:** Port Fa4/1 has become dot1q NativeIOS(config-if)#**switchport trunk native vlan 2** NativeIOS(config-if)#**^Z** **NativeIOS#**
Changing trunk mode	CatOS (enable) **set trunk mod/port {on** **| off | desirable | auto | nonegotiate}** **[vlans] [isl | dot1q | negotiate]**	NativeIOS(config-if)#**switchport mode {access |** **trunk | multi | dynamic {auto | desirable}}**

Displays trunking status	**show trunk** **show trunk** <*mod*> **show port** <*mod*>/<*port*>	**show interfaces trunk** **show interfaces trunk module** <*number*> **show interfaces** <*interface-type*> <*mod*>/<*port*> **show interfaces status**

Defining the Allowed VLANs on a Trunk



- Not all VLANs should be carried on a trunk link
- Choose which VLANs should be on the trunk

When you configure a trunk port, all VLANs are added to the allowed VLANs list for that trunk. However, you can remove VLANs from the allowed list to prevent traffic for those VLANs from passing over the trunk. You cannot remove VLAN 1, the default VLAN, from the allowed list.

When you first configure a port as a trunk, the **set trunk** command always adds all VLANs to the allowed VLAN list for the trunk, even if you specify a VLAN range (any specified VLAN range is ignored). To modify the allowed VLANs list, use a combination of the **clear trunk** and **set trunk** commands to specify the allowed VLANs.

To define the allowed VLAN list for a trunk port, perform this task in privileged mode:

Table 2-25: Clearing VLANs from a Trunk

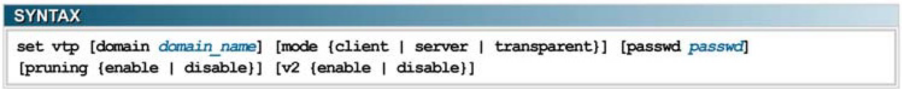
	Task	Command
Step 1	Remove VLANs from the allowed VLANs list for a trunk.	**clear trunk** *mod_num/port_num vlans*
Step 2	(Optional) Add specific VLANs to the allowed VLANs list for a trunk.	**set trunk** *mod_num/port_num vlans*
Step 3	Verify the allowed VLAN list for the trunk.	**show trunk** *[mod_num/port_num]*

This example shows how to define the allowed VLANs list for trunk port 1/1 to allow VLANs 1–100, VLAN 250, and VLANs 500–1005, and how to verify the allowed VLAN list for the trunk:

```
Console> (enable) clear trunk 1/1 101-499
Removing Vlan(s) 101-499 from allowed list.
Port 1/1 allowed vlans modified to 1-100,500-1005.
```

Configuring VTP

This section illustrates how to configure VTP on a Catalyst 6500.



SYNTAX

```
set vtp [domain domain_name] [mode {client | server | transparent}] [passwd passwd]
[pruning {enable | disable}] [v2 {enable | disable}]
```

In Native IOS, the VTP configuration is defined in the VLAN database mode. Changes to the VLAN database and VTP occur when the VLAN data is applied (this occurs when the user exits from the VLAN database configuration mode). The default Native IOS VTP configuration is below; note that the default VTP mode is Server.

```
NativeIOS#show vtp status
VTP Version : 2
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 6
VTP Operating Mode : Server
VTP Domain Name : null
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0xE2 0x4F 0xC0 0xD6 0x94 0xBB 0x31 0x9A
Configuration last modified by 0.0.0.0 at 6-27-01 02:04:20
Local updater ID is 0.0.0.0 (no valid interface found)
```

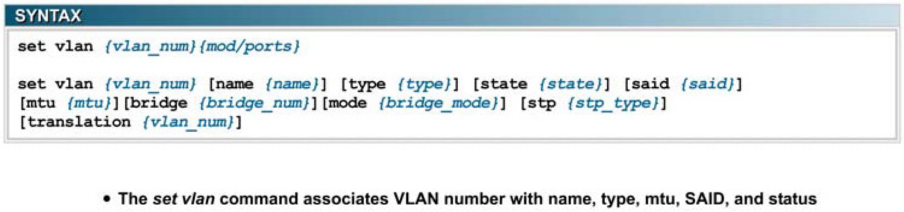
Note that in both the Cat OS and the Native IOS, the default mode is the server.

Table 2-26: Configuring VTP

Function	CatOS	Native IOS
Configuring VTP	CatOS (enable) **set vtp domain cisco** VTP domain cisco modified	NativeIOS#**vlan database** NativeIOS(vlan)#**vtp domain cisco** Changing VTP domain name from null to cisco NativeIOS(vlan)#**exit** APPLY completed. Exiting....
Changing VTP mode	CatOS (enable) **set vtp mode client** VTP domain cisco modified CatOS (enable) **set vtp mode server** VTP domain cisco modified CatOS (enable) **set vtp mode transparent** VTP domain cisco modified	NativeIOS#**vlan database** NativeIOS(vlan)#**vtp client** Setting device to VTP CLIENT mode. NativeIOS(vlan)#**vtp server** Setting device to VTP SERVER mode. NativeIOS(vlan)#**vtp transparent** Setting device to VTP TRANSPARENT mode. NativeIOS(vlan)#**exit** APPLY completed. Exiting....
Enabling VTP pruning	CatOS (enable) **set vtp pruning enable** This command will enable the pruning function in the entire management domain. All devices in the management domain should be pruning-capable before enabling. Do you want to continue (y/n) [n]? **y** VTP domain cisco modified	NativeIOS#**vlan database** NativeIOS(vlan)#**vtp pruning** Pruning switched ON NativeIOS(vlan)#**exit** APPLY completed.
Display VTP configuration	CatOS (enable) **show vtp domain**	NativeIOS#**show vtp status**

Configuring VLANs

This section shows how to configure VLANs on a Catalyst 6500.



SYNTAX

```
set vlan {vlan_num}{mod/ports}

set vlan {vlan_num} [name {name}] [type {type}] [state {state}] [said {said}]
[mtu {mtu}][bridge {bridge_num}][mode {bridge_mode}] [stp {stp_type}]
[translation {vlan_num}]
```

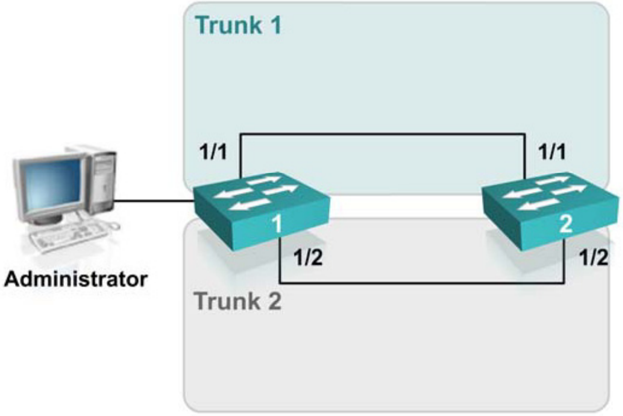
- **The *set vlan* command associates VLAN number with name, type, mtu, SAID, and status**

The concept and functionality of VLANs are identical between Native IOS and CatOS. However, the configuration methods between the two implementations differ significantly. While VLANs are created via **`set`** commands in CatOS, they are created via the **`'VLAN Database'`** configuration mode in Native IOS.

Table 2-27: Configuring VLANs

Function	CatOS	Native IOS
Creating VLAN	CatOS (enable) **set vlan 2** Vlan 2 configuration successful	NativeIOS#**vlan database** NativeIOS(vlan)#**vlan 2** VLAN 2 added: Name: VLAN0002 NativeIOS(vlan)#**exit** APPLY completed. Exiting....
Deleting a VLAN	CatOS (enable) **clear vlan 2** This command will deactivate all ports on vlan 2 Do you want to continue(y/n) [n]?y Vlan 2 deleted	NativeIOS#**vlan database** NativeIOS(vlan)#**no vlan 2** Deleting VLAN 2... NativeIOS(vlan)#**exit** APPLY completed. Exiting....
Assigning port to the VLAN	CatOS (enable) **set vlan 2 1/1** VLAN 2 modified. VLAN 10 modified. VLAN Mod/Ports ---- --------------- 2 1/1	NativeIOS#**conf t** Enter configuration commands, one per line. End with CNTL/Z. NativeIOS(config)#**interface gigabit2/2** NativeIOS(config-if)#**switchport** NativeIOS(config-if)#**switchport access vlan 2** NativeIOS(config-if)#**^Z** NativeIOS#
To see the VLAN status	**show vlan**	**show vlan**

Example: Load-Sharing VLAN Traffic Over Parallel Trunks



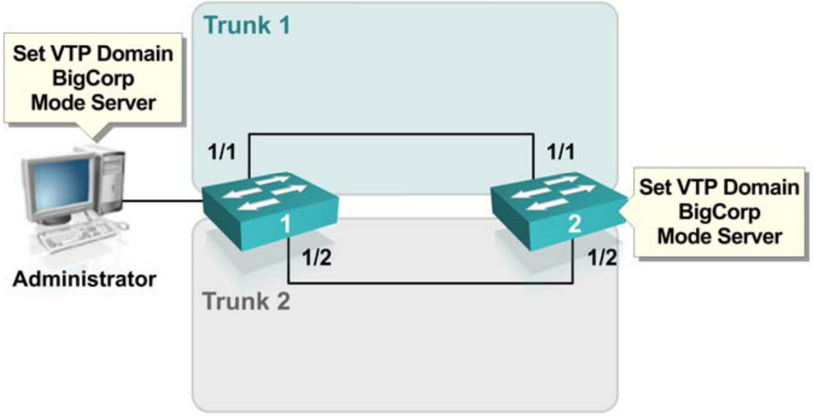
Using spanning-tree port-VLAN priorities, you can load-share VLAN traffic over parallel trunk ports so that traffic from some VLANs travels over one trunk, while traffic from other VLANs travels over the other trunk. This configuration allows traffic to be carried over both trunks simultaneously (instead of keeping one trunk in blocking mode), which reduces the total traffic carried over each trunk while still maintaining a fault-tolerant configuration.

The diagram above shows a parallel trunk configuration between two switches, using the Fast Ethernet uplink ports on the Supervisor Engine.

By default, the port-VLAN priority for both trunks is equal (a value of 32). Therefore, STP blocks port 1/2 (Trunk 2) for each VLAN on Switch 1 to prevent forwarding loops. Trunk 2 is not used to forward traffic unless Trunk 1 fails.

This example shows how to configure the switches so that traffic from multiple VLANs is load-balanced over the parallel trunks.

Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step1)

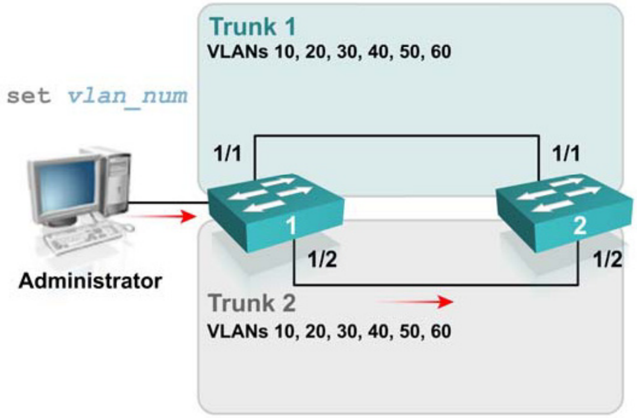


Step 1 Configure a VTP domain on both Switch 1 and Switch 2 (by entering the set vtp command) so that the VLAN information configured on Switch 1 is learned by Switch 2. Make sure Switch 1 is a VTP server.

You can configure Switch 2 as a VTP client or as a VTP server.

```
Switch_1> (enable) set vtp domain BigCorp mode server
VTP domain BigCorp modified
Switch_1> (enable)
Switch_2> (enable) set vtp domain BigCorp mode server
VTP domain BigCorp modified
Switch_2> (enable)
```

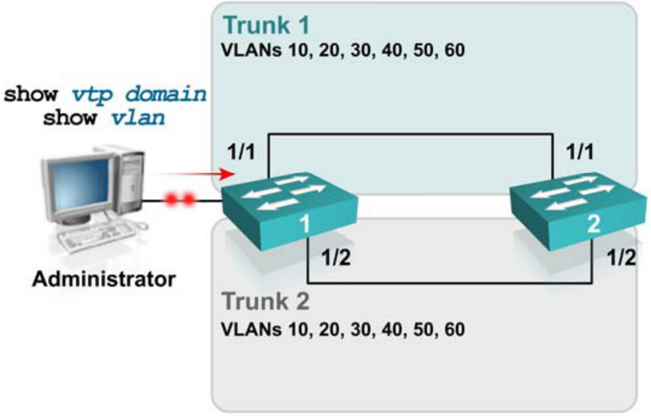
Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step2)



Step 2 Create the VLANs on Switch 1 by entering the set vlan command. In this example, you see VLANs 10,20, 30, 40, 50, and 60.

```
Switch_1> (enable) set vlan 10
Vlan 10 configuration successful
Switch_1> (enable) set vlan 20
Vlan 20 configuration successful
Switch_1> (enable) set vlan 30
Vlan 30 configuration successful
Switch_1> (enable) set vlan 40
Vlan 40 configuration successful
Switch_1> (enable) set vlan 50
Vlan 50 configuration successful
Switch_1> (enable) set vlan 60
Vlan 60 configuration successful
Switch_1> (enable)
```

Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step3)

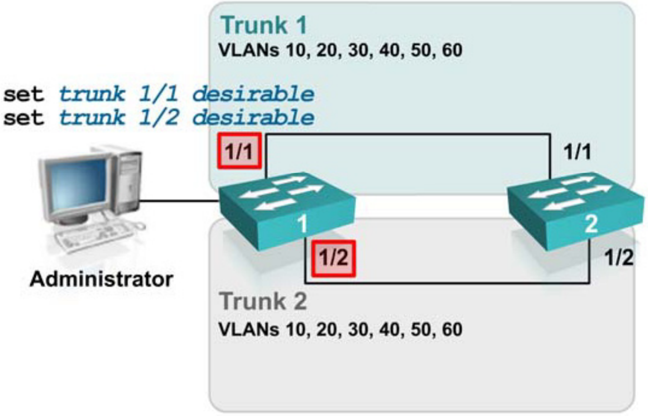


Step 3 Verify the VTP and VLAN configuration on Switch 1 by entering the **`show vtp domain`** and show vlan commands.

```
Switch_1> (enable) show vtp domain
Domain Name                      Domain Index VTP Version Local Mode  Password
-------------------------------- ------------ ----------- ----------- ----------
BigCorp                           1            2           server   -

Vlan-count Max-vlan-storage Config Revision Notifications
---------- ---------------- --------------- -------------
11         1023             13              disabled
Last Updater    V2 Mode  Pruning  PruneEligible on Vlans
--------------- -------- -------- -------------------------
172.20.52.10    disabled enabled   2-1000.
Switch_1> (enable) show vlan
VLAN Name                             Status    Mod/Ports, Vlans
---- -------------------------------- --------- ----------------------------
1     default                         active     1/1-2
                                        2/1-12
                                        5/1-2
10    VLAN0010                        active
20    VLAN0020                        active
30    VLAN0030                        active
40    VLAN0040                        active
50    VLAN0050                        active
```

```
60    VLAN0060                        active
1002  fddi-default                    active
1003  token-ring-default              active
1004  fddinet-default                 active
1005  trnet-default active
<...output truncated...>
Switch_1> (enable)
```

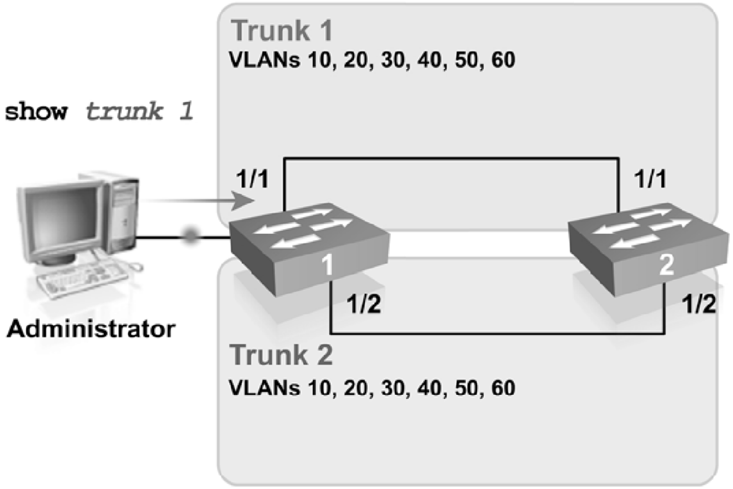
Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 4)



Step 4 Configure the Supervisor Engine uplinks on Switch 1 as ISL trunk ports by entering the **set trunk** command. Specifying the **desirable** mode on the Switch 1 ports causes the ports on Switch 2 to negotiate to become trunk links (assuming that the Switch 2 uplinks are in the default **auto** mode).

```
Switch_1> (enable) set trunk 1/1 desirable
Port(s) 1/1 trunk mode set to desirable.
Switch_1> (enable) 04/21/1998,03:05:05:DISL-5:Port 1/1 has become isl trunk
Switch_1> (enable) set trunk 1/2 desirable
Port(s) 1/2 trunk mode set to desirable.
Switch_1> (enable) 04/21/1998,03:05:13:DISL-5:Port 1/2 has become isl trunk
```

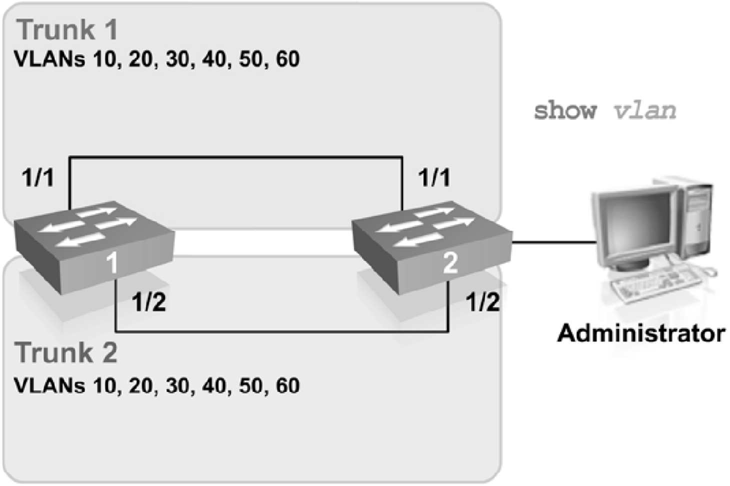
Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 5)



Step 5 Verify that the trunk links are up by entering the **`show trunk`** command.

```
Switch_1> (enable) show trunk 1
Port     Mode        Encapsulation  Status      Native vlan
-------- ----------- ------------- ------------ -----------
1/1      desirable    isl           trunking     1
1/2      desirable    isl           trunking     1
Port     Vlans allowed on trunk
-------- ---------------------------------------------------------------------
1/1       1-1005
1/2       1-1005
Port     Vlans allowed and active in management domain
-------- ---------------------------------------------------------------------
1/1       1,10,20,30,40,50,60
1/2       1,10,20,30,40,50,60
Port     Vlans in spanning tree forwarding state and not pruned
-------- ---------------------------------------------------------------------
1/1
1/2
Switch_1> (enable)
```

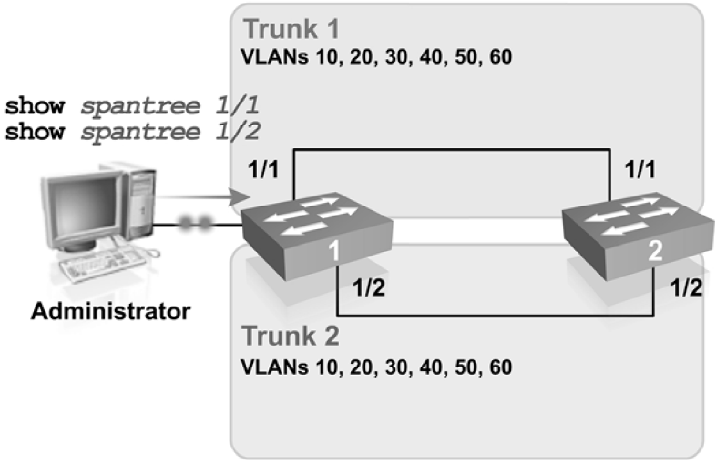
Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 6)



Step 6 Note that when the trunk links come up, VTP passes the VTP and VLAN configuration to Switch 2. Verify that Switch 2 has learned the VLAN configuration by entering the **show vlan** command on Switch 2.

```
Switch_2> (enable) show vlan
vLAN  Name                             Status     Mod/Ports, Vlans
---- -------------------------------- --------- ----------------------------
1     default                          active
10    VLAN0010                         active
20    VLAN0020                         active
30    VLAN0030                         active
40    VLAN0040                         active
50    VLAN0050                         active
60     VLAN0060                        active
1002   fddi-default                    active
1003   token-ring-default              active
1004   fddinet-default                 active
1005   trnet-default                   active
<...output truncated...>
Switch_2> (enable)
```

Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 7)

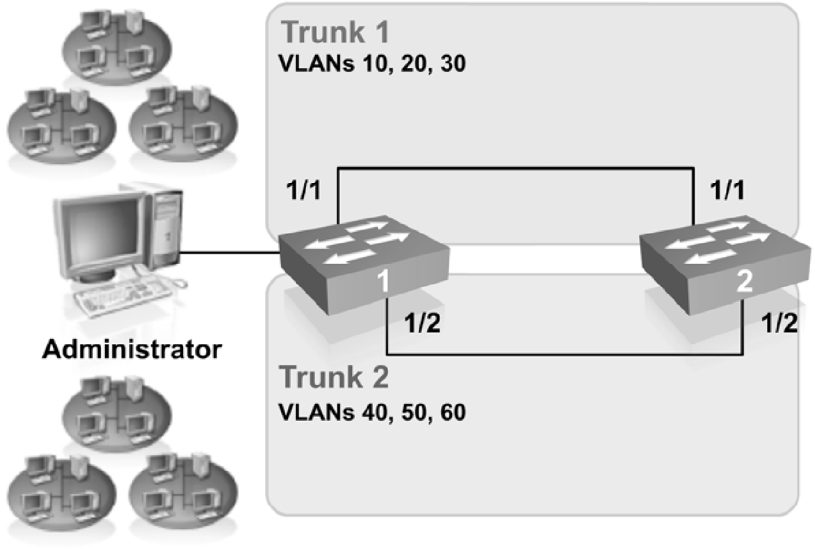


Step 7 Note that spanning tree takes one to two minutes to converge. Once the network stabilizes, check the spanning-tree state of each trunk port on Switch 1 by entering the **`show spantree`** command. Trunk 1 is forwarding for all VLANs. Trunk 2 is blocking for all VLANs. On Switch 2, both trunks are forwarding for all VLANs, but no traffic passes over Trunk 2 because port 1/2 on Switch 1 is blocking.

```
Switch_1> (enable) show spantree 1/1
Port      Vlan Port-State    Cost  Priority Fast-Start Group-method
--------- ---- ------------- ----- -------- ---------- ------------
1/1        1   forwarding    19     32       disabled
1/1        10  forwarding    19     32       disabled
1/1        20  forwarding    19     32       disabled
1/1        30  forwarding    19     32       disabled
1/1        40  forwarding    19     32       disabled
1/1        50  forwarding    19     32       disabled
1/1        60 forwarding     19     32       disabled
1/1       1003 not-connected 19     32       disabled
1/1       1005 not-connected 19      4       disabled
Switch_1> (enable) show spantree 1/2
Port      Vlan Port-State    Cost  Priority Fast-Start Group-method
--------- ---- ------------- ----- -------- ---------- ------------
1/2        1    blocking      19    32      disabled
1/2        10   blocking      19    32      disabled
1/2        20   blocking      19    32      disabled
1/2        30   blocking      19    32      disabled
```

```
1/2        40   blocking      19    32      disabled
1/2        50   blocking      19    32      disabled
1/2        60   blocking      19    32      disabled
1/2        1003 not-connected 19    32      disabled
1/2        1005 not-connected 19     4      disabled
Switch_1> (enable)
```

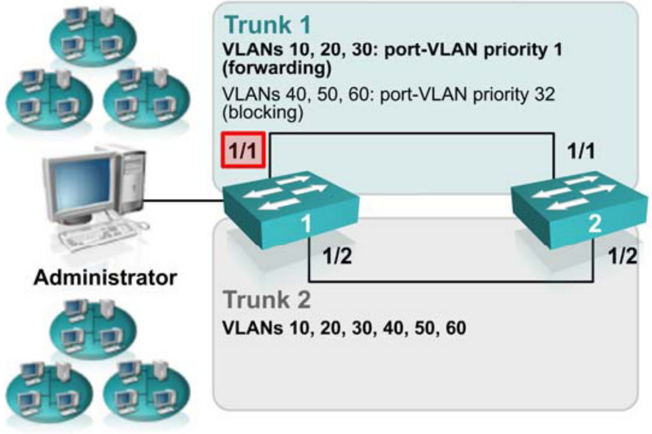
Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 8)



Step 8 Divide the configured VLANs into two groups. You might want traffic from half of the VLANs to go over one trunk link and half over the other, or if one VLAN has heavier traffic than the others, you can have traffic from that VLAN go over one trunk and traffic from the other VLANs go over the other trunk link.

In the following steps, VLANs 10, 20, and 30 (Group 1) are forwarded over Trunk 1, and VLANs 40, 50, and 60 (Group 2) are forwarded over Trunk 2.

Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 9)

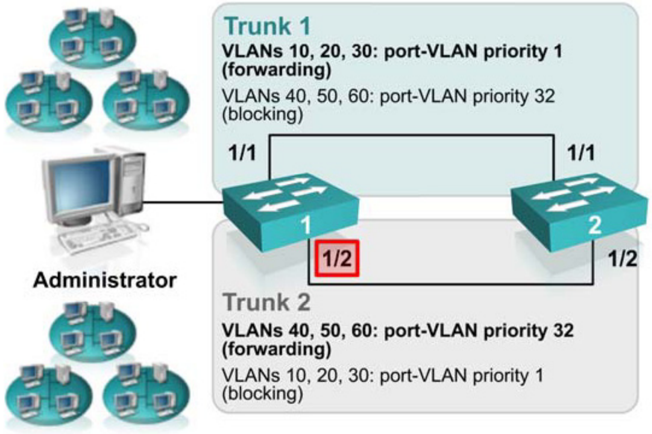

Trunk 1
VLANs 10, 20, 30: port-VLAN priority 1 (forwarding)
VLANs 40, 50, 60: port-VLAN priority 32 (blocking)
1/1
1/1
1
2
1/2
1/2
Administrator
Trunk 2
VLANs 10, 20, 30, 40, 50, 60

- **Set the port-VLAN priority on port 1/1 to a value of 1 for VLANs 40, 50, and 60**

Step 9 On Switch 1, enter the set spantree portvlanpri command to change the port-VLAN priority for the Group 1 VLANs on Trunk 1 (port 1/1) to an integer value lower than the default of 32.

```
Switch_1> (enable) set spantree portvlanpri 1/1 1 10
Port 1/1 vlans 1-9,11-1004 using portpri 32.
Port 1/1 vlans 10 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_1> (enable) set spantree portvlanpri 1/1 1 20
Port 1/1 vlans 1-9,11-19,21-1004 using portpri 32.
Port 1/1 vlans 10,20 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_1> (enable) set spantree portvlanpri 1/1 1 30
Port 1/1 vlans 1-9,11-19,21-29,31-1004 using portpri 32.
Port 1/1 vlans 10,20,30 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_1> (enable)
```

Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 10)

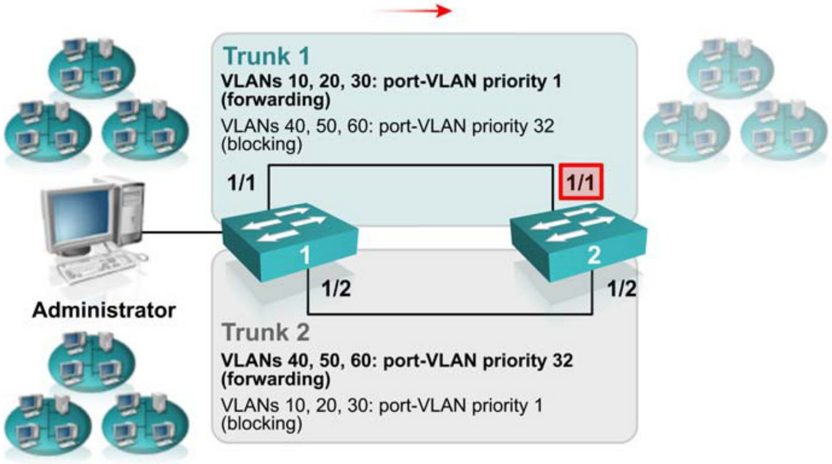

Trunk 1
VLANs 10, 20, 30: port-VLAN priority 1 (forwarding)
VLANs 40, 50, 60: port-VLAN priority 32 (blocking)
1/1
1/1
1
2
1/2
1/2
Administrator
Trunk 2
VLANs 40, 50, 60: port-VLAN priority 32 (forwarding)
VLANs 10, 20, 30: port-VLAN priority 1 (blocking)

- **Set the port-VLAN priority on port 1/2 to a value of 1 for VLANs 10, 20, and 30**

Step 10 On Switch 1, change the port-VLAN priority for the Group 2 VLANs on Trunk 2 (port 1/2) to an integer value lower than the default of 32.

```
Switch_1> (enable) set spantree portvlanpri 1/2 1 40
Port 1/2 vlans 1-39,41-1004 using portpri 32.
Port 1/2 vlans 40 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_1> (enable) set spantree portvlanpri 1/2 1 50
Port 1/2 vlans 1-39,41-49,51-1004 using portpri 32.
Port 1/2 vlans 40,50 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_1> (enable) set spantree portvlanpri 1/2 1 60
Port 1/2 vlans 1-39,41-49,51-59,61-1004 using portpri 32.
Port 1/2 vlans 40,50,60 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_1> (enable)
```

Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 11)


Trunk 1
VLANs 10, 20, 30: port-VLAN priority 1 (forwarding)
VLANs 40, 50, 60: port-VLAN priority 32 (blocking)
1/1
1/1
1
2
1/2
1/2
Administrator
Trunk 2
VLANs 40, 50, 60: port-VLAN priority 32 (forwarding)
VLANs 10, 20, 30: port-VLAN priority 1 (blocking)

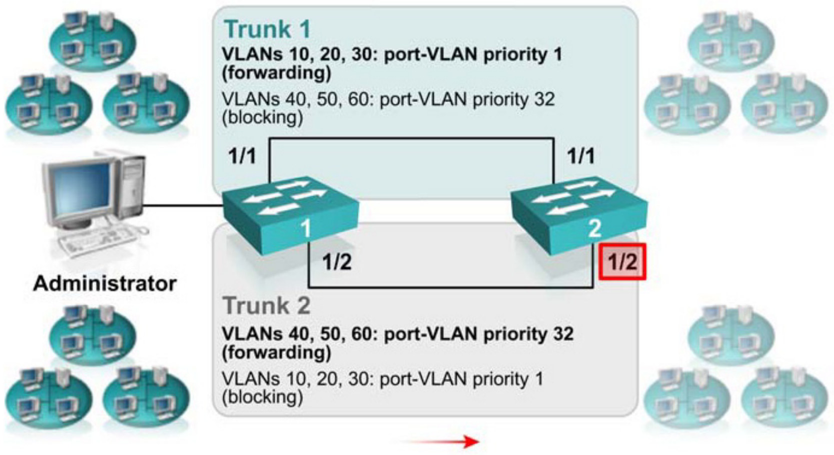
- **Set the port-VLAN priority on port 1/1 to match switch 1**

Step 11 On Switch 2, change the port-VLAN priority for the Group 1 VLANs on Trunk 1 (port 1/1) to the same value you configured for those VLANs on Switch 1.

Warning The port-VLAN priority for each VLAN must be equal on both ends of the link.

```
Switch_2> (enable) set spantree portvlanpri 1/1 1 10
Port 1/1 vlans 1-9,11-1004 using portpri 32.
Port 1/1 vlans 10 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_2> (enable)  set spantree portvlanpri 1/1 1 20
Port 1/1 vlans 1-9,11-19,21-1004 using portpri 32.
Port 1/1 vlans 10,20 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_2> (enable) set spantree portvlanpri 1/1 1 30
Port 1/1 vlans 1-9,11-19,21-29,31-1004 using portpri 32.
Port 1/1 vlans 10,20,30 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_2> (enable)
```

Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 12)

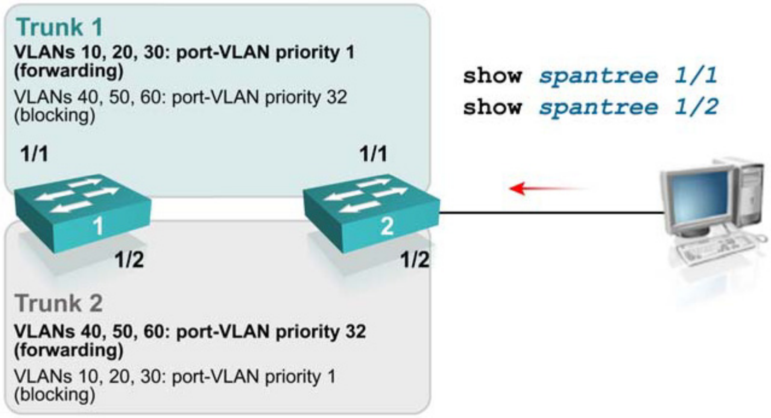

Trunk 1
VLANs 10, 20, 30: port-VLAN priority 1 (forwarding)
VLANs 40, 50, 60: port-VLAN priority 32 (blocking)
1/1
1/1
1
2
1/2
1/2
Administrator
Trunk 2
VLANs 40, 50, 60: port-VLAN priority 32 (forwarding)
VLANs 10, 20, 30: port-VLAN priority 1 (blocking)

- **Set the port-VLAN priority on port 1/2 to match switch 1**

Step 12 On Switch 2, change the port-VLAN priority for the Group 2 VLANs on Trunk 2 (port 1/2) to the same value you configured for those VLANs on Switch 1.

```
Switch_2> (enable) set spantree portvlanpri 1/2 1 40
Port 1/2 vlans 1-39,41-1004 using portpri 32.
Port 1/2 vlans 40 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_2> (enable) set spantree portvlanpri 1/2 1 50
Port 1/2 vlans 1-39,41-49,51-1004 using portpri 32.
Port 1/2 vlans 40,50 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_2> (enable) set spantree portvlanpri 1/2 1 60
Port 1/2 vlans 1-39,41-49,51-59,61-1004 using portpri 32.
Port 1/2 vlans 40,50,60 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_2> (enable)
```

Example: Load-Sharing VLAN Traffic Over Parallel Trunks (Step 13)


Trunk 1
VLANs 10, 20, 30: port-VLAN priority 1 (forwarding)
VLANs 40, 50, 60: port-VLAN priority 32 (blocking)
1/1
1/1
1
2
1/2
1/2
Trunk 2
VLANs 40, 50, 60: port-VLAN priority 32 (forwarding)
VLANs 10, 20, 30: port-VLAN priority 1 (blocking)

```
show spantree 1/1
show spantree 1/2
```

- **Set the port-VLAN priority on port 1/2 to match switch 1**

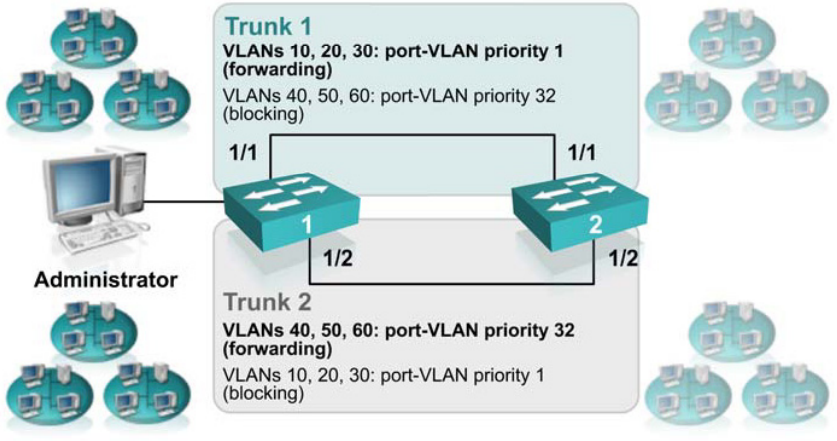
Step 13 Note that when you have configured the port-VLAN priorities on both ends of the link, the spanning tree converges to use the new configuration.

Check the spanning-tree port states on Switch 1 by entering the **show spantree** command. The Group 1 VLANs should be forwarding on Trunk 1 and blocking on Trunk 2. The Group 2 VLANs should be blocking on Trunk 1 and forwarding on Trunk 2.

```
Switch_1> (enable) show spantree 1/1
Port      Vlan Port-State    Cost  Priority Fast-Start Group-method
--------- ---- ------------- ----- -------- ---------- ------------
1/1        1    forwarding    19    32       disabled
1/1        10   forwarding    19     1       disabled
1/1        20   forwarding    19     1       disabled
1/1        30   forwarding    19     1       disabled
1/1        40   blocking      19    32       disabled
1/1        50   blocking      19    32       disabled
1/1        60   blocking      19    32       disabled
1/1       1003 not-connected  19    32       disabled
1/1       1005 not-connected  19     4       disabled
Switch_1> (enable) show spantree 1/2
Port      Vlan Port-State    Cost  Priority Fast-Start Group-method
--------- ---- ------------- ----- -------- ---------- ------------
1/2       1    blocking       19    32       disabled
1/2       10   blocking       19    32       disabled
1/2       20   blocking       19    32       disabled
1/2       30   blocking       19    32       disabled
```

```
1/2       40   forwarding     19     1       disabled
1/2       50   forwarding     19     1       disabled
1/2       60   forwarding     19     1       disabled
1/2       1003 not-connected  19    32       disabled
1/2       1005 not-connected  19     4       disabled
Switch_1> (enable)
```

Example: Load-Sharing VLAN Traffic Over Parallel Trunks


Trunk 1
VLANs 10, 20, 30: port-VLAN priority 1 (forwarding)
VLANs 40, 50, 60: port-VLAN priority 32 (blocking)
1/1
1/1
1
2
1/2
1/2
Administrator
Trunk 2
VLANs 40, 50, 60: port-VLAN priority 32 (forwarding)
VLANs 10, 20, 30: port-VLAN priority 1 (blocking)

The above figure shows that both trunks are utilized when the network is operating normally and, if one trunk link fails, the other trunk link acts as an alternate forwarding path for the traffic previously traveling over the failed link.

Lesson Summary

This lesson accomplished the following:

- Explained the purpose of VLANs
- Described the configuration of VLANs in both Hybrid and Native modes
- Explained the need advantages of trunking
- Described the configuration of trunking in both Hybrid and Native modes

Next Steps

After completing this lesson, go to:

- Layer 3 Services

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_6_3/confg_gd/vlans.htm

summary

- Demonstrated the configuration of an IP address for a Catalyst 6500's management interface
- Demonstrated how to enable the High Availability feature
- Detailed how to create an EtherChannel (in both Hybrid and Native modes)
- Explained how to configure VLANs and trunks (in both Hybrid and Native modes)
- Discussed how to optimize Spanning Tree Performance using PortFast, UplinkFast, and BackboneFast

Review Questions

Q1) What is the maximum number of ports that can be used in a single Catalyst 6500 EtherChannel?

A) 2

B) 4

C) 6

D) 8

Answer: D

Q2) Which of the following are trunking protocols supported by the Catalyst 6500?

A) IEEE 802.1d

B) IEEE 802.1q

C) ISL

D) VTP

Answers: B and C

Q3) Which of the following supports the propagation of VLAN information across trunk links?

A) VTP

B) ISL

C) DTP

D) PAgP

Answer: A

Q4) Which STP optimization feature should only be used on wiring closet switches?

A) UplinkFast

B) BackboneFast

C) BPDUFast

D) AccessFat

Answer: A

Q5) If STP parameters are left at their default values, what determines which switch will become the Root Bridge?

A) Bridge Priority

B) MAC Address

C) Port Count

D) Port Number

Answer: B

Configuring Layer 3 Services

Overview

This module covers the Layer 3 capabilities of the Catalyst 6500. This module also details the configuration of those Layer 3 features.

Outline

- The module contains these lessons:
- InterVLAN Routing
- Multilayer Switching
- Layer 3 Redundancy
- Multicast Routing

InterVLAN Routing

Overview

This lesson discusses the concept of InterVLAN Routing. VLANs are used to segment networks into different broadcast domains. In order for traffic to pass from one VLAN to another, a Layer 3 router is required. On the Catalyst 6500 this Layer 3 router will be the MSFC. This lesson will detail the configuration of the MSFC for InterVLAN routing.

Importance

Most networks are segmented into many VLANs. The MSFC configured for InterVLAN routing is required into order for users in these different VLANs to communicate with each other.

Objectives

Upon completing this lesson, you will be able to:

- Describe InterVLAN routing
- Describe the need for InterVLAN routing
- Access the MSFC and configure InterVLAN routing for IP, IPX, and AppleTalk

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Have a basic knowledge of Layer 2 switching and concepts such as VLANs

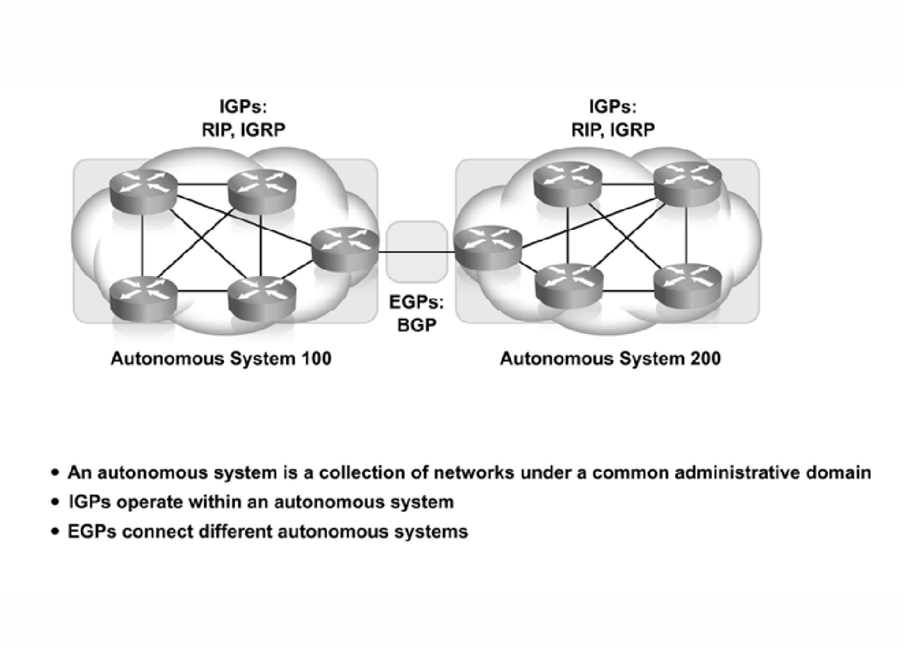
Outline

This lesson includes these sections:

- Overview
- Understanding How InterVLAN Routing Works
- MSFC Routing Configuration Guidelines
- Summary

Understanding How InterVLAN Routing Works

Before you can configure InterVLAN routing, you must understand how it works. In order for two VLANs to communicate, a Layer 3 device, such as a router, is required.



- An autonomous system is a collection of networks under a common administrative domain
- IGPs operate within an autonomous system
- EGPs connect different autonomous systems

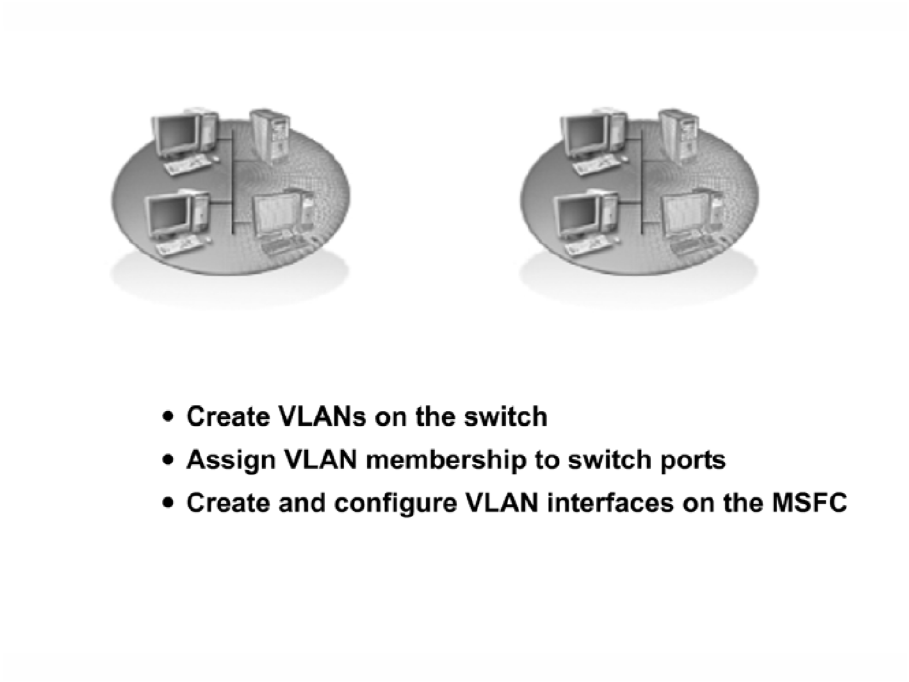
VLANs allow you to group ports on a switch to limit unicast, multicast, and broadcast traffic flooding. Flooded traffic originating from a particular VLAN is only flooded out ports belonging to that VLAN. Network devices in different VLANs cannot communicate with one another without a router to forward traffic between the VLANs. In most network environments, VLANs are associated with individual networks or subnetworks.

For example, in an IP network, each subnetwork is mapped to an individual VLAN. In an IPX network, each VLAN is mapped to an IPX network number.

Configuring VLANs helps control the size of the broadcast domain and keeps local traffic local. When an end station in one VLAN needs to communicate with an end station in another VLAN, interVLAN communication is required. This communication is provided by interVLAN routing. You configure one or more routers to route traffic to the appropriate destination VLAN.

MSFC Routing Configuration Guidelines

On the Catalyst 6500, the Layer 3 device is the MSFC. Therefore, InterVLAN routing is configured on the MSFC.



- **Create VLANs on the switch**
- **Assign VLAN membership to switch ports**
- **Create and configure VLAN interfaces on the MSFC**

InterVLAN routing can be configured using either an external router that supports ISL trunking or an internal MSFC. The former term is commonly referred as a Router on a Stick. Configuring interVLAN routing on the MSFC consists of two main procedures:

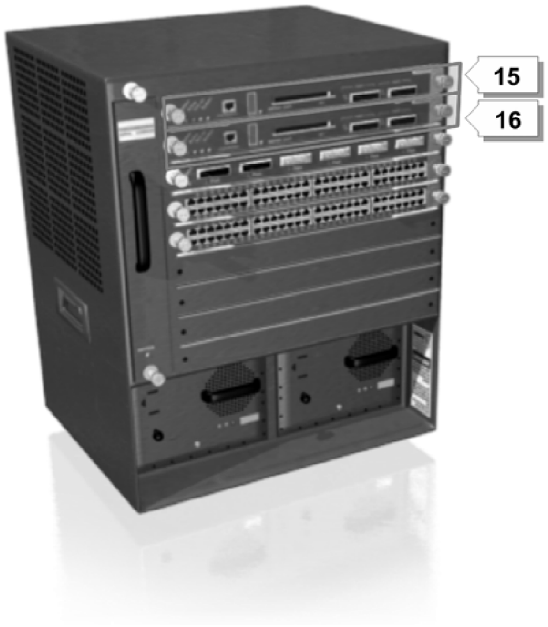
1. Create and configure VLANs on the switch and assign VLAN membership to switch ports.
2. Create and configure VLAN interfaces for interVLAN routing on the MSFC. Configure a VLAN interface for each VLAN for which you want to route traffic.

VLAN interfaces on the MSFC are virtual interfaces. However, you configure them much as you do a physical router interface.

MSFC2 and MSFC support the same range of VLANs as the Supervisor Engine. MSFC2 supports up to 1,000 VLAN interfaces. MSFC supports up to 256 VLAN interfaces.

Accessing the MSFC

SWITCH CONSOLE NUMBER OR SESSION NUMBER



Accessing the MSFC from the Console Port

You can enter the **`switch console`** command to access the MSFC from the switch CLI directly connected to the Supervisor Engine console port. To exit from the MSFC CLI and return to the switch CLI, press **`Ctrl-C`** three times at the **`Router>`** prompt.

To access the MSFC from the switch CLI, perform this task:

Table 3-1: Accessing the MSFC via a Console Connection

Task	Command
Access the MSFC from the switch CLI.	**`switch console`** `[`*`mod`*`]`[1]

[1] The mod keyword specifies the module number of the MSFC, either 15 (if the MSFC is installed on the Supervisor Engine in slot 1) or 16 (if the MSFC is installed on the Supervisor Engine in slot 2). If no module number is specified, the console will switch to the MSFC on the active Supervisor Engine.

Note To access the Cisco IOS CLI on the standby MSFC, connect to the console port of the standby Supervisor Engine.

This example shows how to access the active MSFC from the switch CLI from the active Supervisor Engine, and how to exit the MSFC CLI and return to the switch CLI:

```
Console> (enable) switch console 15
Trying Router-15...
Connected to Router-15.
Type ^C^C^C to switch back...
```

```
Router> ^C^C^C
Console> (enable)
```

Accessing the MSFC from a Telnet Session

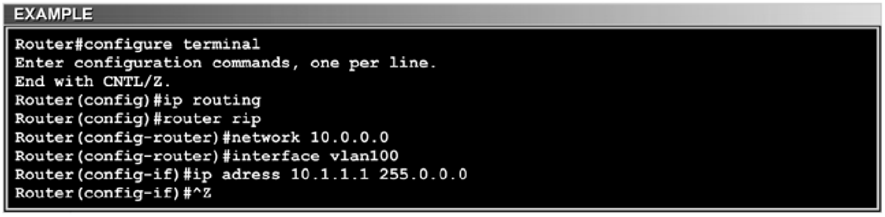
You can enter the **`session`** *mod* command to access the MSFC from the switch CLI using a Telnet session. To exit from the MSFC CLI back to the switch CLI, enter the **`exit`** command at the **`Router>`** prompt.

Note The Supervisor Engine software sees the MSFC as module 15 (when installed on a Supervisor Engine in slot 1) or module 16 (when installed on a Supervisor Engine in slot 2).

This example shows how to access the MSFC from the switch CLI, and how to exit the MSFC CLI and return to the switch CLI:

```
Console> (enable) session 15
Router> exit
Console> (enable)
```

Configuring IP InterVLAN Routing on the MSFC



```
Router#configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
Router(config)#ip routing
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#interface vlan100
Router(config-if)#ip adress 10.1.1.1 255.0.0.0
Router(config-if)#^Z
```

To configure interVLAN routing for IP, perform this task:

Table 3-2: InterVLAN Routing with IP

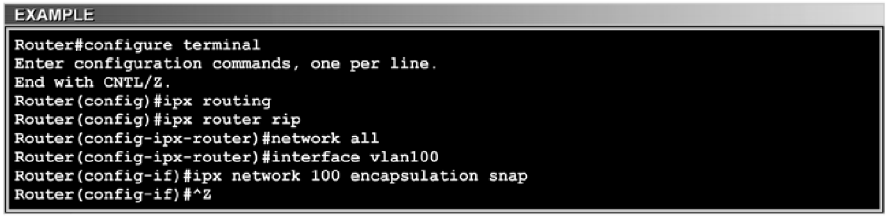
Task	Command
Step 1: (Optional) Enable IP routing on the router[1].	**MSFC(config)# ip routing**
Step 2: (Optional) Specify an IP routing protocol[2].	**MSFC(config)# router *ip_routing_protocol***
Step 3: Specify a VLAN interface on the MSFC.	**MSFC(config)# interface *vlan-id***
Step 4: Assign an IP address to the VLAN.	**MSFC(config-if)# ip address *n.n.n.n mask***
Step 5: Exit configuration mode.	**MSFC(config-if)# Ctrl-Z**

[1] This step is necessary if you have multiple routers in the network.
[2] This step is necessary if you enabled IP routing in Step 1. This step might include other commands, such as using the **network** router configuration command to specify the networks to route. Refer to the documentation for your router platform for detailed information on configuring routing protocols.

This example shows how to enable IP routing on the MSFC, create a VLAN interface, and assign the interface an IP address:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip routing
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# interface vlan 100
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# end
Router#
```

Configuring IPX InterVLAN Routing on the MSFC



```
Router#configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
Router(config)#ipx routing
Router(config)#ipx router rip
Router(config-ipx-router)#network all
Router(config-ipx-router)#interface vlan100
Router(config-if)#ipx network 100 encapsulation snap
Router(config-if)#^Z
```

To configure interVLAN routing for Internetwork Packet Exchange (IPX), perform this task:

Table 3-3: InterVLAN Routing with IPX

Task	Command
Step 1: (Optional) Enable IPX routing on the router[1].	`Router(config)#` **`ipx routing`**
Step 2: (Optional) Specify an IPX routing protocol[2].	`Router(config)#` **`ipx router`** *`ipx_routing_protocol`*
Step 3: Specify a VLAN interface on the MSFC.	`Router(config)#` **`interface`** *`vlan-id`*
Step 4: Assign a network number to the VLAN[3].	`Router(config-if)#` **`ipx network`** `[`*`network`* | **`unnumbered`**`]` **`encapsulation`** *`encapsulation-type`*
Step 5: Exit configuration mode.	`Router(config-if)#` **`Ctrl-Z`**

[1] This step is necessary if you have multiple routers in the network.

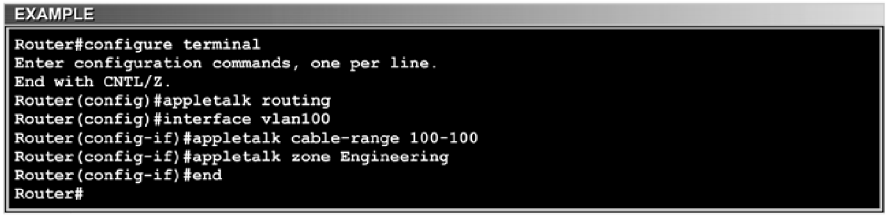
[2] This step is necessary if you enabled IPX routing in Step 1. This step might include other commands, such as using the **network** router configuration command to specify the networks to route. Refer to the documentation for your router platform for detailed information on configuring routing protocols.

[3] This enables IPX routing on the VLAN. When you enable IPX routing on the VLAN, you can also specify an encapsulation type.

This example shows how to enable IPX routing on the MSFC, create a VLAN interface, and assign the interface an IPX network address:

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ipx routing
Router(config)# ipx router rip
Router(config-ipx-router)# network all
Router(config-ipx-router)# interface vlan100
Router(config-if)# ipx network 100 encapsulation snap
Router(config-if)# end
Router#
```

Configuring AppleTalk InterVLAN Routing on the MSFC



EXAMPLE

```
Router#configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
Router(config)#appletalk routing
Router(config)#interface vlan100
Router(config-if)#appletalk cable-range 100-100
Router(config-if)#appletalk zone Engineering
Router(config-if)#end
Router#
```

To configure interVLAN routing for AppleTalk, perform this task:

Table 3-4: InterVLAN Routing with AppleTalk

Task	Command
Step 1: (Optional) Enable AppleTalk routing on the router[1].	Router(config)# **appletalk routing**
Step 2: Specify a VLAN interface on the MSFC.	Router(config)# **interface** *vlan-id*
Step 3: Assign a cable range to the VLAN.	Router(config-if)# **appletalk cable-range** *cable-range*
Step 4: Assign a zone name to the VLAN.	Router(config-if)# **appletalk zone** *zone-name*
Step 5: Exit configuration mode.	Router(config-if)# **Ctrl-Z**

[1] This step is necessary if you have multiple routers in the network.

This example shows how to enable AppleTalk routing on the MSFC, create a VLAN interface, and assign the interface an AppleTalk cable-range and zone name:

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# appletalk routing
Router(config)# interface vlan100
Router(config-if)# appletalk cable-range 100-100
Router(config-if)# appletalk zone Engineering
Router(config-if)# end
Router#
```

Lesson Summary

This lesson accomplished the following:

- Described InterVLAN routing
- Described the need for InterVLAN routing
- Accessed the MSFC and configured InterVLAN routing for IP, IPX, and AppleTalk

Next Steps

After completing this lesson, go to:

- Multilayer Switching

References

For additional information, refer to these resources:

- Configuring InterVLAN Routing Using an Internal Routing (Layer 3 Card) on the Catalyst 5000 and 6000 Switches Running Cat OS: http://www.cisco.com/warp/public/473/75.html

Multilayer Switching (MLS)

Overview

Multilayer Switching (MLS) is hardware-based routing. In particular, the packet forwarding is handled by specialized hardware, usually ASICs. Depending on the protocols, interfaces, and features supported, Layer 3 switches can be used in place of routers in a campus design.

Importance

Multilayer Switching provides high-performance Layer 3 switching for Cisco routers and switches.

Objectives

Upon completing this lesson, you will be able to:

- Describe Multilayer Switching (MLS)
- List the advantages of Multilayer Switching vs. traditional Layer 2 switching and Layer 3 routing
- Explain and list the different Flow Masks involved with Multilayer Switching
- Configure Multilayer Switching on the Catalyst 6500

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Have a basic understanding of Layer 2 switching and Layer 3 routing

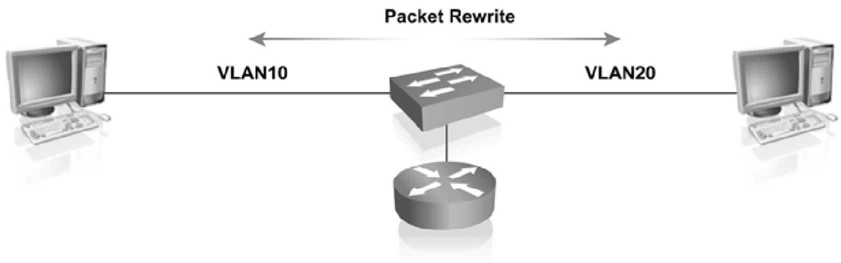
Outline

This lesson includes these sections:

- Overview
- Understanding How Layer 3 Switching Works
- Understanding Flow Masks
- MLS Examples
- Configuring MLS
- Summary

Understanding How Layer 3 Switching Works

Before you can configure MLS, you must an understanding of how Layer 3 switching works and what advantages it has over traditional Layer 2 switching.



Layer 3 switching allows the switch, instead of a router, to forward IP and IPX unicast traffic and IP multicast traffic between VLANs. Layer 3 switching is implemented in hardware and provides wire-speed interVLAN forwarding on the switch, rather than on the MSFC. Layer 3 switching requires minimal support from the MSFC. The MSFC routes any traffic that cannot be Layer 3 switched.

Note Layer 3 switching supports the routing protocols configured on the MSFC. Layer 3 switching does not replace the routing protocols configured on the MSFC. Layer 3 switching uses IP Protocol Independent Multicast (IP PIM) for multicast route determination.

Layer 3 switching on Catalyst 6000 family switches provides traffic statistics that you can use to identify traffic characteristics for administration, planning, and troubleshooting. Layer 3 switching uses NetFlow Data Export (NDE) to export flow statistics.

When a packet is Layer 3 switched from a source in one VLAN to a destination in another VLAN, the switch performs a packet rewrite at the egress port based on information learned from the MSFC so that the packets appear to have been routed by the MSFC.

Note Rather than just forwarding multicast packets, the switch replicates them as necessary on the appropriate VLANs.

Packet rewrite alters five fields:

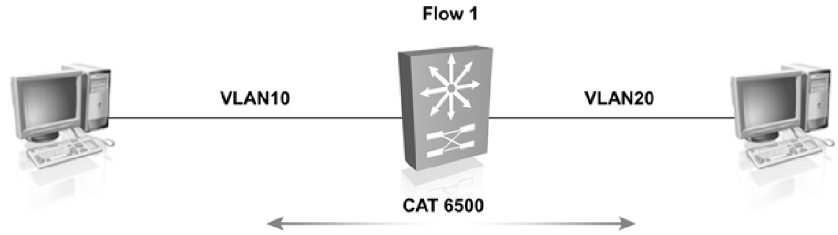
- Layer 2 (MAC) destination address
- Layer 2 (MAC) source address
- Layer 3 IP Time to Live (TTL) or IPX Transport Control
- Layer 3 checksum
- Layer 2 (MAC) checksum (also called the frame checksum or FCS)

If Source A and Destination B are on different VLANs and Source A sends a packet to the MSFC to be routed to Destination B, the switch recognizes that the packet was sent to the Layer 2 (MAC) address of the MSFC.

To perform Layer 3 switching the switch rewrites the Layer 2 frame header, changing the Layer 2 destination address to the Layer 2 address of Destination B, and the Layer 2 source address to the Layer 2 address of the MSFC. The Layer 3 addresses remain the same.

In IP unicast and IP multicast traffic, the switch decrements the Layer 3 Time to Live (TTL) value by 1 and recomputes the Layer 3 packet checksum. In IPX traffic, the switch increments the Layer 3 Transport Control value by 1 and recomputes the Layer 3 packet checksum. The switch recomputes the Layer 2 frame checksum and forwards (or, for multicast packets, replicates as necessary) the rewritten packet to Destination B's VLAN.

Understanding MLS



Note Supervisor Engine 1, PFC, and MSFC or MSFC2 can only do MLS internally with the MSFC or MSFC2 in the same chassis; an external MLS-RP cannot be used in place of the internal MLS-RP.

Supervisor Engine 1, PFC, and MSFC or MSFC2 provide Layer 3 switching with MLS, which identifies flows on the switch after the first packet has been routed by the MSFC and transfers the process of forwarding the remaining traffic in the flow to the switch, which reduces the load on the MSFC.

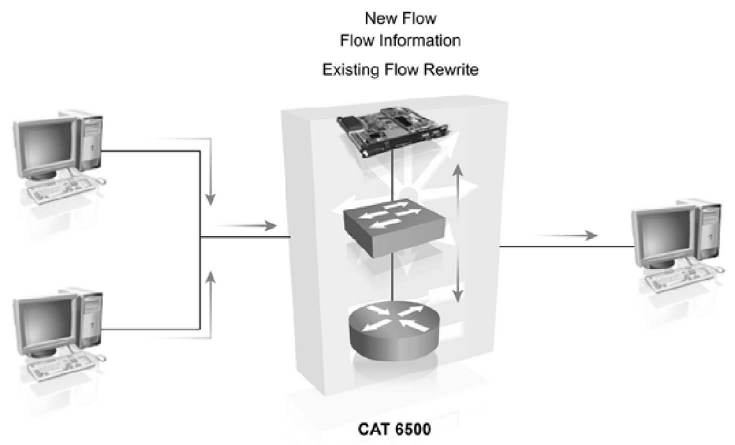
Understanding MLS Flows

Layer 3 protocols, such as IP and IPX, are connectionless—they deliver every packet independently of every other packet. However, actual network traffic consists of many end-to-end conversations, or flows, between users or applications.

- MLS supports unicast and multicast flows. A unicast flow can be any of the following:
  - All traffic to a particular destination
  - All traffic from a particular source to a particular destination
- All traffic from a particular source to a particular destination that shares the same protocol and transport-layer information.
- A multicast flow is all traffic with the same protocol and transport-layer information from a particular source to the members of a particular destination multicast group.

For example, communication from a client to a server, and from the server to the client, are separate flows. Telnet traffic transferred from a particular source to a particular destination comprises a separate flow from File Transfer Protocol (FTP) packets between the same source and destination.

Understanding the MLS Cache



MLS Cache

The PFC maintains a Layer 3 switching table (the MLS cache) for Layer 3-switched flows. The cache also includes entries for traffic statistics that are updated in tandem with the switching of packets. After the PFC creates an MLS cache entry, packets identified as belonging to an existing flow can be Layer 3 switched based on the cached information. The MLS cache maintains flow information for all active flows.

Unicast Traffic

For unicast traffic, the PFC creates an MLS cache entry for the initial routed packet of each unicast flow. Upon receipt of a routed packet that does not match any unicast flow currently in the MLS cache, the PFC creates a new MLS entry.

Multicast Traffic

For multicast traffic, the PFC populates the MLS cache using information learned from the MSFC. Whenever the MSFC receives traffic for a new multicast flow, it updates its multicast routing table and forwards the new information to the PFC. In addition, if an entry in the multicast routing table ages out, the MSFC deletes the entry and forwards the updated information to the PFC.

For each multicast flow cache entry, the PFC maintains a list of outgoing interfaces for the destination IP multicast group. The PFC uses this list to identify the VLANs on which traffic to a given multicast flow should be replicated.

These MSFC IOS commands affect the multicast MLS cache entries on the switch:

Using the clear ip mroute command to clear the multicast routing table on the MSFC clears all multicast.

MLS cache entries on the PFC

Using the `no ip multicast-routing` command to disable IP multicast routing on the MSFC purges all multicast MLS cache entries on the PFC.

MLS Cache Aging

The state and identity of flows are maintained while packet traffic is active; when traffic for a flow ceases, the entry ages out. You can configure the aging time for MLS entries kept in the MLS cache. If an entry is not used for the specified period of time, the entry ages out, and statistics for that flow can be exported to a flow collector application.

MLS Cache Size

The maximum MLS cache size is 128K entries. The MLS cache is shared by all MLS processes on the switch (IP MLS, IP MMLS, and IPX MLS). An MLS cache larger than 32K entries increases the probability that a flow will not be Layer 3 switched, but will instead be forwarded to the MSFC.

Understanding Flow Masks

The PFC uses flow masks to determine how MLS entries are created.

- **destination-ip**
- **destination-ipx**
- **source-destination-ip**
- **source-destination-vlan**

Flow Mask Modes

The PFC supports only one flow mask (the most specific one) for all MSFCs that are Layer 3 switched by that PFC. If the PFC detects different flow masks from different MSFCs for which it is performing Layer 3 switching, it changes its flow mask to the most specific flow mask detected.

When the PFC flow mask changes the entire MLS cache is purged. When a PFC exports cached entries, flow records are created based on the current flow mask. Depending on the current flow mask, some fields in the flow record might not have values. Unsupported fields are filled with a zero (0).

The MLS flow masks are:

- **destination-ip:** The least-specific flow mask. The PFC maintains one MLS entry for each Layer 3 destination address. All flows to a given Layer 3 destination address use this MLS entry.

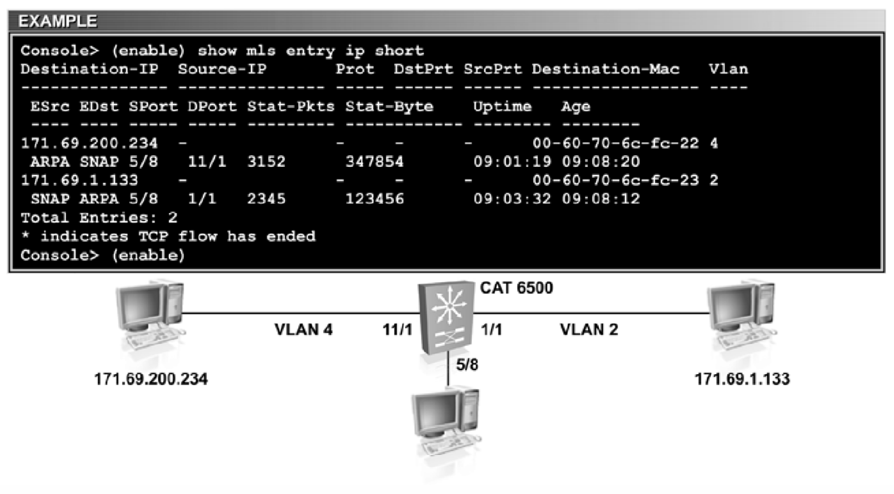
- **destination-ipx:** The only flow mask mode for IPX MLS is destination mode. The PFC maintains one IPX MLS entry for each destination IPX address (network and node). All flows to a given destination IPX address use this IPX MLS entry.

- **source-destination-ip:** The PFC maintains one MLS entry for each source and destination IP address pair. All flows between a given source and destination use this MLS entry regardless of the IP protocol ports.

- **source-destination-vlan:** For IP MMLS. The PFC maintains one MMLS cache entry for each {source IP, destination group IP, source VLAN}. The multicast source-destination-vlan flow mask differs from the IP unicast MLS source-destination-ip flow mask in that, for IP MMLS, the source VLAN is included as part of the entry. The source VLAN is the multicast reverse path forwarding (RPF) interface for the multicast flow.

- **full flow:** The most-specific flow mask. The PFC creates and maintains a separate MLS cache entry for each IP flow. A full flow entry includes the source IP address, destination IP address, protocol, and protocol ports.

Flow Mask Mode and show mls entry Command Output



```
Console> (enable) show mls entry ip short
Destination-IP  Source-IP       Prot  DstPrt SrcPrt Destination-Mac   Vlan
--------------- --------------- ----- ------ ------ ----------------- ----
 ESrc EDst SPort DPort Stat-Pkts Stat-Byte    Uptime   Age
 ---- ---- ----- ----- --------- ------------ -------- --------
171.69.200.234  -               -     -      -      00-60-70-6c-fc-22 4
 ARPA SNAP 5/8   11/1  3152      347854       09:01:19 09:08:20
171.69.1.133    -               -     -      -      00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345      123456       09:03:32 09:08:12
Total Entries: 2
* indicates TCP flow has ended
Console> (enable)
```

Destination IP Flow Mask Example

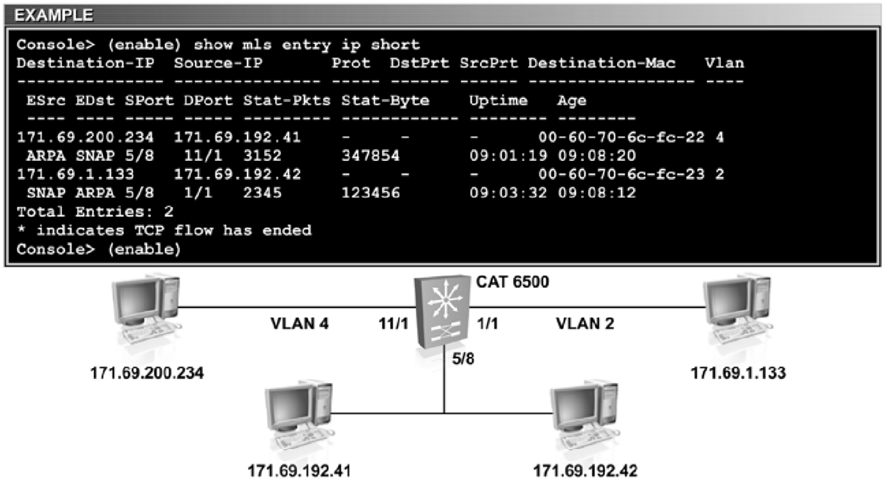
With the destination-ip flow mask, the source IP, protocol, and source and destination port fields show the details of the last packet that was Layer-3 switched using the MLS cache entry.

This example shows how the **`show mls entry`** command output appears in destination-ip mode:

```
Console> (enable) show mls entry ip short

Destination-IP   Source-IP       Prot  DstPrt SrcPrt Destination-Mac   Vlan
--------------- --------------- ----- ------ ------ ----------------- ----
 ESrc EDst  SPort DPort Stat-Pkts Stat-Byte    Uptime   Age
 ---- ---- ----- ----- --------- ------------ -------- --------
171.69.200.234  -               -     -      -      00-60-70-6c-fc-22 4
 ARPA SNAP 5/8   11/1  3152      347854       09:01:19 09:08:20
171.69.1.133    -               -     -      -      00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345      123456       09:03:32 09:08:12

Total Entries: 2
* indicates TCP flow has ended
Console> (enable)
```

Note The **`short`** keyword exists for some **`show`** commands and displays the output by wrapping the text after 80 characters. The default is **`long`** (no text wrap).

Source Destination IP Flow Mask Example



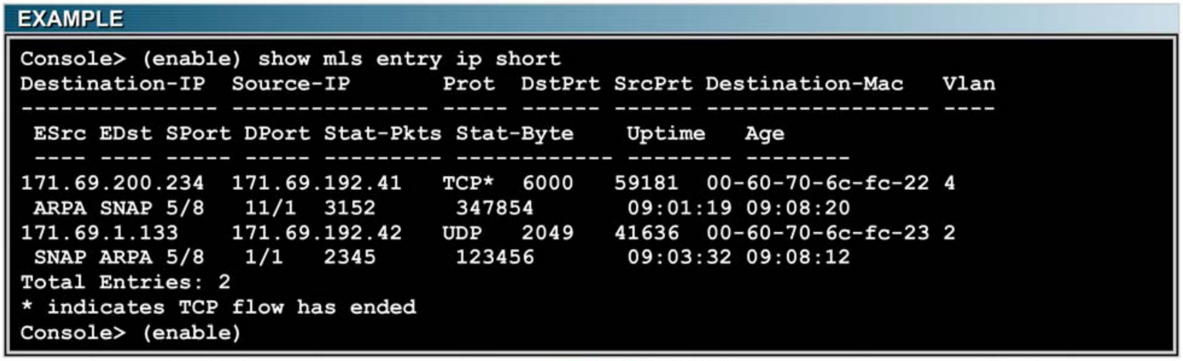
```
Console> (enable) show mls entry ip short
Destination-IP  Source-IP       Prot  DstPrt SrcPrt Destination-Mac   Vlan
--------------- --------------- ----- ------ ------ ----------------- ----
 ESrc EDst SPort DPort Stat-Pkts Stat-Byte    Uptime   Age
 ---- ---- ----- ----- --------- ------------ -------- --------
171.69.200.234  171.69.192.41    -     -      -      00-60-70-6c-fc-22 4
 ARPA SNAP 5/8   11/1  3152      347854       09:01:19 09:08:20
171.69.1.133    171.69.192.42    -     -      -      00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345      123456       09:03:32 09:08:12
Total Entries: 2
* indicates TCP flow has ended
Console> (enable)
```

With the source-destination-ip flow mask, the protocol, source port, and destination port fields display the details of the last packet that was Layer 3 switched using the MLS cache entry.

This example shows how the **show mls entry** command output appears in source-destination-ip mode:

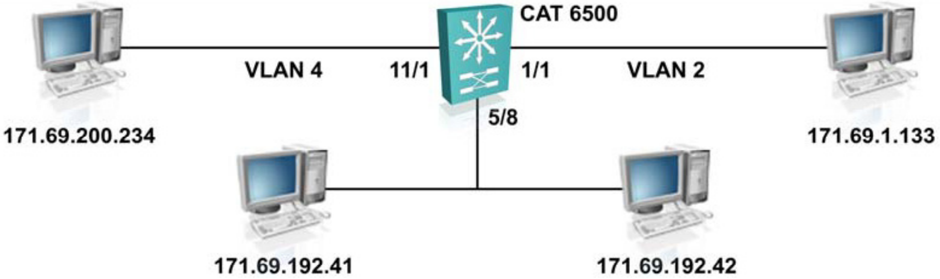
```
Console> (enable) show mls entry ip short

Destination-IP   Source-IP       Prot  DstPrt SrcPrt Destination-Mac   Vlan
--------------- --------------- ----- ------ ------ ----------------- ----
 ESrc EDst SPort DPort Stat-Pkts Stat-Byte    Uptime   Age
 ---- ---- ----- ----- --------- ------------ -------- --------
171.69.200.234  171.69.192.41   -     -      -      00-60-70-6c-fc-22 4
 ARPA SNAP 5/8   11/1  3152      347854       09:01:19 09:08:20
171.69.1.133    171.69.192.42   -     -      -      00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345      123456       09:03:32 09:08:12

Total Entries: 2
* indicates TCP flow has ended
Console> (enable)
```

Full Flow Mask Example



EXAMPLE

```
Console> (enable) show mls entry ip short
Destination-IP   Source-IP       Prot  DstPrt SrcPrt Destination-Mac   Vlan
--------------- --------------- ----- ------ ------ ----------------- ----
 ESrc EDst SPort DPort Stat-Pkts Stat-Byte    Uptime   Age
 ---- ---- ----- ----- --------- ------------ -------- --------
171.69.200.234  171.69.192.41   TCP*  6000   59181  00-60-70-6c-fc-22 4
 ARPA SNAP 5/8   11/1  3152      347854       09:01:19 09:08:20
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345      123456       09:03:32 09:08:12
Total Entries: 2
* indicates TCP flow has ended
Console> (enable)
```

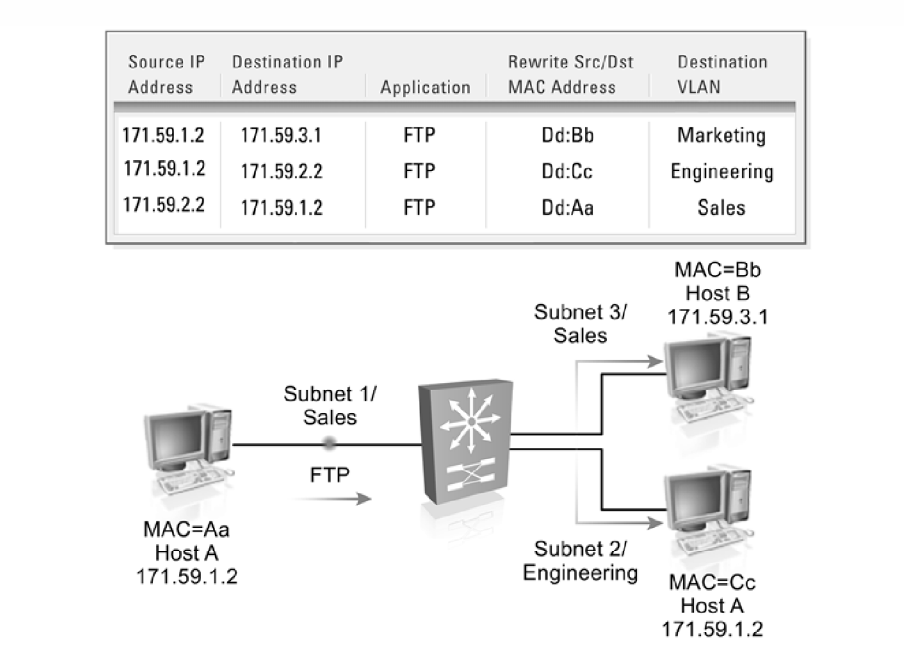


With the full-flow flow mask, because a separate MLS entry is created for every ip flow, details are shown for each flow.

This example shows how the **show mls entry** command output appears in full flow mode:

```
Console> (enable) show mls entry ip short

Destination-IP   Source-IP       Prot  DstPrt SrcPrt Destination-Mac   Vlan
--------------- --------------- ----- ------ ------ ----------------- ----
 ESrc EDst SPort DPort Stat-Pkts Stat-Byte    Uptime   Age
 ---- ---- ----- ----- --------- ------------ -------- --------
171.69.200.234  171.69.192.41   TCP*  6000   59181  00-60-70-6c-fc-22 4
 ARPA SNAP 5/8   11/1  3152      347854       09:01:19 09:08:20
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345      123456       09:03:32 09:08:12

Total Entries: 2
* indicates TCP flow has ended
Console> (enable)
```

IP MLS Example

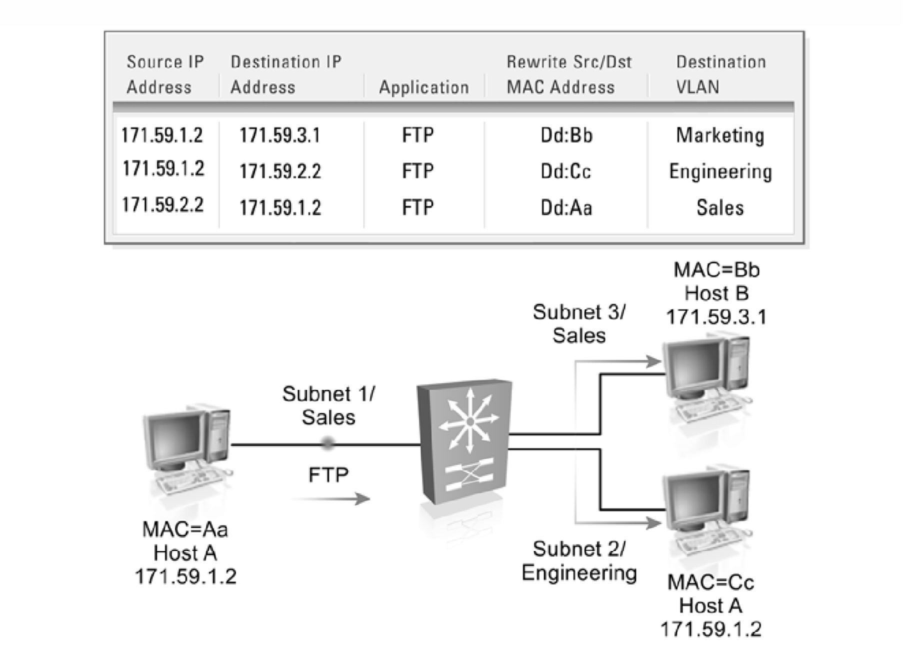


Source IP Address	Destination IP Address	Application	Rewrite Src/Dst MAC Address	Destination VLAN
171.59.1.2	171.59.3.1	FTP	Dd:Bb	Marketing
171.59.1.2	171.59.2.2	FTP	Dd:Cc	Engineering
171.59.2.2	171.59.1.2	FTP	Dd:Aa	Sales

The figure above shows a simple IP MLS network topology. In this example, Host A is on the Sales VLAN (IP subnet 171.59.1.0), Host B is on the Marketing VLAN (IP subnet 171.59.3.0), and Host C is on the Engineering VLAN (IP subnet 171.59.2.0).

When Host A initiates an HTTP file transfer to Host C, an MLS entry for this flow is created (this entry is the second item in the MLS cache shown in the Figure). The PFC stores the MAC addresses of the MSFC and Host C in the MLS entry when the MSFC forwards the first packet from Host A through the switch to Host C. The PFC uses this information to rewrite subsequent packets from Host A to Host C.

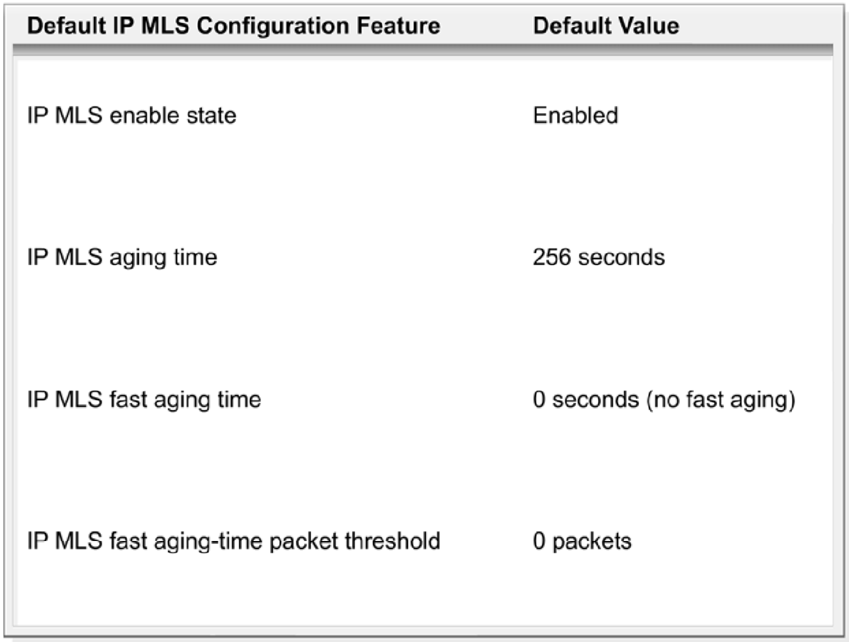
IP MLS Example Topology



Source IP Address	Destination IP Address	Application	Rewrite Src/Dst MAC Address	Destination VLAN
171.59.1.2	171.59.3.1	FTP	Dd:Bb	Marketing
171.59.1.2	171.59.2.2	FTP	Dd:Cc	Engineering
171.59.2.2	171.59.1.2	FTP	Dd:Aa	Sales

With traditional routing, when the router receives a packet, it strips off the Layer 2 header information (i.e., the MAC address), because the destination MAC address for a packet entering the router is the MAN address of the router. Then, the router reassembles the packet using the destination MAC address of the actual destination, or the next hop router. TO make MLS transparent to ecternal systems, the Catalyst 6500 must perform this packet manipulation in hardware, without consulting the MSFC for subsequent packets in a flow. This process is referred to as “Inline Header Rewrite.” The Catalyst’s Policy Feature Card performs this function.

Default MLS Configuration



Default IP MLS Configuration Feature	Default Value
IP MLS enable state	Enabled
IP MLS aging time	256 seconds
IP MLS fast aging time	0 seconds (no fast aging)
IP MLS fast aging-time packet threshold	0 packets

The table here displays the default IP MLS configuration.

Table 3-5: Default IP MLS Configuration

Default IP MLS Configuration Feature	Default Value
IP MLS enable state	Enabled
IP MLS aging time	256 seconds
IP MLS fast aging time	0 seconds (no fast aging)
IP MLS fast aging-time packet threshold	0 packets

The table here displays the default IP MMLS switch configuration.

Table 3-6: Default IP MMLS Configuration (Supervisor Engine)

Default IP MMLS Supervisor Engine Configuration Feature	Default Value
Multicast services (IGMP snooping or GMRP)	Disabled
IP MMLS	Enabled

The table here displays the default IP MMLS MSFC configuration.

Table 3-7: Default IP MMLS Configuration (MSFC)

Default IP MMLS MSFC Configuration Feature	Default Value
Multicast routing	Disabled globally
IP PIM routing	Disabled on all interfaces
IP MMLS Threshold	Unconfigured—no default value
IP MMLS	Enabled when multicast routing is enabled and IP PIM is enabled on the interface

The table here displays the default IPX MLS configuration.

Table 3-8: Default IPX MLS Configuration

Default IPX MLS Configuration Feature	Default Value
IPX MLS enable state	Enabled
IPX MLS aging time	256 seconds

IP MLS Configuration Guidelines and Restrictions

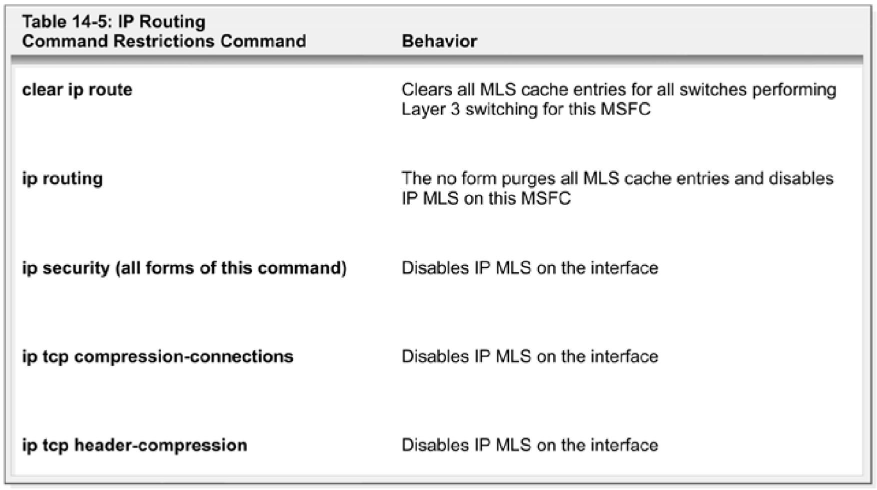


Table 14-5: IP Routing Command Restrictions Command	Behavior
clear ip route	Clears all MLS cache entries for all switches performing Layer 3 switching for this MSFC
ip routing	The no form purges all MLS cache entries and disables IP MLS on this MSFC
ip security (all forms of this command)	Disables IP MLS on the interface
ip tcp compression-connections	Disables IP MLS on the interface
ip tcp header-compression	Disables IP MLS on the interface

Maximum Transmission Unit Size

The default maximum transmission unit (MTU) for IP MLS is 1500. To change the MTU on an IP MLS-enabled interface, enter the **`ip mtu`** *`mtu`* command.

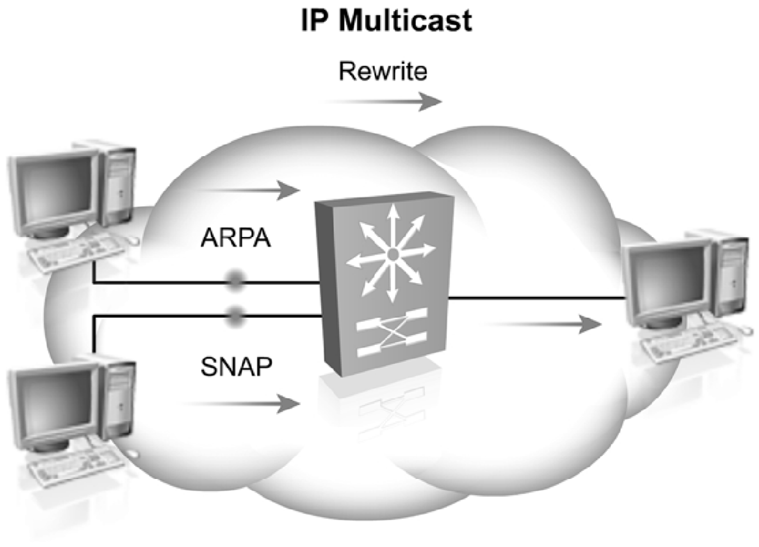
Restrictions on Using IP Routing Commands with IP MLS Enabled

Enabling certain IP processes on an interface will affect IP MLS on the interface. The Table shows the affected commands and the resulting behavior.

Table 3-9: MLS affecting

IP Routing Command Restrictions Command	Behavior
`clear ip route`	Clears all MLS cache entries for all switches performing Layer 3 switching for this MSFC.
`ip routing`	The **`no`** form purges all MLS cache entries and disables IP MLS on this MSFC.
`ip security (all forms of this command)`	Disables IP MLS on the interface.
`ip tcp compression-connections`	Disables IP MLS on the interface.
`ip tcp header-compression`	Disables IP MLS on the interface.

IP MMLS Supervisor Engine Guidelines and Restrictions



These guidelines and restrictions apply when configuring Supervisor Engine 1 for IP MMLS:

- Only ARPA rewrites are supported for IP multicast packets.
- Subnetwork Address Protocol (SNAP) rewrites are not supported.
- You must enable one of the multicast services (IGMP snooping or GMRP) on the switch in order to use IP MMLS.
- IP multicast flows are not multilayer switched if there is no entry in the Layer 2 multicast forwarding table (for example, if no Layer 2 multicast services are enabled or the forwarding table is full). Enter the **`show multicast group`** command to check for a Layer 2 entry for a particular IP multicast destination.
- If a Layer 2 entry is cleared, the corresponding Layer 3 flow information is purged.
- When using two MSFCs that have one or more interfaces in the same VLAN, the switch uses two reserved VLANs (VLANs 1012 and 1013) internally to forward multicast flows properly.
- The MSFC will not act as an external router for a Catalyst 5000 family switch that has Layer 3 switching hardware.

IP MMLS MSFC Configuration Restrictions

IP MMLS does not perform multilayer switching for an IP multicast flow in the following situations:

- For IP multicast groups that fall into these ranges (where * is in the range 0-255):
  - 224.0.0.* through 239.0.0.*
  - 224.128.0.* through 239.128.0.*

Note Groups in the 224.0.0.* range are reserved for routing control packets and must be flooded to all forwarding ports of the VLAN. These addresses map to the multicast MAC address range 01-00-5E-00-00-xx, where xx is in the range 0-0xFF.

- For IP PIM auto-RP multicast groups (IP multicast group addresses 224.0.1.39 and 224.0.1.40).

Note In systems with redundant MSFCs, the IP PIM interface configuration must be the same on both the active and redundant MSFCs.

- For flows that are forwarded on the multicast-shared tree (that is, {*,G,*} forwarding) when the interface or group is running IP PIM sparse-mode.
- If the shortest-path tree (SPT) bit for the flow is cleared when running IP PIM sparse-mode for the interface or group.
- For fragmented IP packets and packets with IP options. However, packets in the flow that are not fragmented or that do not specify IP options are multilayer switched.
- For source traffic received on tunnel interfaces (such as MBONE traffic).
- For any RPF interface with multicast tag switching enabled.

Unsupported IP MMLS Features

If you enable IP MMLS, IP accounting for the interface will not reflect accurate values.

Configuring MLS

This section details the configuration of MLS on the Catalyst 6500.

EXAMPLE

```
Router(config)#interface vlan 100
Router(config-if)#mls ip
```

Disabling and Enabling Unicast MLS on an MSFC Interface

Unicast MLS for IP and IPX is enabled globally by default, but can be disabled and enabled on a specified interface.

This example shows how to disable IP MLS on an MSFC interface:

```
Router(config)# interface vlan 100
Router(config-if)# no mls ip
```

This example shows how to disable IPX MLS on an MSFC interface:

```
Router(config)# interface vlan 100
Router(config-if)# no mls ipx
```

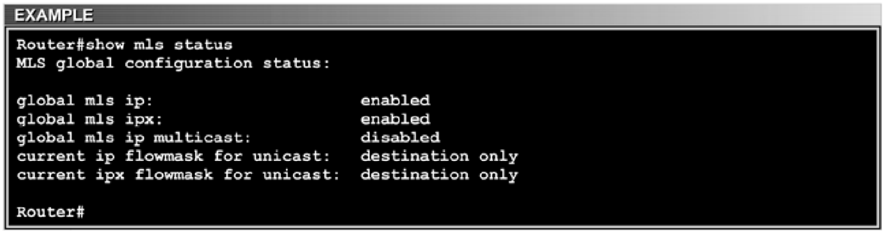
This example shows how to enable IP MLS on an MSFC interface:

```
Router(config)# interface vlan 100
Router(config-if)# mls ip
```

This example shows how to enable IPX MLS on an MSFC interface:

```
Router(config)# interface vlan 100
Router(config-if)# mls ipx
```

Monitoring MLS on the MSFC



EXAMPLE

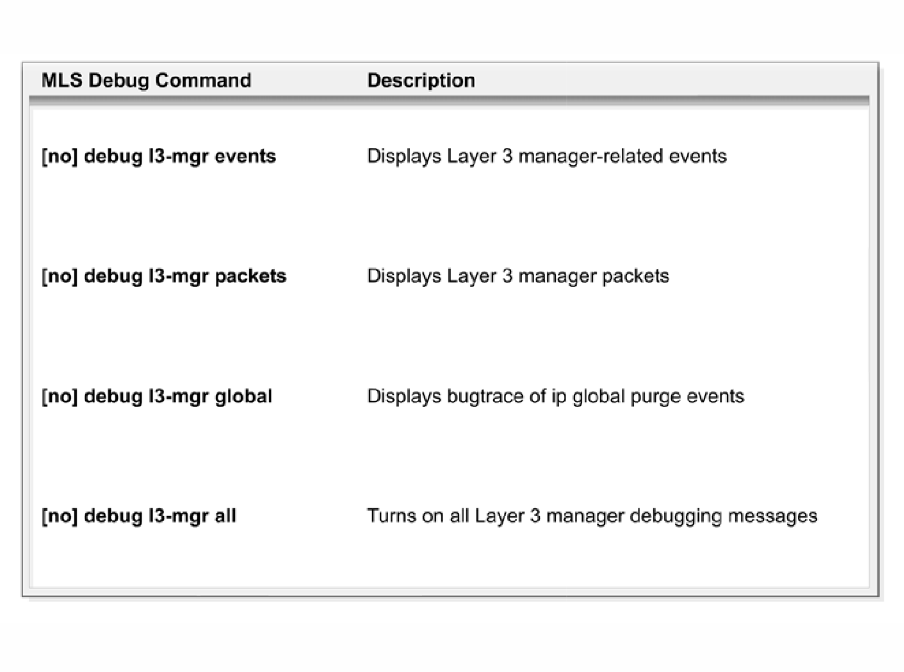
```
Router#show mls status
MLS global configuration status:

global mls ip:                     enabled
global mls ipx:                    enabled
global mls ip multicast:           disabled
current ip flowmask for unicast:   destination only
current ipx flowmask for unicast:  destination only

Router#
```

The **show mls status** command displays MLS details.

This example shows how to display MLS status on the MSFC:

```
Router# show mls status
MLS global configuration status:

global mls ip:                      enabled
global mls ipx:                     enabled
global mls ip multicast:            disabled
current ip flowmask for unicast:    destination only
current ipx flowmask for unicast:   destination only

Router#
```

Using Debug Commands on the MSFC



MLS Debug Command	Description
[no] debug l3-mgr events	Displays Layer 3 manager-related events
[no] debug l3-mgr packets	Displays Layer 3 manager packets
[no] debug l3-mgr global	Displays bugtrace of ip global purge events
[no] debug l3-mgr all	Turns on all Layer 3 manager debugging messages

The table here describes MLS-related debug commands that you can use to troubleshoot MLS problems on the MSFC.

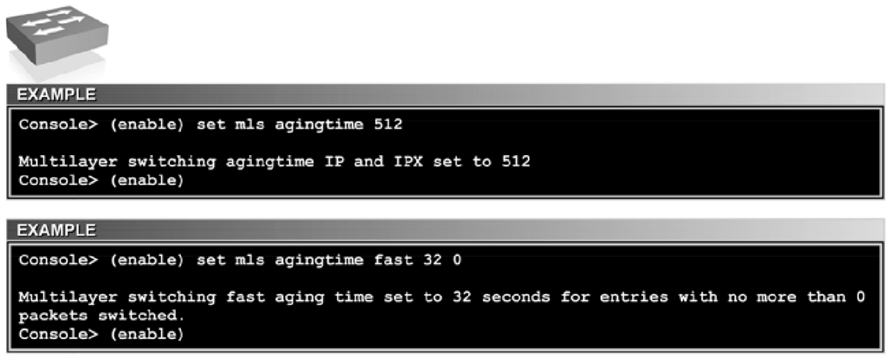
Table 3-10: debug l3-mgr

MLS Debug Command	Description
`[no] debug l3-mgr events`	Displays Layer 3 manager-related events.
`[no] debug l3-mgr packets`	Displays Layer 3 manager packets.
`[no] debug l3-mgr global`	Displays bugtrace of ip global purge events.
`[no] debug l3-mgr all`	Turns on all Layer 3 manager debugging messages.

Table 3-11: MLS Debug Commands—External Router Function

Command	Description
`[no] debug mls ip`	Turns on IP-related events for MLS, including route purging and changes of access lists and flow masks.
`[no] debug mls ipx`	Turns on IPX-related events for MLS, including route purging and changes of access lists and flow masks.
`[no] debug mls rp`	Turns on route processor-related events.
`[no] debug mls locator`	Identifies which switch is switching a particular flow by using MLS explorer packets.
`[no] debug mls all`	Turns on all MLS debugging events.

Configuring MLS on Supervisor Engine 1



EXAMPLE

```
Console> (enable) set mls agingtime 512

Multilayer switching agingtime IP and IPX set to 512
Console> (enable)
```

EXAMPLE

```
Console> (enable) set mls agingtime fast 32 0

Multilayer switching fast aging time set to 32 seconds for entries with no more than 0
packets switched.
Console> (enable)
```

MLS is enabled by default on Catalyst 6000 family switches. You only need to configure Supervisor Engine 1 in these circumstances:

- You want to change the MLS aging time.
- You want to enable NDE.

Note	When you disable IP or IPX MLS on the MSFC, IP or IPX MLS is automatically disabled on Supervisor Engine 1. All existing protocol-specific MLS cache entries are purged.

Specifying MLS Aging-Time Value

The MLS aging time for each protocol (IP and IPX) applies to all protocol-specific MLS cache entries. Any MLS entry that has not been used for *agingtime* seconds is aged out. The default is 256 seconds.

You can configure the aging time in the range of 8 to 2032 seconds in 8-second increments. Any aging-time value that is not a multiple of 8 seconds is adjusted to the closest multiple of 8 seconds. For example, a value of 65 is adjusted to 64 and a value of 127 is adjusted to 128.

This example sets the MLS aging-time to 512 seconds:

```
Console> (enable) set mls agingtime 512

Multilayer switching agingtime IP and IPX set to 512
Console> (enable)
```

To keep the MLS cache size below 32K entries, enable IP MLS fast aging time. The IP MLS fast aging time applies to MLS entries that have no more than pkt_threshold packets switched within fastagingtime seconds after they are created. A typical cache entry that is removed is the entry for flows to and from a Domain Name Server (DNS) or TFTP server; the entry might never be used again after it is created.

Detecting and aging out these entries saves space in the MLS cache for other data traffic. The default fastagingtime value is 0 (no fast aging). You can configure the fastagingtime value to 32, 64, 96, or 128 seconds. Any fastagingtime value that is not configured exactly as the indicated values is adjusted to the closest one. You can configure the pkt_threshold value to 0, 1, 3, 7, 15, 31, or 63 packets.

If you need to enable IP MLS fast aging time, initially set the value to 128 seconds. If the size of the MLS cache continues to grow over 32K entries, decrease the setting until the cache size stays below 32K. If the cache continues to grow over 32K entries, decrease the normal IP MLS aging time.

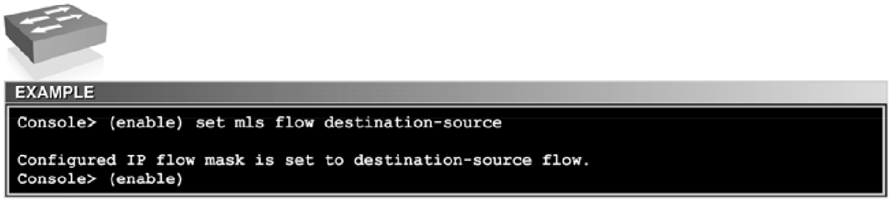
Typical values for fastagingtime and pkt_threshold are 32 seconds and 0 packets (no packets switched within 32 seconds after the entry is created).

To specify the IP MLS fast aging time and packet threshold, perform this task in privileged mode:

This example shows how to set the IP MLS fast aging time to 32 seconds with a packet threshold of 0 packets:

```
Console> (enable) set mls agingtime fast 32 0


Multilayer switching fast aging time set to 32 seconds for entries with no more
than 0

packets switched.

Console> (enable)
```

Setting the Minimum IP MLS Flow Mask



```
Console> (enable) set mls flow destination-source

Configured IP flow mask is set to destination-source flow.
Console> (enable)
```

You can set the minimum granularity of the flow mask for the MLS cache on the PFC. The actual flow mask used will be at least of the granularity specified by this command.

For example, if you do not configure access lists on any MSFC, then the IP MLS flow mask on the PFC is destination-ip by default. However, you can force the PFC to use the source-destination-ip flow mask by setting the minimum IP MLS flow mask using the **`set mls flow destination-source`** command.

Caution The **`set mls flow destination-source`** command purges all existing shortcuts in the MLS cache and affects the number of active shortcuts on the PFC. Exercise care when using this command.

To set the minimum IP MLS flow mask, perform this task in privileged mode:

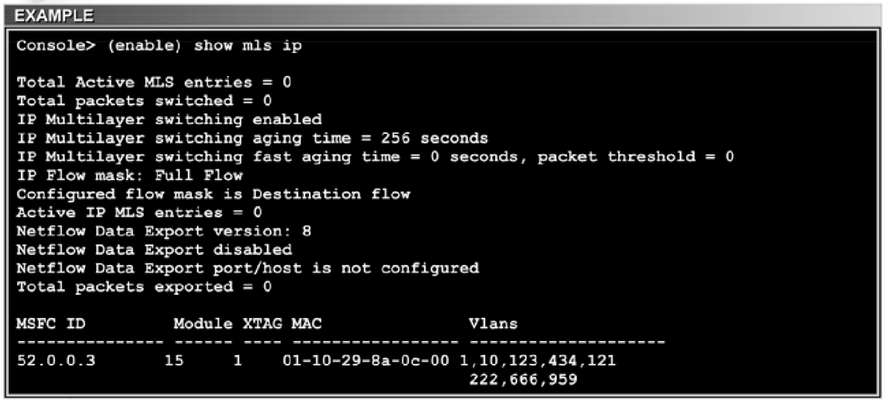
Table 3-12: set mls flow

Task	Command
Set the minimum IP MLS flow mask.	**`set mls flow {destination | destination-source | full}`**

This example shows how to set the minimum IP MLS flow mask to destination-source-ip:

```
Console> (enable) set mls flow destination-source

Configured IP flow mask is set to destination-source flow.
Console> (enable)
```

Displaying MLS Information



```
Console> (enable) show mls ip

Total Active MLS entries = 0
Total packets switched = 0
IP Multilayer switching enabled
IP Multilayer switching aging time = 256 seconds
IP Multilayer switching fast aging time = 0 seconds, packet threshold = 0
IP Flow mask: Full Flow
Configured flow mask is Destination flow
Active IP MLS entries = 0
Netflow Data Export version: 8
Netflow Data Export disabled
Netflow Data Export port/host is not configured
Total packets exported = 0

MSFC ID         Module XTAG MAC               Vlans
--------------- ------ ---- ----------------- --------------------
52.0.0.3        15     1    01-10-29-8a-0c-00 1,10,123,434,121
                                              222,666,959
```

The **show mls** command displays protocol-specific MLS information and MSFC-specific information.

To display protocol-specific MLS information and MSFC-specific information, perform this task:

Table 3-13: show mls

Task	Command
Display general IP or IPX MLS information and MSFC-specific information for all MSFCs.	**show mls {ip** | **ipx}** [*mod*[1]]

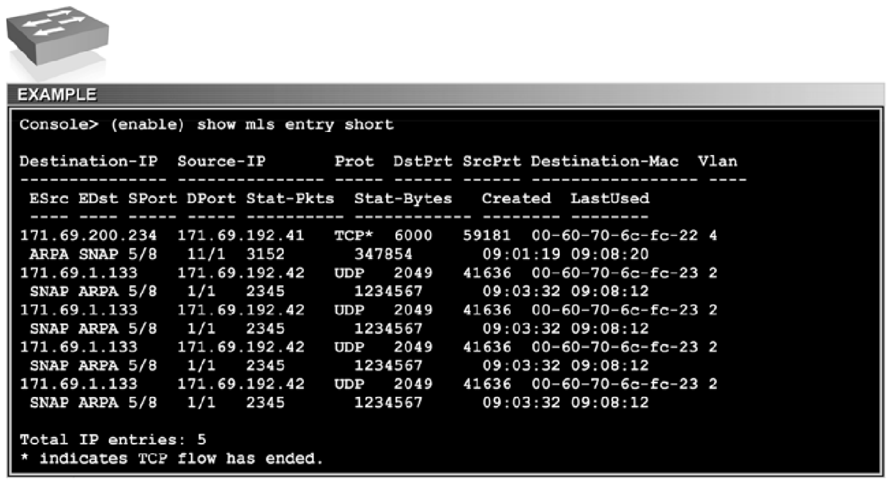
[1] The *mod* keyword specifies the module number of the MSFC; either 15 (if the MSFC is installed on Supervisor Engine 1 in slot 1) or 16 (if the MSFC is installed on Supervisor Engine 1 in slot 2.)

This example shows how to display IP MLS information and MSFC-specific information:

```
Console> (enable) show mls ip

Total Active MLS entries = 0
Total packets switched = 0
IP Multilayer switching enabled
IP Multilayer switching aging time = 256 seconds
IP Multilayer switching fast aging time = 0 seconds, packet threshold = 0
IP Flow mask: Full Flow
Configured flow mask is Destination flow
Active IP MLS entries = 0
```

```
Netflow Data Export version: 8
Netflow Data Export disabled
Netflow Data Export port/host is not configured
Total packets exported = 0

MSFC ID         Module XTAG MAC               Vlans
--------------- ------ ---- ----------------- --------------------
52.0.0.3         15     1    01-10-29-8a-0c-00 1,10,123,434,121
                                              222,666,959

Console> (enable)
```

Displaying IP MLS Cache Entries



```
Console> (enable) show mls entry short

Destination-IP  Source-IP       Prot  DstPrt SrcPrt Destination-Mac  Vlan
--------------- --------------- ----- ------ ------ ----------------- ----
 ESrc EDst SPort DPort Stat-Pkts  Stat-Bytes   Created  LastUsed
 ---- ---- ----- ----- ---------- ------------ -------- --------
171.69.200.234  171.69.192.41   TCP*  6000   59181  00-60-70-6c-fc-22 4
 ARPA SNAP 5/8   11/1  3152       347854       09:01:19 09:08:20
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345       1234567      09:03:32 09:08:12
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345       1234567      09:03:32 09:08:12
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345       1234567      09:03:32 09:08:12
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345       1234567      09:03:32 09:08:12

Total IP entries: 5
* indicates TCP flow has ended.
```

To display all MLS entries (IP and IPX), perform this task in privileged mode:

Table 3-14: show mls entry

Task	Command
Display all MLS entries.	**show mls entry [short | long]**

This example shows how to display all MLS entries (IP and IPX):

```
Console> (enable) show mls entry short

Destination-IP  Source-IP       Prot  DstPrt SrcPrt Destination-Mac  Vlan
--------------- --------------- ----- ------ ------ ----------------- ----
 ESrc EDst SPort DPort Stat-Pkts  Stat-Bytes   Created  LastUsed
 ---- ---- ----- ----- ---------- ------------ -------- --------
171.69.200.234  171.69.192.41   TCP*  6000   59181  00-60-70-6c-fc-22 4
 ARPA SNAP 5/8   11/1  3152       347854       09:01:19 09:08:20
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345       1234567      09:03:32 09:08:12
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345       1234567      09:03:32 09:08:12
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
 SNAP ARPA 5/8   1/1   2345       1234567      09:03:32 09:08:12
171.69.1.133    171.69.192.42   UDP   2049   41636  00-60-70-6c-fc-23 2
```

```
 SNAP ARPA 5/8   1/1   2345       1234567      09:03:32 09:08:12

Total IP entries: 5
* indicates TCP flow has ended.

Destination-IPX           Source-IPX-net Destination-Mac   Vlan Port
 Stat-Pkts Stat-Bytes
------------------------- ------------- ----------------- ---- -----
 --------- -----------
BABE.0000.0000.0001       -             00-a0-c9-0a-89-1d 211  13/37
 30230     1510775
201.00A0.2451.7423        -             00-a0-24-51-74-23 201  14/33
 30256     31795084
501.0000.3100.0501        -             31-00-05-01-00-00 501  9/37
 12121     323232
401.0000.0000.0401        -             00-00-04-01-00-00 401  3/1
 4633      38676

Total IPX entries: 4
Console>
```

Lesson Summary

This lesson accomplished the following:

- Described Multilayer Switching (MLS)
- Listed the advantages of Multilayer Switching vs. traditional Layer 2 switching and Layer 3 routing
- Explained and listed the different Flow Masks involved with Multilayer Switching
- Configured Multilayer Switching on the Catalyst 6500

Next Steps

After completing this lesson, go to:

- Layer 3 Redundancy

References

For additional information, refer to these resources:

- IP Multilayer Switching Sample Configuration: http://www.cisco.com/warp/public/473/39.html
- Troubleshooting IP Multilayer Switching: http://www.cisco.com/warp/public/473/13.html

Layer 3 Redundancy

Overview

This lesson describes the features available on the Catalyst 6500 to provide Layer 3 redundancy. There are different levels of Layer 3 redundancy that can be configured on the Catalyst 6500. This lesson details the configuration of all of them.

Importance

Redundancy at every layer is a critical component to networks of all sizes. The Catalyst 6500 offers many features that provide redundancy at both Layer 2 and Layer 3.

Objectives

Upon completing this lesson, you will be able to:

- Describe the Layer 3 redundancy features offered on the Catalyst 6500
- Describe the use of the Hot Standby Routing Protocol (HSRP)
- Configure HSRP on the MSFCs to provide default gateway redundancy for clients
- Configure High Availability redundancy on the Supervisor Engines

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Have a basic understanding of Layer 3 routing and how the Catalyst 6500 accomplishes routing via the MSFC.

Outline

This lesson includes these sections:

- Overview
- Configuring MSFC Redundancy
- Configuring Redundancy with HSRP
- MSFC Configuration Synchronization Overview
- Single Router Mode Redundancy
- Summary

Configuring MSFC Redundancy

MSFC Redundancy is available in a single chassis with two MSFCs or two chassis each with an MSFC.

Single chassis :

- **Supervisor Engine 1 with Policy Feature Card (PFC) and MSFC or MSFC 2**
- **Supervisor Engine 2 with PFC2 MSFC2**

Dual chassis :

- **Must have at least one supervisor engine**
- **Supervisor must be equipped with a PFC and An MSFC**

Hardware and Software Requirements

To configure Layer 3 redundancy you must have at least one of the following configurations:

- A single chassis with two identical Supervisor Engine daughtercard configurations:
  - Supervisor Engine 1 with Policy Feature Card (PFC) and MSFC or MSFC2 (both Supervisor Engines must have the same type of MSFC)
  - Supervisor Engine 2 with PFC2 and MSFC2
- Two chassis with a Supervisor Engine in each—You must have at least one Supervisor Engine in each chassis. Each Supervisor Engine must be equipped with a PFC and an MSFC.

Note Each MSFC must be running the same release of Cisco IOS software.

Layer 3 Redundancy for a Single Chassis

In a single Catalyst 6000 family chassis, you can have redundant Supervisor Engines, each with an MSFC. You can configure HSRP on the MSFCs to provide transparent default gateway redundancy for IP hosts in the network. HSRP configuration can coexist with IPX and AppleTalk configuration on the same interfaces. If one MSFC fails, HSRP allows one MSFC (router) to assume the function automatically of the other.

Combined with the High Availability feature of Supervisor Engine software release 5.4(1), this configuration provides an added level of redundancy for your network.

Caution You *must* configure both MSFCs identically. This Table summarizes the identical requirements and the exceptions for Layer 3 redundancy for a single switch chassis.

Table 3-15: Single Chassis Layer 3-Redundancy Requirements

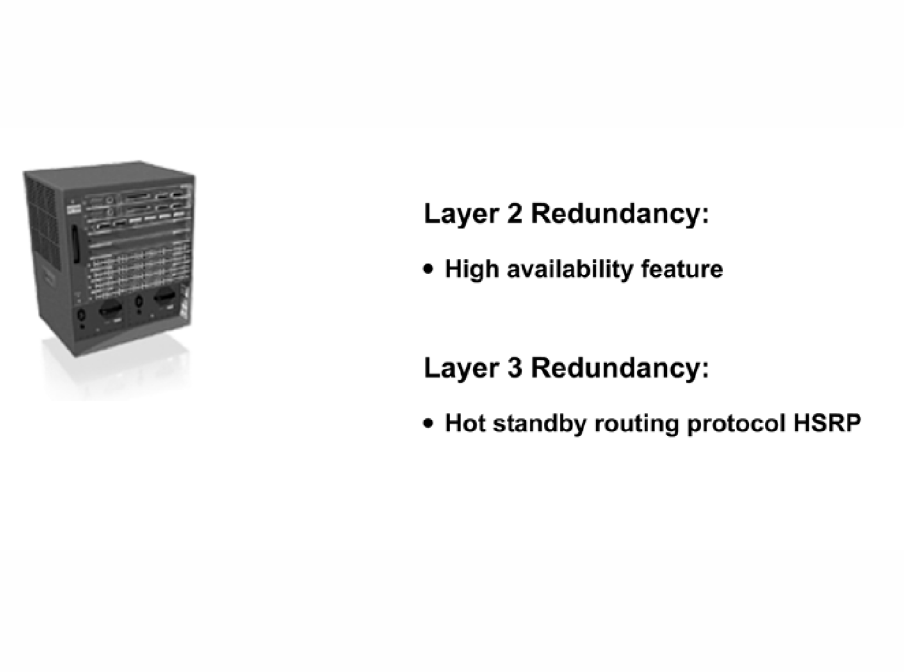
Identical Requirements— Global and Interface Levels	○ Both MSFCs *must* have the following: – Same routing protocols – Same static routes – Same default routes – Same policy routes – Same VLAN interfaces – Same IOS ACLs[1], [2] ■ All interfaces *must* have the same administrative status
Exceptions—Interface Level	■ HSRP standby commands ■ IP address commands[3] ■ IPX network[3]
Exceptions—Global Level	■ IP `default-gateway` ■ IPX internal-network ■ IPX default-route

[1] Dynamic and reflexive ACLs, which are based on actual data flow, may be programmed by either MSFC.

[2] In addition to defining the same ACLs on both MSFCs, you must also apply the ACLs to the same VLAN interfaces, in the same direction, on both MSFCs.

[3] The IP or IPX addresses do not have to be identical on both MSFCs, but there must be an IP or IPX address configured on both MSFCs.

Routing Protocol Peering



Layer 2 Redundancy:

- **High availability feature**

Layer 3 Redundancy:

- **Hot standby routing protocol HSRP**

In a redundant Supervisor Engine and dual MSFC configuration, one Supervisor Engine is fully operational (active) and the other Supervisor Engine is in standby mode; however, both MSFCs are operational (in terms of programming the PFC on the active Supervisor Engine) and act as independent routers.

Note PFC: With the PFC, MLS entries can be associated with either MSFC (based on which MSFC routed the first packet). Only the PFC on the active Supervisor Engine switches the packets.

Note PFC2: With PFC2, only the designated MSFC programs the forwarding information base (FIB) the adjacency table, IOS software, and policy routing ACLs on the active Supervisor Engine. If you configure static routes or policy routing, you must have the identical configuration on both MSFCs. If you have a static route on the nondesignated MSFC that is not on the designated MSFC, that route will not be programmed in the PFC2.

Although the MSFCs (from a peering perspective) act as independent routers, the two MSFCs in the chassis operate at the same time, have the same interfaces, and run the same routing protocols.

If you combine High Availability on the Supervisor Engines with HSRP on the MSFCs, you have the following Layer 2 and Layer 3 redundancy mechanisms:

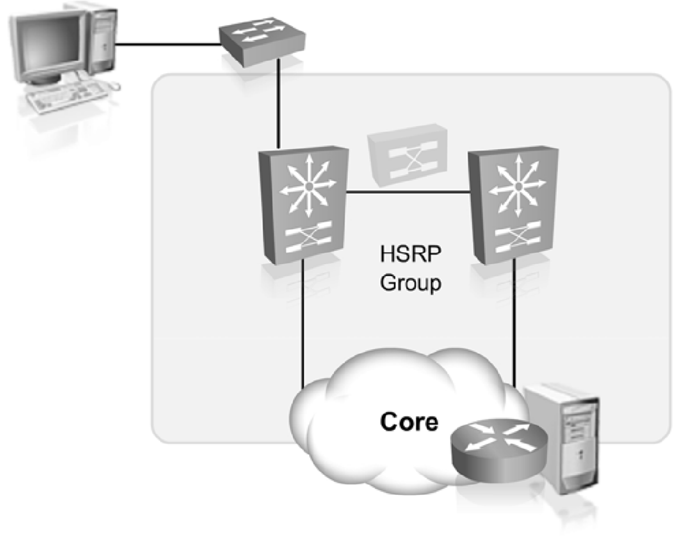
- Layer 2 redundancy for the Supervisor Engines (one active and one in standby)—If the active Supervisor Engine fails (the MSFC installed on it will also fail), both Layer 2 and Layer 3 functions roll over to the redundant Supervisor Engine and MSFC combination.

- Layer 3 redundancy and load sharing for the two MSFCs—If one MSFC fails, the other MSFC takes over almost immediately (using HSRP) without any Layer 2 disruption (the active Supervisor Engine continues to forward Layer 2 traffic).

The Layer 3 entries programmed by the failed MSFC on the active Supervisor Engine are used until they gracefully age out and are replaced by the Layer 3 entries populated by the newly active MSFC. Aging takes 4 minutes and allows the newly active MSFC to repopulate the MLS entries using its XTAG value, while concurrently hardware-switching flows yet to be aged. In addition, this process prevents a newly active MSFC from being overwhelmed with initial flow traffic.

Configuring Redundancy with HSRP

HSRP can be used to provide default gateway redundancy to clients in the network.



- **HSRP defines a set of routers working together to represent one virtual fault-tolerant router**

Understanding How HSRP Works

HSRP uses a priority scheme to determine which HSRP-configured router is to be the default active router. To configure a router as the active router you assign it a priority that is higher than the priority of all the other HSRP-configured routers. The default priority is 100, so if you configure just one router to have a higher priority, that router will be the default active router. The interface with the highest HSRP priority is the active interface for that HSRP group.

HSRP works by the exchange of multicast messages that advertise priority among HSRP-configured routers. When the active router fails to send a hello message within a configurable period of time, the standby router with the highest priority becomes the active router. The transition of packet-forwarding functions between routers is completely transparent to all hosts on the network.

HSRP-configured routers exchange three types of multicast messages:

- Hello: The hello message conveys to other HSRP routers the router's HSRP priority and state information. By default an HSRP router sends hello messages every three seconds.
- Coup: When a standby router assumes the function of the active router, it sends a coup message.
- Resign: A router that is the active router sends this message when it is about to shut down, or when a router that has a higher priority sends a hello message.

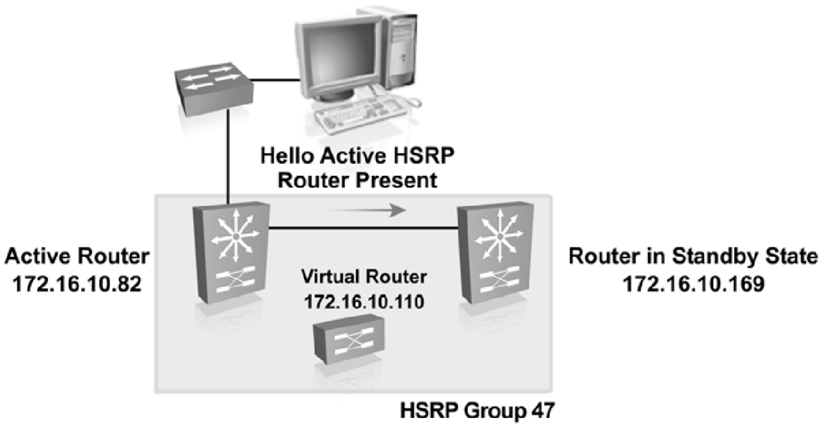
At any time, HSRP-configured routers are in one of the following states:

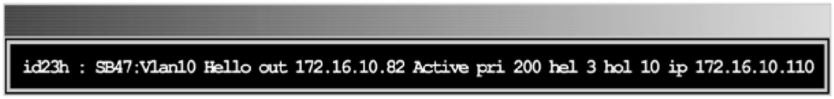
- Active: The router is performing packet-transfer functions.
- Standby: The router is prepared to assume packet-transfer functions if the active router fails.
- Speaking and listening: The router is sending and receiving hello messages.
- Listening: The router is receiving hello messages.

Note PFC2: The PFC2 supports a maximum of 16 unique HSRP group numbers. You can use the same HSRP group numbers in different VLANs. If you configure more than 16 HSRP groups, this restriction prevents use of the VLAN number as the HSRP group number.

Note PFC2: Identically numbered HSRP groups use the same virtual MAC address, which might cause errors if you configure bridging on the MSFC. Because of the restriction to 16 unique HSRP group numbers, CEF for PFC2 cannot support the **`standby use-bia HSRP`** command.

HSRP Router Communication



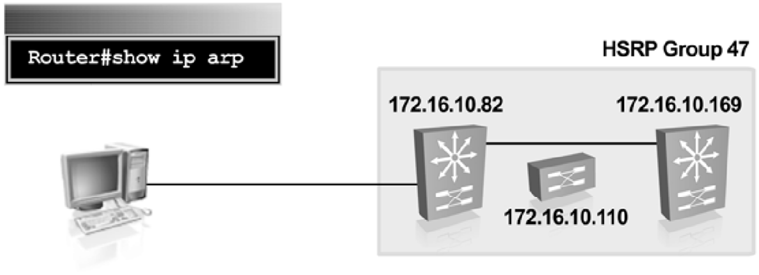


```
id23h : SB47:Vlan10 Hello out 172.16.10.82 Active pri 200 hel 3 hol 10 ip 172.16.10.110
```

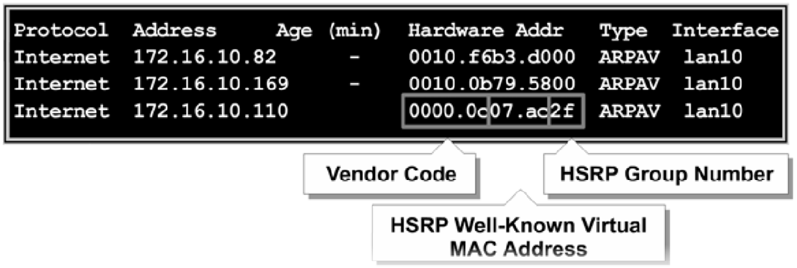
Routers running HSRP communicate HSRP information between each other via HSRP hello packets. These packets are sent to the destination IP multicast address 224.0.0.2 (reserved multicast address used to communicate to all routers) on User Datagram Protocol (UDP) port 1985. These hello packets are sourced with the configured IP address on the interface, and the burned-in MAC address of the interface, as opposed to the HSRP or virtual IP, and MAC address. This use of source addressing is necessary so that HSRP routers can correctly identify each other.

The only exception to the above behavior is for Cisco 2500, 4000, and 4500 routers. These routers have Ethernet hardware that only recognizes a single MAC address. Therefore, these routers will use the HSRP MAC address when they are the active router, and their burned-in address for HSRP hello packets.

HSRP Standby IP Address Communication



```
Router#show ip arp
```



```
Protocol  Address        Age (min)  Hardware Addr   Type   Interface
Internet  172.16.10.82       -      0010.f6b3.d000  ARPAV   lan10
Internet  172.16.10.169      -      0010.0b79.5800  ARPAV   lan10
Internet  172.16.10.110             0000.0c07.ac2f  ARPAV   lan10
```

Since host workstations are configured with their default gateway as the HSRP standby IP address, hosts must communicate with the MAC address associated with the HSRP standby IP address. This MAC address will be a virtual MAC address composed of 0000.0c07.ac**, where ** is the HSRP group number in hexadecimal based on the respective interface.

For example, HSRP group one will use the HSRP virtual MAC address of 0000.0c07.ac01. Hosts on the adjoining LAN segment use the normal ARP process to resolve the associated MAC addresses.

0000.0c = Cisco's Manufacturer ID

07ac = Cisco's well-known code for HSRP

01 = The HSRP group number

Configuring HSRP on an MSFC VLAN Interface

To configure HSRP on an MSFC VLAN interface, perform this task in interface configuration mode:

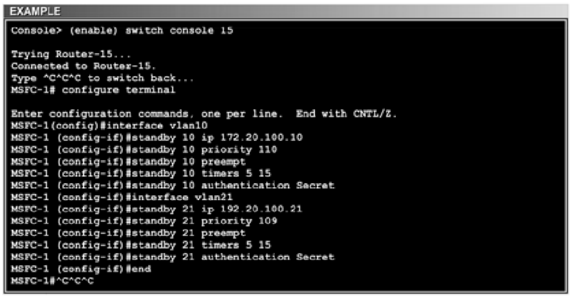
Table 3-16: HSRP Commands

Task	Command
Step 1: Enable HSRP and specify the HSRP IP address. If you do not specify a group-number, group 0 is used. To assist in troubleshooting, configure the group number to match the VLAN number.	`Router(config-if)#` **`standby`** `[`*`group-number`*`]` **`ip`** `[`*`ip-address`*`]`
Step 2: Specify the priority for the HSRP interface. Increase the priority of at least one interface in the HSRP group (the default is 100). The interface with the highest priority becomes active for that HSRP group.	`Router(config-if)#` **`standby`** `[`*`group-number`*`]` **`priority`** *`priority`*
Step 3: Configure the interface to preempt the current active HSRP interface and become active if the interface priority is higher than the priority of the current active interface.	`Router(config-if)#` **`standby`** `[`*`group-number`*`]` **`preempt`** `[`**`delay`** *`delay`*`]`
Step 4: (Optional) Set the HSRP hello timer and holdtime timer for the interface. The default values are 3 (hello) and 10 (holdtime). All interfaces in the HSRP group should use the same timer values.	`Router(config-if)#` **`standby`** `[`*`group-number`*`]` **`timers`** *`hellotime holdtime`*
Step 5: (Optional) Specify a clear-text HSRP authentication string for the interface. All interfaces in the HSRP group should use the same authentication string.	`Router(config-if)#` **`standby`** `[`*`group-number`*`]` **`authentication`** *`string`*

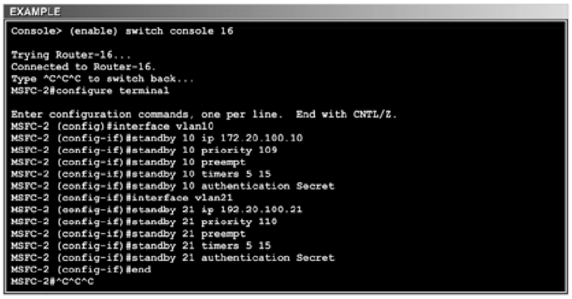
For the following examples, the designated MSFC is on the active Supervisor Engine. To determine the status of the designated MSFC, enter the show fm features or show redundancy command. This example shows that MSFC-2 is the designated MSFC:

```
MSFC-1# show redundancy

Designated Router: 1 Non-designated Router:2

Redundancy Status: non-designated
Config Sync AdminStatus  : enabled
Config Sync RuntimeStatus: enabled

MSFC-2# show redundancy

Designated Router: 1 Non-designated Router:2

Redundancy Status: designated
Config Sync AdminStatus  : enabled
Config sync RuntimeStatus: enabled
```

Example—Single Chassis with Dual Supervisor Engines and MSFCs



```
Console> (enable) switch console 15

Trying Router-15...
Connected to Router-15.
Type ^C^C^C to switch back...
MSFC-1# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
MSFC-1(config)#interface vlan10
MSFC-1 (config-if)#standby 10 ip 172.20.100.10
MSFC-1 (config-if)#standby 10 priority 110
MSFC-1 (config-if)#standby 10 preempt
MSFC-1 (config-if)#standby 10 timers 5 15
MSFC-1 (config-if)#standby 10 authentication Secret
MSFC-1 (config-if)#interface vlan21
MSFC-1 (config-if)#standby 21 ip 192.20.100.21
MSFC-1 (config-if)#standby 21 priority 109
MSFC-1 (config-if)#standby 21 preempt
MSFC-1 (config-if)#standby 21 timers 5 15
MSFC-1 (config-if)#standby 21 authentication Secret
MSFC-1 (config-if)#end
MSFC-1#^C^C^C
```



```
Console> (enable) switch console 16

Trying Router-16...
Connected to Router-16.
Type ^C^C^C to switch back...
MSFC-2#configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
MSFC-2 (config)#interface vlan10
MSFC-2 (config-if)#standby 10 ip 172.20.100.10
MSFC-2 (config-if)#standby 10 priority 109
MSFC-2 (config-if)#standby 10 preempt
MSFC-2 (config-if)#standby 10 timers 5 15
MSFC-2 (config-if)#standby 10 authentication Secret
MSFC-2 (config-if)#interface vlan21
MSFC-2 (config-if)#standby 21 ip 192.20.100.21
MSFC-2 (config-if)#standby 21 priority 110
MSFC-2 (config-if)#standby 21 preempt
MSFC-2 (config-if)#standby 21 timers 5 15
MSFC-2 (config-if)#standby 21 authentication Secret
MSFC-2 (config-if)#end
MSFC-2#^C^C^C
```

In the following example, High Availability is configured on the Supervisor Engines, and HSRP is configured on the MSFCs.

Single Chassis with Redundant Supervisors and MSFCs

The example shown here explains how to configure HSRP on the MSFC in Switch S1:

```
Console> (enable) switch console 15

Trying Router-15...
Connected to Router-15.
Type ^C^C^C to switch back...
MSFC-1# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
MSFC-1(config)# interface vlan10
MSFC-1 (config-if)# standby 10 ip 172.20.100.10
MSFC-1 (config-if)# standby 10 priority 110
MSFC-1 (config-if)# standby 10 preempt
MSFC-1 (config-if)# standby 10 timers 5 15
MSFC-1 (config-if)# standby 10 authentication Secret
MSFC-1 (config-if)# interface vlan21
MSFC-1 (config-if)# standby 21 ip 192.20.100.21
MSFC-1 (config-if)# standby 21 priority 109
```

```
MSFC-1 (config-if)# standby 21 preempt
MSFC-1 (config-if)# standby 21 timers 5 15
MSFC-1 (config-if)# standby 21 authentication Secret
MSFC-1 (config-if)# end
MSFC-1# ^C^C^C

Console> (enable) switch console 16

Trying Router-16...
Connected to Router-16.
Type ^C^C^C to switch back...
MSFC-2# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
MSFC-2 (config)# interface vlan10
MSFC-2 (config-if)# standby 10 ip 172.20.100.10
MSFC-2 (config-if)# standby 10 priority 109
MSFC-2 (config-if)# standby 10 preempt
MSFC-2 (config-if)# standby 10 timers 5 15
MSFC-2 (config-if)# standby 10 authentication Secret
MSFC-2 (config-if)# interface vlan21
MSFC-2 (config-if)# standby 21 ip 192.20.100.21
MSFC-2 (config-if)# standby 21 priority 110
MSFC-2 (config-if)# standby 21 preempt
MSFC-2 (config-if)# standby 21 timers 5 15
MSFC-2 (config-if)# standby 21 authentication Secret
MSFC-2 (config-if)# end
MSFC-2# ^C^C^C
```

MSFC Configuration Synchronization Overview

If you are using some of the Layer 3 redundancy options discussed in this lesson, it is probably a good idea to have the same IOS and configuration on both MSFCs. MSFC High Availability allows for automatic synchronization of the startup configuration and running configuration on the MSFCs.

MSFC High Availability allows for automatic synchronization of the startup configuration and running configuration between the designated MSFC (the MSFC to come online first, or the MSFC that has been online the longest) and the nondesignated MSFC. High-availability redundancy is disabled by default.

Caution Configuration synchronization is only supported for IP and IPX configurations, and before enabling synchronization you must ensure that both MSFCs have the exact same configurations for all protocols. If you are using AppleTalk, DECnet, VINES or any other routing you must manually ensure that the exact same configurations are on both MSFCs for all protocols.

To determine the status of the designated MSFC, enter the show fm features or show redundancy command:

```
MSFC-1# show redundancy

Designated Router: 1 Non-designated Router:2

Redundancy Status: non-designated
Config Sync AdminStatus  : enabled
Config Sync RuntimeStatus: enabled

MSFC-2# show redundancy

Designated Router: 1 Non-designated Router:2

Redundancy Status: designated
Config Sync AdminStatus  : enabled
Config sync RuntimeStatus: enabled
```

High-Availability redundancy provides startup and running configuration synchronization.

When you enable High-Availability redundancy, the startup configuration of both MSFCs is updated when you enter either of these commands on the designated MSFC:

```
write mem
copy source startup-config
```

When you enable High-Availability redundancy, every configuration command executed on the designated MSFC is sent to the nondesignated MSFC. Also, the running configuration synchronization is updated when you enter the **`copy`** *`source`* **`running-config`** command on the designated MSFC.

Configuration Synchronization States

The two states for the configuration synchronization are as follows:

- Config Sync AdminStatus—signifies what the user has configured for this feature at that moment.
- Config Sync RuntimeStatus—enabled only when the following occurs:
  - The Config Sync AdminStatus is enabled on both designated and nondesignated MSFCs.
  - The designated and nondesignated MSFCs are running compatible images.

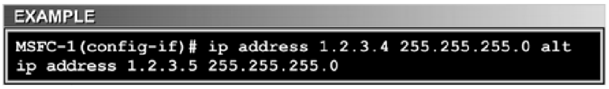
When you enable the Config Sync RuntimeStatus, the following occurs:

- No configuration mode is available on the CLI of the nondesignated MSFC; EXEC mode is available.
- The **`alt`** keyword is available and required.
- The running and startup configurations are synchronized.

When the Config Sync RuntimeStatus is in disabled mode, the following occurs:

- Configuration mode is available on the CLI of both MSFCs.
- The **`alt`** keyword is available but optional.
- The running and startup configurations are not synchronized.

alt Keyword Usage



EXAMPLE

```
MSFC-1(config-if)# ip address 1.2.3.4 255.255.255.0 alt
ip address 1.2.3.5 255.255.255.0
```

When you enable the Config Sync RuntimeStatus, the configuration mode on the nondesignated MSFC is disabled; only the EXEC mode is still available. Configuration of both MSFCs is made through the console or a Telnet session on the designated MSFC.

To configure both MSFCs from a single console, enter the `alt` keyword to specify an alternate configuration. When specifying the alternate configuration, the configuration specified before the `alt` keyword relates to the MSFC on the Supervisor Engine in slot 1 of the switch; the configuration specified after the `alt` keyword relates to the MSFC on the Supervisor Engine in slot 2.

Note The `alt` keyword is required when Config Sync AdminStatus is enabled.

The interface and global configuration commands using the `alt` keyword are below.

Interface Configuration Commands

- Router(config-if)# [**no**] **standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **alt** [**no**] **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]

- Router(config-if)# [**no**] **standby** [*group-number***]** **priority** *priority* [**preempt** [**delay** *delay*]] **alt** [**no**] **standby** [*group-number*] **priority** *priority* [**preempt** [**delay** *delay*]]

- Router(config-if)# [**no**] **ip address** *ip-address mask* [**secondary**] **alt** [**no**] **ip address** *ip-address mask* [**secondary**]

- Router(config-if)# [**no**] **ipx network** *network* [**encapsulation** *encapsulation-type* [**secondary**]] [**alt** [**no**] **ipx network** *network* [**encapsulation** *encapsulation-type* [**secondary**]]]

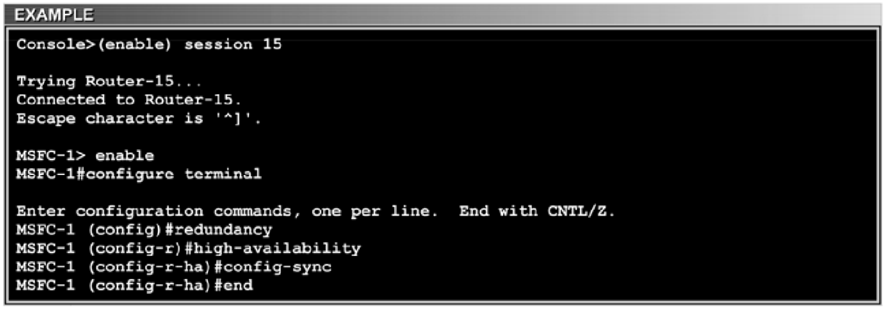
Global Configuration Commands

- Router(config)# [**no**] **hostname** *hostname* **alt hostname** *hostname*
- Router(config)# [**no**] **ip default-gateway** *ip-address* **alt** [**no**] **ip default-gateway** *ip-address*
- Router(config)# **router bgp** *autonomous_system_number*
- Router(config-router)# **bgp router-id** *ip-address* [**alt** *ip-address*]
- Router(config)# **router ospf** *process-id*
- Router(config-router)# **router-id** *ip-address* [**alt** *ip-address*]

This example shows how the `alt` keyword is used when entering the `ip address` command:

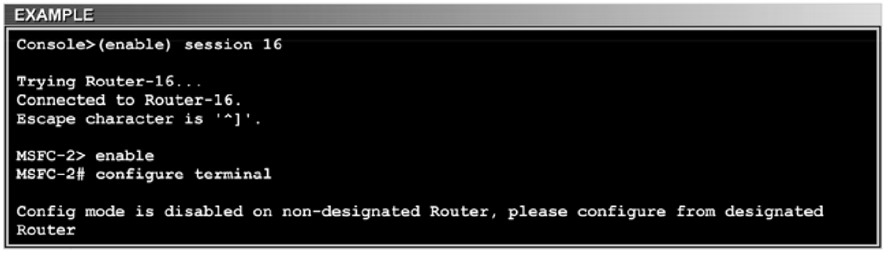
```
MSFC-1(config-if)# ip address 1.2.3.4 255.255.255.0 alt ip address 1.2.3.5
255.255.255.0
```

Enabling or Disabling Configuration Synchronization



EXAMPLE

```
Console>(enable) session 15

Trying Router-15...
Connected to Router-15.
Escape character is '^]'.

MSFC-1> enable
MSFC-1#configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
MSFC-1 (config)#redundancy
MSFC-1 (config-r)#high-availability
MSFC-1 (config-r-ha)#config-sync
MSFC-1 (config-r-ha)#end
```



EXAMPLE

```
Console>(enable) session 16

Trying Router-16...
Connected to Router-16.
Escape character is '^]'.

MSFC-2> enable
MSFC-2# configure terminal

Config mode is disabled on non-designated Router, please configure from designated
Router
```

To enable High-Availability redundancy, perform this task in privileged mode:

Table 3-17: Enabling High-Availability Redundancy

Task	Command
Step 1: Enable redundancy.	`redundancy`
Step 2: Enable High Availability.	`high-availability`
Step 3: Enable or disable configuration synchronization.	`[no] config-sync`

This example shows how to enable High-Availability redundancy and configuration synchronization (MSFC-1 is the designated MSFC):

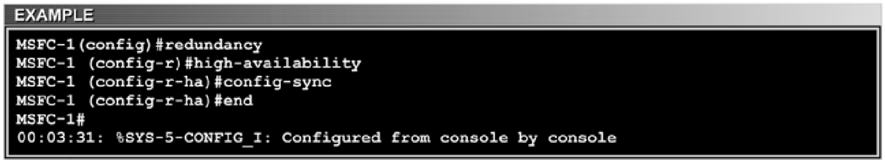
```
Console>(enable) session 15

Trying Router-15...
Connected to Router-15.
Escape character is '^]'.

MSFC-1> enable
MSFC-1# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
MSFC-1 (config)# redundancy
MSFC-1 (config-r)# high-availability
MSFC-1 (config-r-ha)# config-sync
MSFC-1 (config-r-ha)# end
```

Note When you enable High-Availability redundancy, the configuration mode is disabled on the nondesignated MSFC; only the EXEC mode is available.

In this example, MSFC-2 is the nondesignated MSFC; High-Availability redundancy and configuration synchronization are enabled:

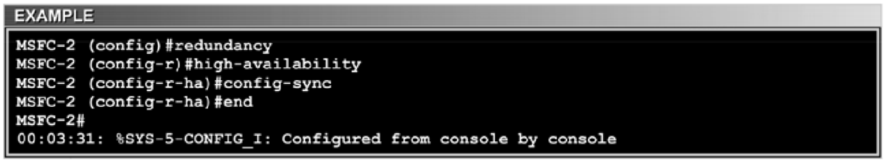
```
Console>(enable) session 16

Trying Router-16...
Connected to Router-16.
Escape character is '^]'.

MSFC-2> enable
MSFC-2# configure terminal

Config mode is disabled on non-designated Router, please configure from
designated Router
```

EXAMPLE: Enabling Configuration Synchronization on Both MSFCs



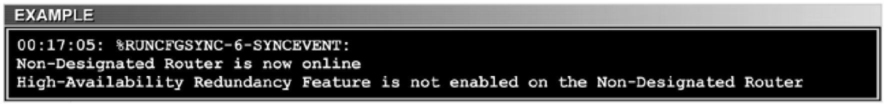
EXAMPLE

```
MSFC-1(config)#redundancy
MSFC-1 (config-r)#high-availability
MSFC-1 (config-r-ha)#config-sync
MSFC-1 (config-r-ha)#end
MSFC-1#
00:03:31: %SYS-5-CONFIG_I: Configured from console by console
```



EXAMPLE

```
MSFC-2 (config)#redundancy
MSFC-2 (config-r)#high-availability
MSFC-2 (config-r-ha)#config-sync
MSFC-2 (config-r-ha)#end
MSFC-2#
00:03:31: %SYS-5-CONFIG_I: Configured from console by console
```



EXAMPLE

```
00:17:05: %RUNCFGSYNC-6-SYNCEVENT:
Non-Designated Router is now online
High-Availability Redundancy Feature is not enabled on the Non-Designated Router
```

This scenario assumes both MSFCs are up.

When you enable configuration synchronization on both MSFCs, the IP addresses on all the interfaces are checked first. If an IP address is specified for the designated MSFC, but not specified for the nondesignated MSFC, a message is displayed indicating the first interface for which the alternate IP address was not specified.

After checking IP addresses, the HSRP addresses are checked; if an HSRP address is specified for the designated MSFC, but not specified for the nondesignated MSFC, a message is displayed indicating the first interface for which the alternate HSRP (standby) address was not specified.

After checking the HSRP addresses, the IPX network address is checked.

The designated MSFC is configured first. This example shows a missing alternate configuration for the VLAN 1 interface:

```
MSFC-2# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
MSFC-2 (config)# redundancy
MSFC-2 (config-r)# high-availability
MSFC-2 (config-r-ha)# config-sync

Alternate IP address missing for Vlan1
The alternate configuration is missing. The auto-config sync can not be enabled
```

Note When specifying the alternate IP configuration, the configuration specified before the alt keyword relates to the MSFC on the Supervisor Engine in slot 1 of the switch; the configuration specified after the alt keyword relates to the MSFC on the Supervisor Engine in slot 2.

This example shows how to specify the alternate configuration for VLAN 1:

```
MSFC-2 (config)# interface vlan 1
MSFC-2 (config-if)# ip address 70.0.70.4 255.255.0.0 alt ip address 70.0.70.5
255.255.0.0
MSFC-2 (config-if)# exit
```

This example shows that High-Availability redundancy is accepted:

```
MSFC-2 (config)# redundancy
MSFC-2 (config-r)# high-availability
MSFC-2 (config-r-ha)# config-sync
MSFC-2 (config-r-ha)# end
MSFC-2#
00:03:31: %SYS-5-CONFIG_I: Configured from console by console
```

Because the Config Sync AdminStatus on the nondesignated MSFC is disabled, the Config Sync RuntimeStatus on the designated MSFC will remain in disabled mode. The following message is displayed on the designated MSFC:

```
00:17:05: %RUNCFGSYNC-6-SYNCEVENT:
Non-Designated Router is now online
High-Availability Redundancy Feature is not enabled on the Non-Designated Router
```

This example shows how to enable the configuration synchronization feature on the nondesignated MSFC:

```
MSFC-1(config)# redundancy
MSFC-1 (config-r)# high-availability
MSFC-1 (config-r-ha)# config-sync
MSFC-1 (config-r-ha)# end
MSFC-1#
00:03:31: %SYS-5-CONFIG_I: Configured from console by console
```

Note When you enable High-Availability redundancy, the configuration mode is disabled on the console of the nondesignated MSFC; only the EXEC mode is available.

The following message, acknowledging that the High-Availability redundancy is enabled, and that the configuration mode will be automatically exited, is displayed on the nondesignated MSFC:

```
00:18:57: %RUNCFGSYNC-6-SYNCEVENT:
The High-Availability Redundancy Feature is enabled
The config mode is no longer accessible

MSFC-1#

00:19:41: %RUNCFGSYNC-6-SYNCEVENT:
Non-Designated Router is now online
Running Configuration Synchronization will begin in 1 minute
```

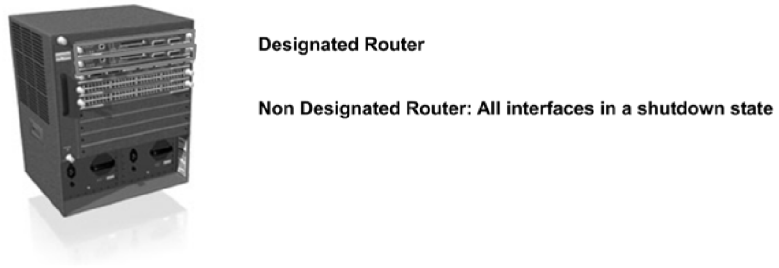
A one-minute timer will start, allowing for stabilization of the nondesignated MSFC. When the timer expires, a snapshot of the current running configuration is sent to the nondesignated MSFC. This message is displayed before the running configuration is synchronized:

```
00:20:41: %RUNCFGSYNC-6-SYNCEVENT:
Syncing Running Configuration to the Non-Designated Router

00:20:41: %RUNCFGSYNC-6-SYNCEVENT:
Syncing Startup Configuration to the Non-Designated Router
```

At this point the designated MSFC and nondesignated MSFC have the same running configuration after synchronization.

Single Router Mode Redundancy

SRM redundancy is an alternative to internally redundant (dual) MSFC2 configurations where both MSFC2s are active at the same time.



In SRM redundancy, only the designated router is visible to the network at any given time. The nondesignated router is booted up completely and participates in configuration synchronization, which is automatically enabled when entering SRM. All configuration following the "alt" keyword is ignored in SRM; the nondesignated router's configuration is exactly the same as the designated router, but its interfaces are kept in a line down state and are not visible to the network.

Processes, such as routing protocols, are created on the nondesignated router and the designated router, but all nondesignated router interfaces are in a line down state; they do not send or receive updates from the network.

When the designated router fails, the nondesignated router changes its state from a nondesignated router to a designated router, and its interface state changes to link up. The newly designated router builds up its routing table while the existing Supervisor Engine switch processor entries are used to forward Layer 3 traffic. The switch processor continues to forward Layer 3 packets using the old entries. After a predefined time, the newly designated router downloads the new Layer 3 switching information to the switch processor.

Note With Cisco IOS Releases 12.1(11b)E and later, you can specify the transition time that the newly designated router waits before downloading the new Layer 3 switching information to the Supervisor Engine switch processor. For configuration details, see the "Specifying the Transition Time on the Newly Designated Active Router" section.

Hardware and Software Requirements

SRM Redundancy Requires

- **A single chassis with two identical supervisor engine daughter card configurations:**
  - Supervisor Engine 2 with Policy Feature Card 2 (PFC2) and MSFC2
  - Supervisor Engine 1 with PFC and MSFC or MSFC2

To configure SRM redundancy, you must have the following hardware and software:

- A single chassis with two identical Supervisor Engine daughter card configurations:
  - Supervisor Engine 2 with Policy Feature Card 2 (PFC2) and MSFC2
  - Supervisor Engine 1 with PFC and MSFC or MSFC2

Note Cisco IOS Release 12.1(8a)E4 provides initial support for SRM redundancy with Supervisor Engine 1 and MSFC.

Note Multicast support: In software releases prior to release 7.1(1), when using Supevisor Engine 1 with the MSFC or MSFC2 for SRM redundancy, be aware that failover to the second MSFC is not stateful for multicast MLS. When the primary MSFC fails, all multicast MLS entries are removed and are then recreated and reinstalled in the hardware by the newly active MSFC.

Note Multicast support: In software releases 7.1(1) and later, there is improved SRM redundancy support for multicast traffic for Supervisor Engine 1 with PFC and MSFC2 and Supervisor Engine 2 with PFC2 and MSFC2. Multicast improvements are not supported on Supervisor Engine 1 with MSFC. When SRM redundancy is enabled there are improved convergence times, and less disruption of multicast traffic during switchovers. The MSFC2 is protected from being overloaded with multicast traffic during the switchover. The switch caches flows from the MSFC2 that went down, and uses the cached flows to forward traffic until the newly activated MSFC2 learns the routes. Only a few flows at a time are provided to the MSFC2 to prevent it from being overwhelmed.

- Supervisor engine software release 6.3(1) or later
- Cisco IOS Release 12.1(8a)E2 or later

Configuration Guidelines

Use these guidelines when configuring SRM redundancy:

- SRM redundancy requires that both the designated router and nondesignated router run the same Cisco IOS image.

- SRM redundancy requires that a Cisco IOS image is present in the bootflash of both the designated router and nondesignated router.

- With SRM redundancy, the nondesignated router cannot connect to external networks.

- With SRM redundancy, we do not recommend booting from an external network with the designated router. Booting from the network could severely degrade SRM functionality.

- With SRM redundancy, the designated router can reach external networks and copy commands such as **`copy tftp`**: can be used without any restrictions.

- For SRM to work properly, High Availability must be enabled on the Supervisor Engine.

- When using authentication methods to control access to the switch such as RADIUS or TACACS+, you need to configure a fallback option to login with a local username and password if you want to be able to access the nondesignated router through the **`switch console`** or **`session`** commands.

To configure SRM redundancy, perform these steps:

Caution Before going from dual router mode to SRM redundancy, we recommend that you use the **`copy running-config`** command on the MSFCs to save the non-SRM configuration to bootflash. When going to SRM redundancy, the alternative configuration (the configuration following the **`alt`** keyword) is lost. Therefore, before enabling SRM redundancy, save the dual router mode configuration to bootflash by entering the following command on both **`MSFCs:copy running-config bootflash:`***`nosrm_dual_router_config`*.

Note This procedure assumes that the designated router is the MSFC2 in slot 1 and the nondesignated router is the MSFC2 in slot 2; the active Supervisor Engine is in slot 1 and the standby Supervisor Engine is in slot 2.

Step 1 Enter the **`show version`** command to ensure that both Supervisor Engines are running Supervisor Engine software release 6.3(1) or later.

Step 2 Enter the **`set system highavailability enable`** command to enable High Availability on the active Supervisor Engine. Enter the **`show system highavailability`** command to verify that High Availability is enabled.

Step 3 If you have a console connection, enter the **`switch console`** command to access the designated router. If connected through a Telnet session, enter the **`session`** *`mod`* command to access the designated router.

Step 4 Copy the Cisco IOS Release 12.1(8a)E2 or later image to the bootflash of the designated router and nondesignated router.

Step 5 Set the boot image and configuration register on the designated router and nondesignated router to boot the new image on a reload. For the designated router, enter **`boot system flash bootflash:`***`image_name`* and ensure that this image is the first in the boot list. Clear any existing **`boot system`** commands that appear in the running configuration (**`show running-config`**) using the **`no`** form of the **`boot system`** command. For the nondesignated router, set the configuration register to auto boot by entering **`config-register 0x102`**.

Step 6 Enter the **`reload`** command to reload the designated router and nondesignated router. (If you already have SRM-capable Cisco IOS images loaded, you do not need to perform Step 6.)

Step 7 Disable configuration synchronization (**`config-sync`**) on the designated router using the **`no`** form of the command. Enter the **`write memory`** command. This lets you have access to configuration mode on both designated and nondesignated routers.

Step 8 Enable SRM on the designated router first, and then enable SRM on the nondesignated router as follows:

```
Router(config)#redundancy
Router(config-r)#high-availability
Router(config-r-ha)#single-router-mode
```

Note With Cisco IOS Releases 12.1(11b)E and later, you can specify the transition time that the newly designated router waits before downloading the new Layer 3 switching information to the Supervisor Engine switch processor. For configuration details, see the "Specifying the Transition Time on the Newly Designated Active Router" section.

Step 9 Enter the **`write memory`** command on the designated router to ensure that the nondesignated router's startup configuration has SRM enabled.

Step 10 Enter the **`show startup-config`** command on the nondesignated router to ensure that the nondesignated router has the following configuration statements:

```
redundancy
high-availability
single-router-mode
```

Step 11 Enter the **`show redundancy`** command on the designated router and nondesignated router to ensure that both have the following configuration statement:

```
Single Router Mode RuntimeStatus: enabled
```

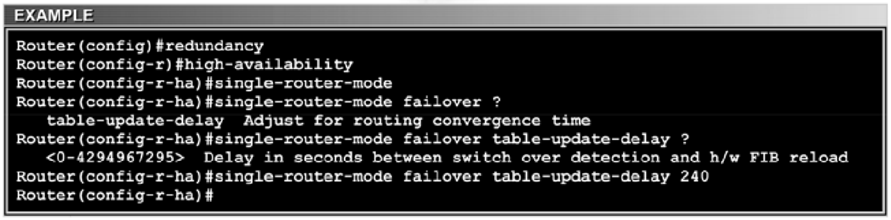
If not, repeat steps 9 and 10 allowing sufficient time between steps.

Step 12 Enter the **`reload`** command to reload the nondesignated router. When asked whether or not the configuration should be saved, enter **`no`**.

This display summarizes the above configuration commands used on the designated router and nondesignated router to enable SRM redundancy:

```
Time   Designated Router                       Nondesignated Router
  ----   ---                                     ----
  t0:    conf t->red->hi->no config-sync
  t1:                                            conf t->red->hi->no config-sync
  t2:    conf t->red->hi->single-router-mode
  t3:                                            conf t->red->hi->single-router-m
  t4:    write mem
  t5:                                            reload
```

Specifying the Transition Time on the Newly Designated Active Router



EXAMPLE

```
Router(config)#redundancy
Router(config-r)#high-availability
Router(config-r-ha)#single-router-mode
Router(config-r-ha)#single-router-mode failover ?
   table-update-delay  Adjust for routing convergence time
Router(config-r-ha)#single-router-mode failover table-update-delay ?
   <0-4294967295>  Delay in seconds between switch over detection and h/w FIB reload
Router(config-r-ha)#single-router-mode failover table-update-delay 240
Router(config-r-ha)#
```

With Cisco IOS Releases prior to release 12.1(11b)E, the transition time was 120 seconds, and was not configurable. Because of differences in routing convergence times, 120 seconds might not be long enough; older Layer 3 switching entries might be erased and the newly downloaded Layer 3 switching information might be incomplete.

With Cisco IOS Releases 12.1(11b)E and later, you can specify the transition time that the newly designated router waits before downloading the new Layer 3 switching information to the switch processor. On switchover, the old Layer 3 switching information is used for a configurable number of seconds before the new Layer 3 switching information is downloaded to the switch processor.

If nonstop forwarding is required, we do not recommend setting the transition time to a value lower than the default (120 seconds). At a minimum, it takes 30 to 60 seconds for routes to converge.

To specify the transition time, enter these commands (in this example the transition time is set to 240 seconds):

```
Router(config)#redundancy
Router(config-r)#high-availability
Router(config-r-ha)#single-router-mode
Router(config-r-ha)#single-router-mode failover ?
   table-update-delay  Adjust for routing convergence time
Router(config-r-ha)#single-router-mode failover table-update-delay ?
   <0-4294967295>  Delay in seconds between switch over detection and h/w FIB
reload
Router(config-r-ha)#single-router-mode failover table-update-delay 240
Router(config-r-ha)#
```

To set the transition time to the 2-minute default, use the **no** form of the command as follows:

```
Router(config-r-ha)#no single-router-mode failover table-update-delay
```

Display the transition time as follows:

```
Router-16#show redundancy
 Designated Router: 2 Non-designated Router: 1

 Redundancy Status: designated

 Config Sync AdminStatus  : enabled

 Config Sync RuntimeStatus: enabled

 Single Router Mode AdminStatus  : enabled

 Single Router Mode RuntimeStatus: enabled

 Single Router Mode transition timer : 240 seconds    <---- transition time

Router-16#
```

Upgrading Images with Single Router Mode Enabled

This section describes how to upgrade the Cisco IOS image on the active and standby MSFC when SRM is running. The new image name is c6msfc2-jsv-mz.9E. The standby MSFC cannot load an image using TFTP, but it can load an image from the Supervisor Engine Flash PC card (sup-slot0:).

To upgrade the images, perform these steps:

Step 1 On the active Supervisor Engine, enter the **`copy tftp sup-slot0:`** command and follow the prompts to load the new (c6msfc2-jsv-mz.9E) image onto the Supervisor Engine Flash PC card.

Step 2 If you have a console connection, enter the **`switch console`** command to access the active MSFC. If you are connected through a Telnet session, enter the **`session`** *`mod`* command to access the active MSFC.

Step 3 On the active MSFC, copy the new image from the Supervisor Engine Flash PC card to the MSFC bootflash as follows:

```
copy sup-slot0:c6msfc2-jsv-mz.9E bootflash:c6msfc2-jsv-mz.9E
```

Step 4 On the standby MSFC, copy the new image from the Supervisor Engine Flash PC card to the MSFC bootflash as follows:

```
copy sup-slot0:c6msfc2-jsv-mz.9E bootflash:c6msfc2-jsv-mz.9E
```

Step 5 On the active MSFC, specify that the new image is booted when the MSFC is reloaded as follows:

```
boot system flash bootflash:c6msfc2-jsv-mz.9E
```

Step 6 On the active MSFC, enter the **`write memory`** command to ensure that the standby MSFC startup configuration gets the boot information.

Step 7 Enter the **`reload`** command to reload the standby MSFC.

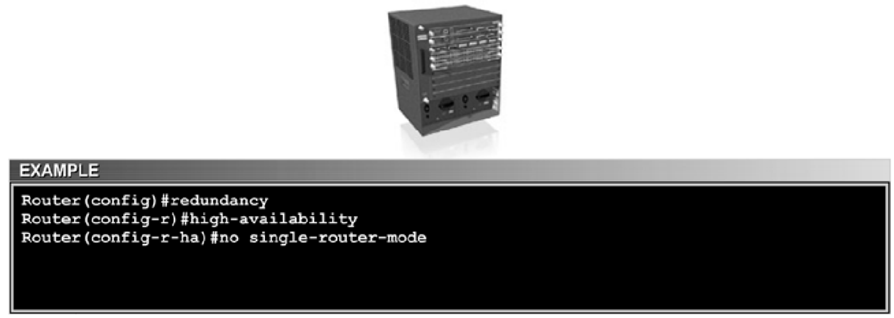
Step 8 Enter the **`show redundancy`** command on the active and standby MSFCs to ensure that both have the following configuration statement:

Single Router Mode RuntimeStatus: enabled

Step 9 Enter the **`reload`** command to reload the active MSFC.

Both MSFCs are now running the c6msfc2-jsv-mz.9E image.

Getting Out of Single Router Mode



EXAMPLE

```
Router(config)#redundancy
Router(config-r)#high-availability
Router(config-r-ha)#no single-router-mode
```

Note If you saved a copy of the running configuration used in dual router mode before configuring SRM redundancy, you do not need to use the procedure in this section. To get out of SRM redundancy and back to dual router mode, enter the following command on both MSFCs :copy bootflash:nosrm_dual_router_config startup-config. After the configurations are copied, reload the MSFCs using the reload command.

To get out of SRM, perform these steps:

Step 1 On the designated router, disable SRM using the **no** form of the command as follows:

```
Router(config)#redundancy
Router(config-r)#high-availability
Router(config-r-ha)#no single-router-mode
```

Step 2 Enter the **write memory** command on the designated router and nondesignated router.

Step 3 Enter the **show startup-config** command on the designated and nondesignated routers to ensure that "single-router mode" is not in the startup configuration.

Step 4 Enter the **reload** command to reload the designated router and nondesignated router.

SRM is now disabled on the designated router and nondesignated router.

Lesson Summary

This lesson accomplished the following:

- Described the Layer 3 redundancy features offered on the Catalyst 6500
- Described the use of the Hot Standby Routing Protocol (HSRP)
- Configured HSRP on the MSFCs to provide default gateway redundancy for clients
- Configured High Availability redundancy on the Supervisor Engines

Next Steps

After completing this lesson, go to:

- Multicast Routing

References

For additional information, refer to these resources:

- Configuring HSRP: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt1/1cdip.htm#xtocid23

Multicast Routing

Overview

Traditional IP communication allows a host to send packets to a single host (*unicast*) or to all hosts (*broadcast*). IP multicast provides a third transmission mechinism, allowing a host to send packets to a subset of all hosts (*multicast*).

Importance

Traditional network computing applications involve communication between two computers. However, some important emerging applications such as LAN TV, desktop conferencing, corporate broadcasts, and collaborative computing require simultaneous communication between groups of computers. These are the types of applications that IP Multicast was designed to support. Using IP Multicast with types of applications can greatly improve their performance and reduce the strain they place on the network.

Objectives

Upon completing this lesson, you will be able to:

- Describe Multicast Routing
- Explain the differences between unicast, broadcast, and multicast routing
- Configure multicast routing at Layer 2 and Layer 3 on the Catalyst 6500

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Have a basic understanding of IP Multicast concepts
- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course of have the equivalent knowledge

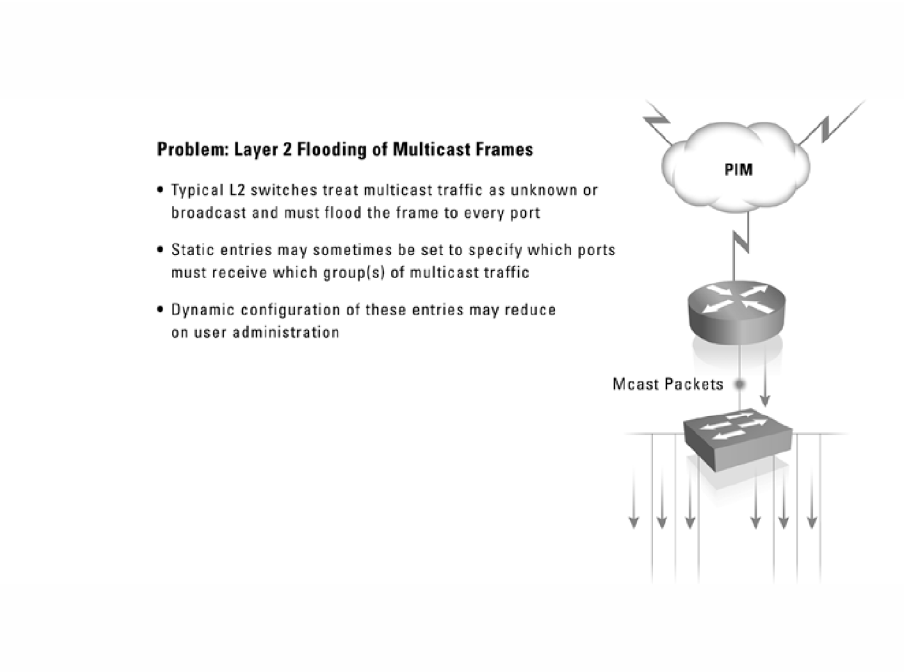
Outline

This lesson includes these sections:

- Overview
- Multicasting and Multicast Services Overview
- Configuring IGMP Snooping
- GMRP Software Requirements
- Configuring Multicast Router Ports and Group Entries
- Summary

Multicasting and Multicast Services Overview

This section provides an overview of IP Multicasting, including an overview of which IP Multicasting services are supported on the Catalyst 6500.



Problem: Layer 2 Flooding of Multicast Frames

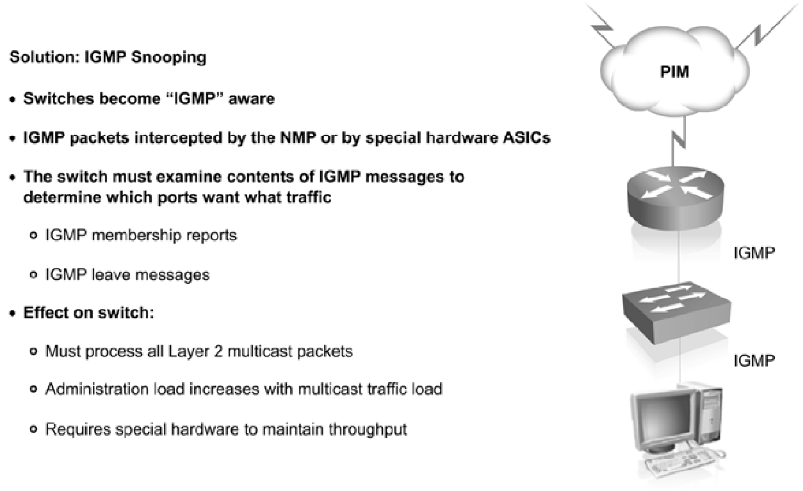
- Typical L2 switches treat multicast traffic as unknown or broadcast and must flood the frame to every port
- Static entries may sometimes be set to specify which ports must receive which group(s) of multicast traffic
- Dynamic configuration of these entries may reduce on user administration

IGMP snooping manages multicast traffic in switches by allowing directed switching of IP multicast traffic. GMRP is protocol independent and can manage both IP multicast traffic and any Layer 2 multicast traffic.

Switches can use IGMP snooping or GMRP to configure switch ports dynamically so that IP multicast traffic is forwarded only to those ports associated with IP multicast hosts. IGMP software components run on both the Cisco router and the switch.

You can statically configure multicast groups using the set cam static command. Multicast groups learned through IGMP snooping are dynamic. If you specify group membership for a multicast group address, your static setting supersedes any automatic manipulation by IGMP snooping or GMRP. Multicast group membership lists can consist of both user-defined setting and setting learned through IGMP snooping or GMRP.

Understanding How IGMP Snooping Works



Solution: IGMP Snooping

- **Switches become "IGMP" aware**
- **IGMP packets intercepted by the NMP or by special hardware ASICs**
- **The switch must examine contents of IGMP messages to determine which ports want what traffic**
  - IGMP membership reports
  - IGMP leave messages
- **Effect on switch:**
  - Must process all Layer 2 multicast packets
  - Administration load increases with multicast traffic load
  - Requires special hardware to maintain throughput

IGMP snooping manages multicast traffic at Layer 2 on the Catalyst 6000 family switches by allowing directed switching of IP multicast traffic.

Switches can use IGMP snooping to configure Layer 2 interfaces dynamically so that IP multicast traffic is forwarded only to those interfaces associated with IP multicast devices.

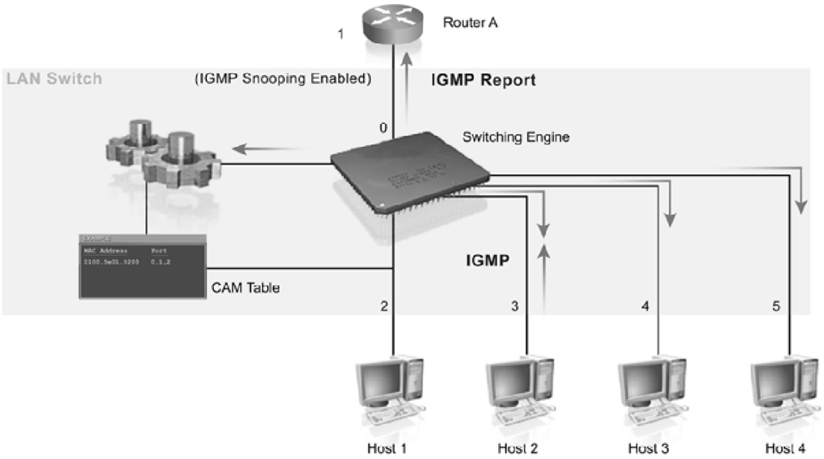
Catalyst 6000 switches can distinguish IGMP control traffic from multicast data traffic. When IGMP is enabled on the switch, IGMP control traffic is redirected to the CPU for further processing. This process is performed in hardware by specialized ASICs, which allow the switch to snoop IGMP control traffic with no performance penalty.

The route processor periodically sends out general queries to all VLANs, and as multicast receivers respond to the router's queries, the switch intercepts them. Only the first IGMP join (report), per VLAN and per IP multicast group, is forwarded to the router. Subsequent reports for the same VLAN and group are suppressed. The switch processor 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 port list of this forwarding table entry.

If a port is disabled, it will be removed from all multicast group entries.

Note You cannot enable IGMP snooping on a switch if GMRP is already enabled on the switch.

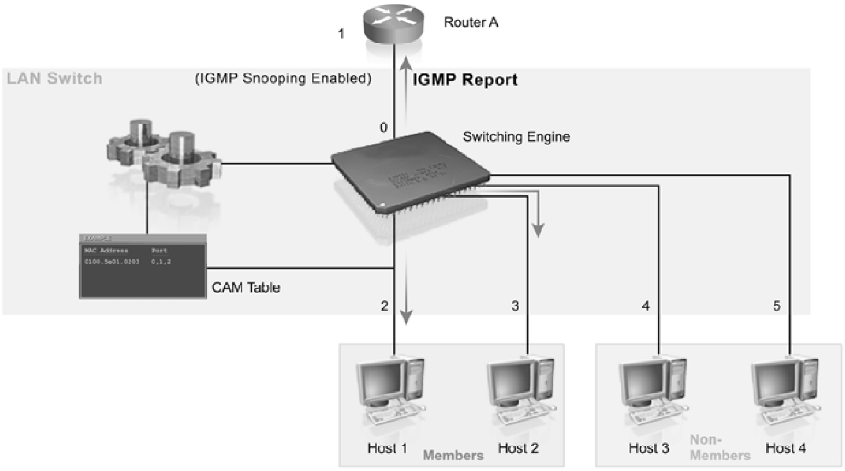
Joining a Multicast Group



When a host wants to join an IP multicast group, it sends an IGMP join (also known as a join message) specifying the IP multicast group it wants to join (for example, group 224.1.2.3). The switch hardware recognizes that the packet is an IGMP report and redirects it to the switch CPU. The switch installs a new group entry for 01-00-5e-01-02-03 and adds the host port and the router port to that entry. The switch then relays the join from the host to all multicast router ports. The designated multicast router for the segment adds the outgoing interface (OIF) to the outgoing interface list (OIL) for the group and begins forwarding multicast traffic for 224.1.2.3 to this segment.

When a second host in this VLAN wants to join group 244.1.2.3, it sends out an IGMP join for this group. The switch hardware recognizes that this is an IGMP control packet and redirects it to the switch CPU. Since the switch already has a group entry for 01-00-5e-01-02-03 in this VLAN, it just adds the second host port to the entry. Because this is not the first host joining the group, the switch suppresses the report (does not send it to the router).

Constraining Multicast Traffic



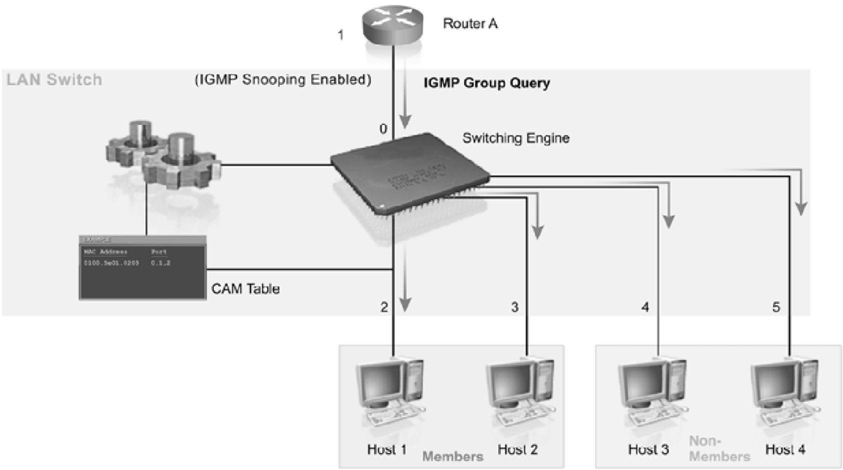
When a host sends multicast traffic to a group, the switch hardware does not recognize the stream as IGMP control packets, and therefore the packets are not redirected to the switch CPU. Instead the multicast traffic hits the MAC group entry and the switch constrains the traffic to only those ports that have been added to that group entry.

The router sends IGMP general queries every 60 seconds by default. The switch floods these queries on all ports in the VLAN, and hosts that are interested in a multicast group respond with an IGMP join for each group in which they are interested.

The switch intercepts these IGMP joins, and only the first join per VLAN, and per IP multicast group, is forwarded on the multicast router ports. Subsequent reports for the same VLAN and group are suppressed (not sent to the router).

Note	If there are CGMP switches in the network, join and leave suppression does not occur. In a network that has both IGMP and CGMP switches, all join and leave messages are forwarded to the multicast routers so that CGMP join and leave messages can be generated by the router.

Leaving a Multicast Group



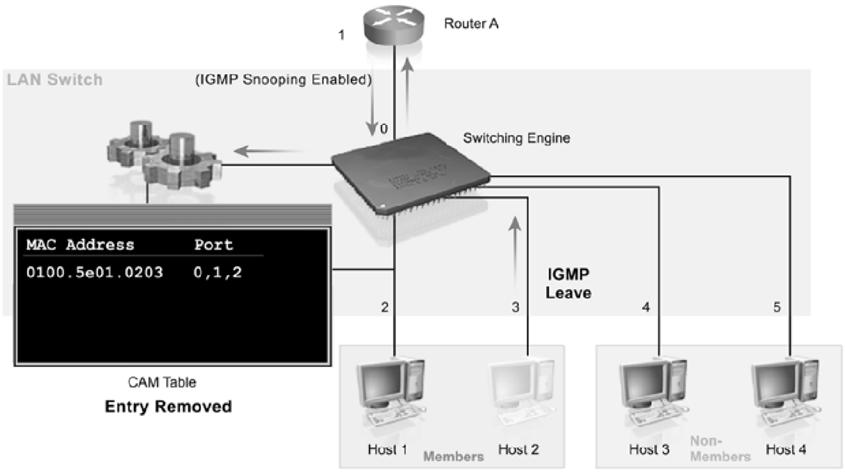
MAC Address	Port
0100.5e01.0203	0,1,2

The designated multicast router for a segment continues forwarding the multicast traffic to that VLAN as long as at least one host in the VLAN wishes to receive multicast traffic. When hosts want to leave a multicast group, they can either ignore the periodic general queries sent by the multicast router (IGMP v1 host behavior), or they can send an IGMP leave (IGMP v2 host behavior). When the switch receives a leave message, it sends out a MAC-based general query on the port on which it received the leave message to determine if any devices connected to this port are interested in traffic for the specific multicast group. If this port is the last port in the VLAN, the switch sends a MAC-based general query to all ports in the VLAN. MAC-based general queries are addressed to the Layer 2 Group Destination Address (GDA) MAC address for which the IGMP leave message was received. At Layer 3, the MAC-based general queries are addressed to 244.0.0.1 (all hosts), and in the IGMP header, the group address field is set to 0.0.0.0.

If no IGMP join is received for any of the IP multicast groups that map to the MAC multicast group address, the port is removed from the multicast forwarding entry. If the port is not the last non-multicast-router port in the entry, the switch suppresses the IGMP leave (does not send it to the router). If the port is the last non-multicast-router port in the entry, the IGMP leave is forwarded to the multicast router ports and the MAC group forwarding entry is removed.

When the router receives the IGMP leave, it sends several IGMP group-specific queries. If no join messages are received in response to the queries, and there are no downstream routers connected through that interface, the router removes the interface from the OIL for that IP multicast group entry in the multicast routing table.

IGMP Fast-Leave Processing

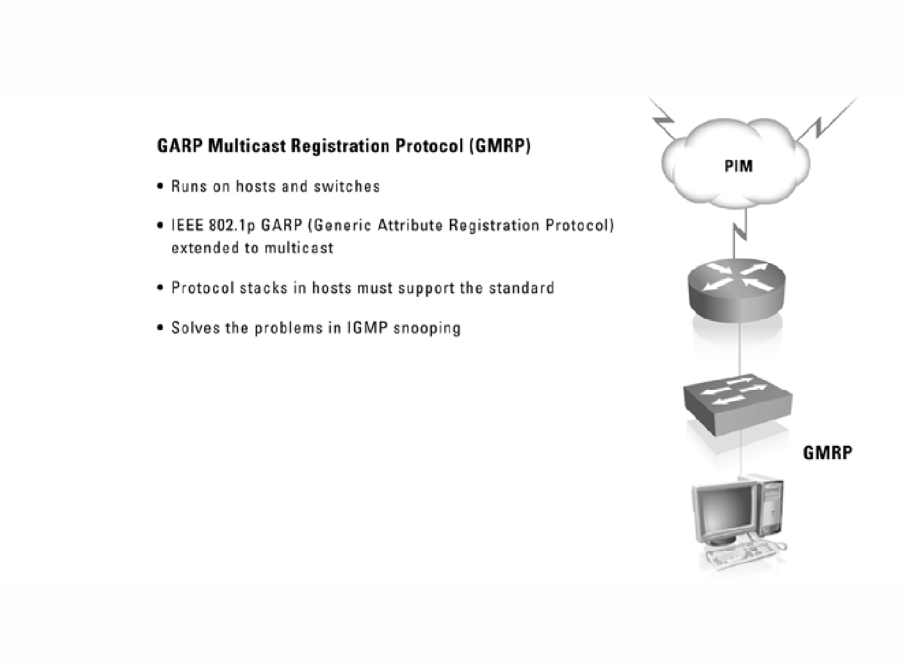


```
MAC Address      Port
0100.5e01.0203   0,1,2
```

IGMP snooping fast-leave processing allows the switch processor to remove an interface from the port list of a forwarding-table entry without first sending out a MAC-based general query on the port. When an IGMP leave is received on a port, the port is immediately removed from the multicast forwarding entry (or the entire entry is removed).

Note	Do not use the fast-leave processing feature if more than one host is connected to each port. If fast-leave is enabled when more than one host is connected to a port, some hosts might be dropped inadvertently. Fast leave is supported with IGMP version 2 hosts only.

Understanding How GMRP Works



GARP Multicast Registration Protocol (GMRP)

- Runs on hosts and switches
- IEEE 802.1p GARP (Generic Attribute Registration Protocol) extended to multicast
- Protocol stacks in hosts must support the standard
- Solves the problems in IGMP snooping

GMRP is a Generic Attribute Registration Protocol (GARP) application that provides a constrained multicast flooding facility similar to IGMP snooping. GMRP and GARP are industry-standard protocols defined by the IEEE. For detailed protocol operational information, refer to 802.1p.

GMRP software components run on both the switch and on the host. On the host, in an IP multicast environment, you must use IGMP with GMRP; the host GMRP software spawns Layer 2 GMRP versions of the host's Layer 3 IGMP control packets. The switch receives both the Layer 2 GMRP and the Layer 3 IGMP traffic from the host. The switch forwards the Layer 3 IGMP control packets to the router and uses the received GMRP traffic to constrain multicasts at Layer 2 in the host's VLAN.

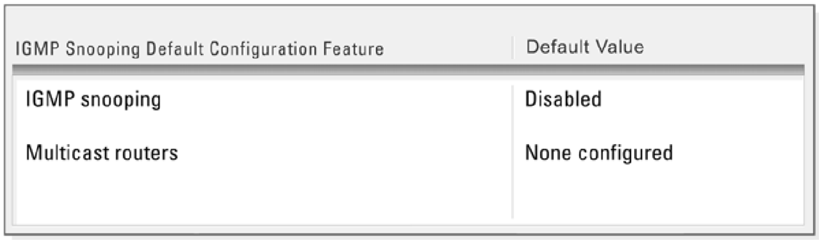
When a host wants to join an IP multicast group, it sends an IGMP join, which spawns a GMRP join. When the switch receives the GMRP join, it adds the port through which the join was received to the appropriate multicast group. The switch propagates the GMRP join to all other hosts in the VLAN, one of which is typically the multicast source. When the source is multicasting to the group, the switch forwards the multicast only to the ports from which it received join messages for the group.

The switch sends periodic GMRP queries. If a host wants to remain in a multicast group, it responds to the query and the switch does nothing. If a host does not want to remain in the multicast group, it can either send a leave message or not respond to the periodic queries from the switch. If the switch receives a leave message or receives no response from the host for the duration of the `leaveall` timer, the switch removes the host from the multicast group.

Note	To use GMRP in a routed environment, enable the GMRP forwardall option on all ports where routers are attached.

Configuring IGMP Snooping

This section details the configuration of IGMP Snooping on the Catalyst 6500.



IGMP Snooping Default Configuration Feature	Default Value
IGMP snooping	Disabled
Multicast routers	None configured

IGMP snooping allows switches to examine IGMP packets and make forwarding decisions based on their content.

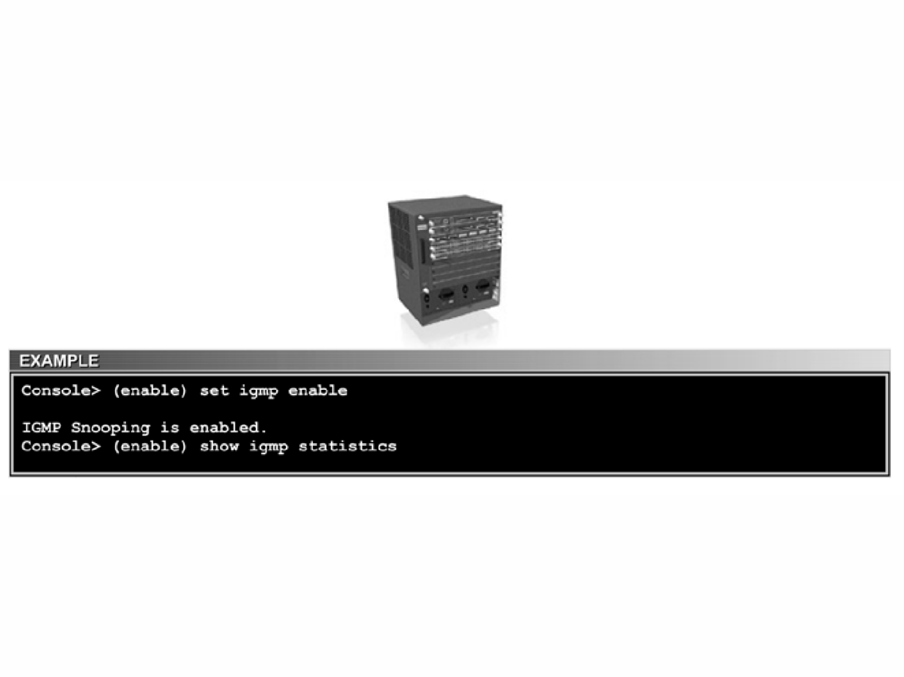
Note Quality of service (QoS) does not support IGMP traffic when IGMP snooping is enabled.

Note IGMP snooping is enabled by default in Supervisor Engine software releases 5.5(9) and later and 6.3(1) and later.

Table 3-18: IGMP Snooping Default Configuration Feature

	Default Value
IGMP snooping	Disabled
Multicast routers	None configured

Enabling IGMP Snooping



EXAMPLE

```
Console> (enable) set igmp enable

IGMP Snooping is enabled.
Console> (enable) show igmp statistics
```

Note You cannot enable IGMP snooping if GMRP is enabled.

To enable IGMP snooping, perform this task in privileged mode:

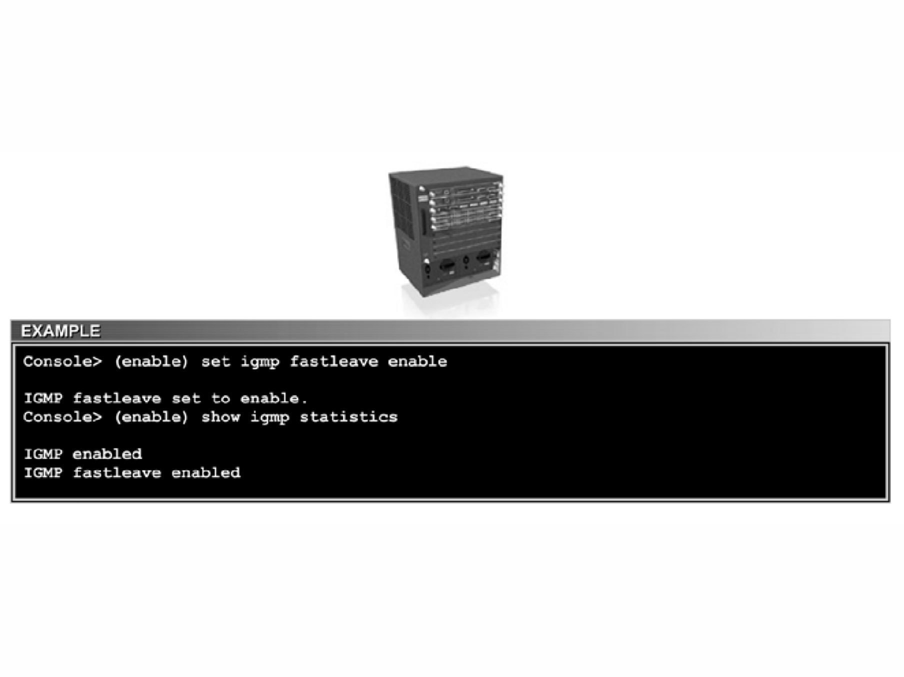
Table 3-19: IGMP Snooping

	Task	Command
Step 1	Enable IGMP snooping on the switch.	**`set igmp enable`**
Step 2	Verify that IGMP snooping is enabled.	**`show igmp statistics`** [*vlan*]

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

```
Console> (enable) set igmp enable

IGMP Snooping is enabled.
Console> (enable) show igmp statistics

IGMP enabled
IGMP statistics for vlan 1:
Total valid pkts rcvd:           18951
Total invalid pkts recvd         0
General Queries recvd            377
Group Specific Queries recvd     0
MAC-Based General Queries recvd  0
Leaves recvd                     14
Reports recvd                    16741
Queries  Xmitted                 0
GS Queries Xmitted               16
Reports Xmitted                  0
Leaves Xmitted                   0
Failures to add GDA to EARL      0
Topology Notifications rcvd      10
IGMP packets dropped             0
Console> (enable)
```

Enabling IGMP Fast-Leave Processing



EXAMPLE

```
Console> (enable) set igmp fastleave enable

IGMP fastleave set to enable.
Console> (enable) show igmp statistics

IGMP enabled
IGMP fastleave enabled
```

To enable IGMP fast-leave processing, perform this task in privileged mode:

Table 3-20: IGMP Fast-Leave

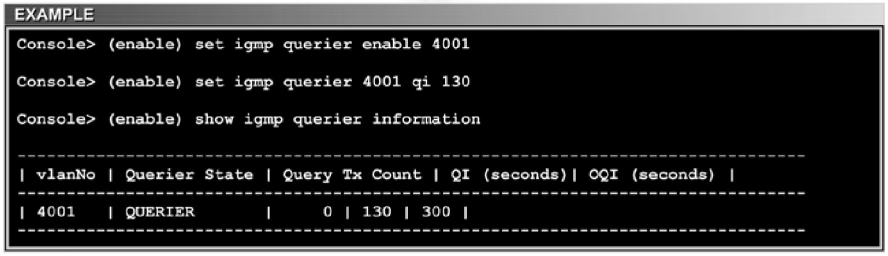
	Task	Command
Step 1	Enable IGMP fast-leave processing on the switch.	**`set igmp fastleave enable`**
Step 2	Verify that IGMP fast-leave processing is enabled.	**`show igmp statistics`**

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

```
Console> (enable) set igmp fastleave enable

IGMP fastleave set to enable.
Console> (enable) show igmp statistics

IGMP enabled
IGMP fastleave enabled

IGMP statistics for vlan 1:
Total valid pkts rcvd:          18951
Total invalid pkts recvd        0
General Queries recvd           377
Group Specific Queries recvd    0
```

```
MAC-Based General Queries recvd  0
Leaves recvd                     14
Reports recvd                    16741
Other Pkts recvd                 0
Queries  Xmitted                 0
GS Queries Xmitted               16
Reports Xmitted                  0
Leaves Xmitted                   0
Failures to add GDA to EARL      0
Topology Notifications rcvd      10
Console> (enable)
```

Enabling the IGMP Querier



EXAMPLE

```
Console> (enable) set igmp querier enable 4001

Console> (enable) set igmp querier 4001 qi 130

Console> (enable) show igmp querier information

---------------------------------------------------------------------------
| vlanNo | Querier State | Query Tx Count | QI (seconds)| OQI (seconds) |
---------------------------------------------------------------------------
| 4001   | QUERIER       |      0 | 130 | 300 |
---------------------------------------------------------------------------
```

Use the IGMP querier to support IGMP snooping within a VLAN where PIM and IGMP are not configured because the multicast traffic does not need to be routed.

Note You can enable the IGMP querier on all the switches in the VLAN. One switch is elected as querier.

To enable the IGMP querier in a VLAN, peform these tasks in privileged mode:

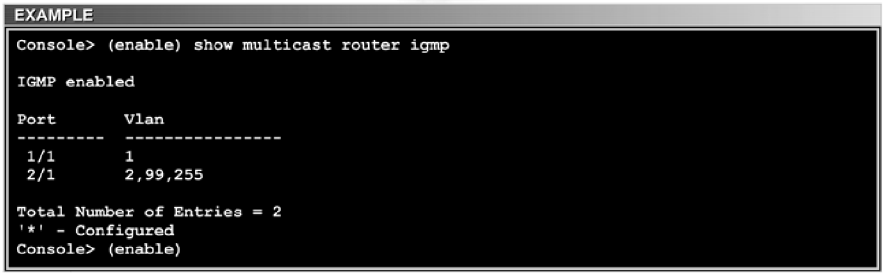
Table 3-21: IGMP Querier

Task	Command
Enable IGMP querier on a VLAN or on all VLANs.	**set igmp querier disable** | **enable** *vlan*
Specify the time interval between the general queries sent by the switch. The default is 125 seconds.	**set igmp querier** *vlan* **qi** *val*
Specify the amount of time the switch should wait before electing itself as the querier in the absence of general queries. The default is 300 seconds.	**set igmp querier** *vlan* **oqi** *val*
Display IGMP querier information.	**show igmp querier information**

This example shows how to enable IGMP querier and display querier information:

```
Console> (enable) set igmp querier enable 4001

Console> (enable) set igmp querier 4001 qi 130

Console> (enable) show igmp querier information

--------------------------------------------------------------------------------
| vlanNo | Querier State | Query Tx Count | QI (seconds)| OQI (seconds) |
--------------------------------------------------------------------------------
| 4001   | QUERIER       |     0    | 130       | 300       |
--------------------------------------------------------------------------------
```

Displaying Multicast Router Information



EXAMPLE

```
Console> (enable) show multicast router igmp

IGMP enabled

Port       Vlan
---------  ----------------
 1/1       1
 2/1       2,99,255

Total Number of Entries = 2
'*' - Configured
Console> (enable)
```

When you enable IGMP snooping, the switch automatically learns to which ports a multicast router is connected.

To display the dynamically learned multicast router information, perform these tasks in privileged mode:

Table 3-22: show multicast router

Task	Command
Display information on dynamically learned and manually configured multicast router ports.	**show multicast router** [*mod/port*] [*vlan_id*]
Display information only on those multicast router ports learned dynamically using IGMP snooping.	**show multicast router igmp** [mod/port] [*vlan_id*]

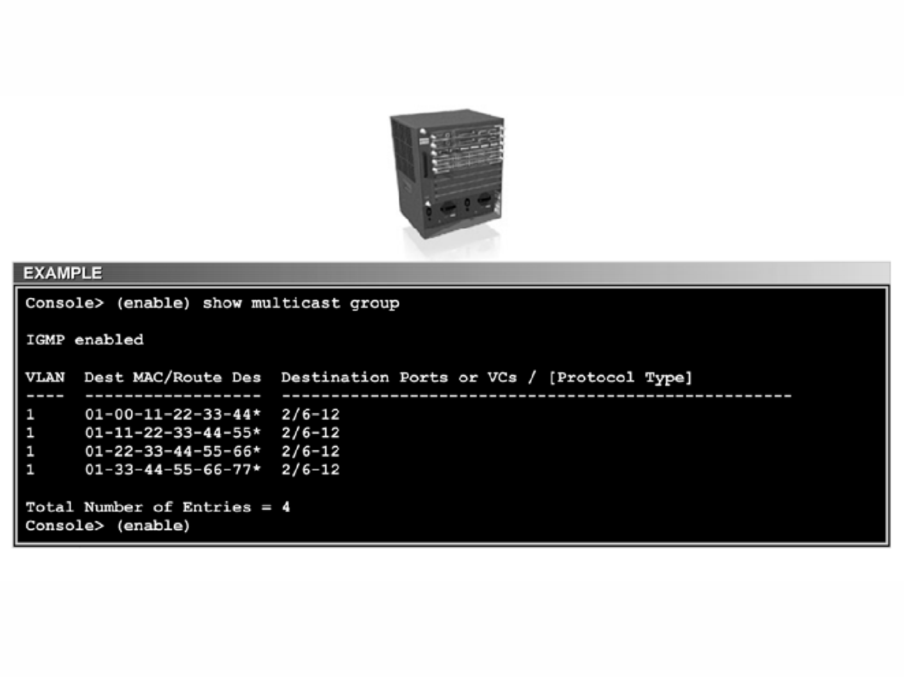
This example shows how to display information on all multicast router ports (the asterisk [*] next to the multicast router on port 5/7 indicates that the entry was configured manually):

```
Console> (enable) show multicast router

IGMP enabled

Port       Vlan
---------  ----------------
 1/1       1
 2/1       2,99,255
 5/7    *  99

Total Number of Entries = 3
'*' - Configured
Console> (enable)
```

This example shows how to display only those multicast router ports that were learned dynamically through IGMP:

```
Console> (enable) show multicast router igmp

IGMP enabled

Port       Vlan
---------  ----------------
 1/1       1
 2/1       2,99,255

Total Number of Entries = 2
'*' - Configured
Console> (enable)
```

Displaying Multicast Group Information



EXAMPLE

```
Console> (enable) show multicast group

IGMP enabled

VLAN  Dest MAC/Route Des  Destination Ports or VCs / [Protocol Type]
----  ------------------  ----------------------------------------------------
1     01-00-11-22-33-44*  2/6-12
1     01-11-22-33-44-55*  2/6-12
1     01-22-33-44-55-66*  2/6-12
1     01-33-44-55-66-77*  2/6-12

Total Number of Entries = 4
Console> (enable)
```

To display information about multicast groups, perform these tasks in privileged mode:

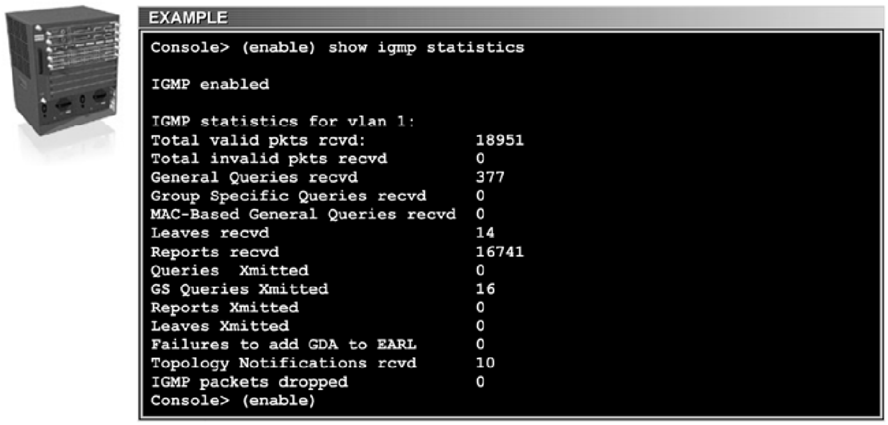
Table 3-23: show multicast group

Task	Command
Display information about multicast groups.	**show multicast group** [*mac_addr*] [vlan_id]
Display information only about multicast groups learned dynamically through IGMP.	**show multicast group igmp** [*mac_addr*] **[vlan_id]**
Display the total number of multicast addresses (groups) in each VLAN.	**show multicast group count** [*vlan_id*]
Display the total number of multicast addresses (groups) in each VLAN that were learned dynamically through IGMP.	**show multicast group count igmp** [*vlan_id*]

This example shows how to display information about all multicast groups on the switch:

```
Console> (enable) show multicast group

IGMP enabled

VLAN  Dest MAC/Route Des  Destination Ports or VCs / [Protocol Type]
----  ------------------  ----------------------------------------------------
1     01-00-11-22-33-44*  2/6-12
1     01-11-22-33-44-55*  2/6-12
1     01-22-33-44-55-66*  2/6-12
1     01-33-44-55-66-77*  2/6-12

Total Number of Entries = 4
Console> (enable)
```

Displaying IGMP Snooping Statistics



```
Console> (enable) show igmp statistics

IGMP enabled

IGMP statistics for vlan 1:
Total valid pkts rcvd:           18951
Total invalid pkts recvd         0
General Queries recvd            377
Group Specific Queries recvd     0
MAC-Based General Queries recvd  0
Leaves recvd                     14
Reports recvd                    16741
Queries  Xmitted                 0
GS Queries Xmitted               16
Reports Xmitted                  0
Leaves Xmitted                   0
Failures to add GDA to EARL      0
Topology Notifications rcvd      10
IGMP packets dropped             0
Console> (enable)
```

To display IGMP snooping statistics on the switch, perform this task:

Table 3-24: show igmp

Task	Command
Display IGMP snooping statistics.	**`show igmp statistics`** [*`vlan_id`*]

This example shows how to display IGMP snooping statistics:

```
Console> (enable) show igmp statistics

IGMP enabled

IGMP statistics for vlan 1:
Total valid pkts rcvd:           18951
Total invalid pkts recvd         0
General Queries recvd            377
Group Specific Queries recvd     0
MAC-Based General Queries recvd  0
Leaves recvd                     14
Reports recvd                    16741
Queries  Xmitted                 0
GS Queries Xmitted               16
Reports Xmitted                  0
Leaves Xmitted                   0
Failures to add GDA to EARL      0
Topology Notifications rcvd      10
IGMP packets dropped             0
Console> (enable)
```

Disabling IGMP Fast-Leave Processing

To disable IGMP fast-leave processing, perform this task in privileged mode:

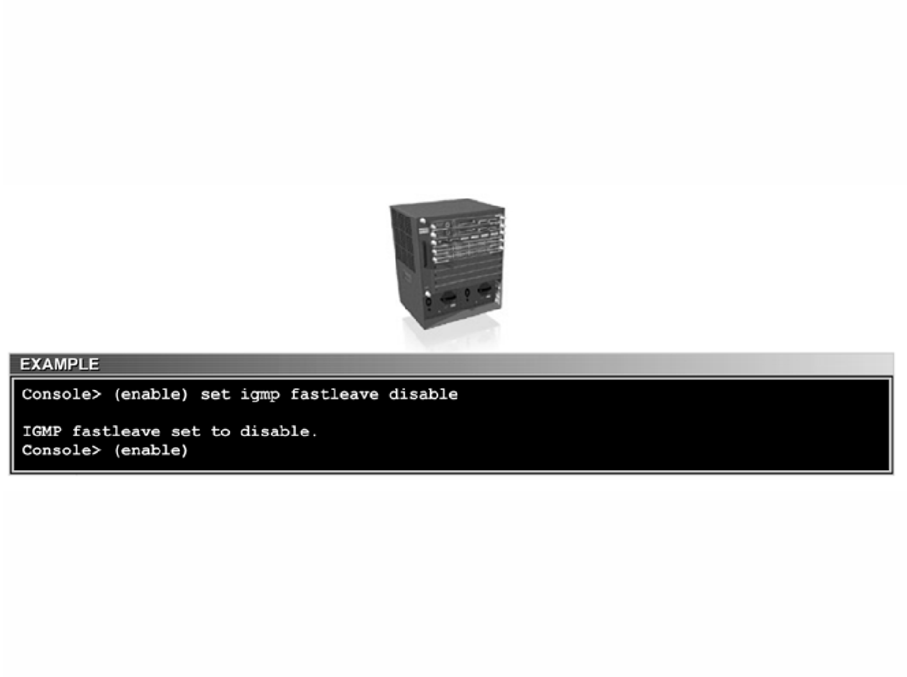
Table 3-25: Disable IGMP Fast-Leave

Task	Command
Disable IGMP fast-leave processing on the switch.	**`set igmp fastleave disable`**

This example shows how to disable IGMP fast-leave processing on the switch:

```
Console> (enable) set igmp fastleave disable

IGMP fastleave set to disable.
Console> (enable)
```

Disabling IGMP Snooping



EXAMPLE

```
Console> (enable) set igmp fastleave disable

IGMP fastleave set to disable.
Console> (enable)
```

To disable IGMP snooping on the switch, perform this task in privileged mode:

Table 3-26: Disable IGMP Snooping

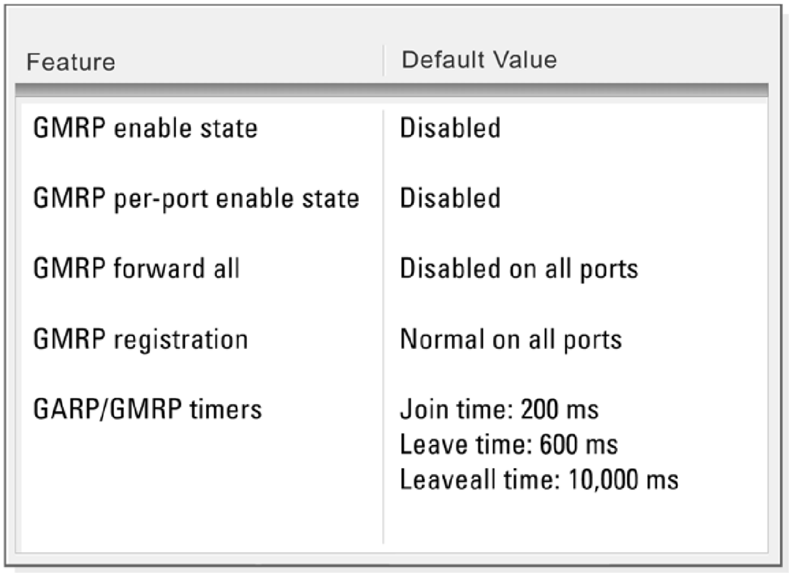
Task	Command
Disable IGMP fast-leave processing on the switch.	**`set igmp disable`**

This example shows how to disable IGMP snooping:

```
Console> (enable) set igmp disable

IGMP feature for IP multicast disabled
Console> (enable)
```

GMRP Software Requirements

GMRP requires Supervisor Engine software release 5.2 or later.



Feature	Default Value
GMRP enable state	Disabled
GMRP per-port enable state	Disabled
GMRP forward all	Disabled on all ports
GMRP registration	Normal on all ports
GARP/GMRP timers	Join time: 200 ms Leave time: 600 ms Leaveall time: 10,000 ms

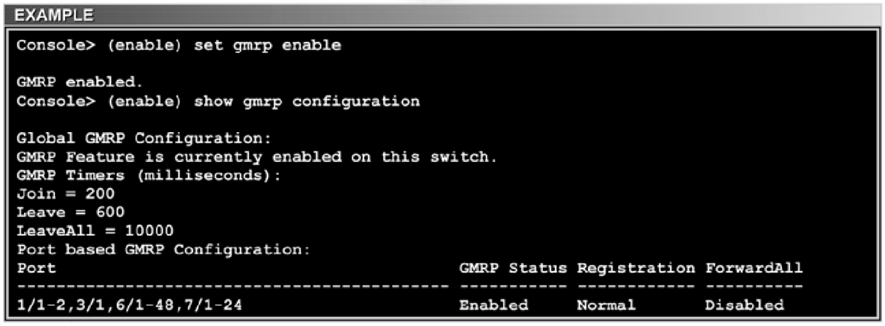
Default GMRP Configuration

The following table shows the default GMRP configuration.

Table 3-27: Default GMRP Configuration

Feature	Default Value
GMRP enable state	Disabled
GMRP per-port enable state	Disabled
GMRP forward all	Disabled on all ports
GMRP registration	Normal on all ports
GARP/GMRP timers	Join time: 200 ms Leave time: 600 ms Leaveall time: 10,000 ms

Enabling GMRP Globally



EXAMPLE

```
Console> (enable) set gmrp enable

GMRP enabled.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
Port                                         GMRP Status Registration ForwardAll
-------------------------------------------- ----------- ------------ ----------
1/1-2,3/1,6/1-48,7/1-24                      Enabled     Normal       Disabled
```

Note You cannot enable GMRP if IGMP snooping is enabled.

To enable GMRP globally, perform this task in privileged mode:

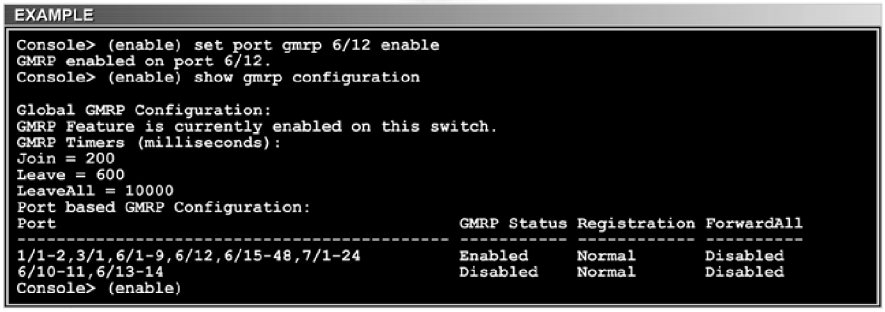
Table 3-28: Enable GMRP

	Task	Command
Step 1	Enable GMRP globally on the switch.	`set gmrp enable`
Step 2	Verify the configuration.	`show gmrp configuration`

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

```
Console> (enable) set gmrp enable

GMRP enabled.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
Port                                         GMRP Status Registration ForwardAll
-------------------------------------------- ----------- ------------ ----------
1/1-2,3/1,6/1-48,7/1-24                      Enabled     Normal       Disabled
Console> (enable)
```

Enabling GMRP on Individual Switch Ports



EXAMPLE

```
Console> (enable) set port gmrp 6/12 enable
GMRP enabled on port 6/12.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
Port                                          GMRP Status Registration ForwardAll
--------------------------------------------- ----------- ------------ ----------
1/1-2,3/1,6/1-9,6/12,6/15-48,7/1-24           Enabled     Normal       Disabled
6/10-11,6/13-14                               Disabled    Normal       Disabled
Console> (enable)
```

Note	You can change the per-port GMRP configuration regardless of whether GMRP is enabled globally. However, GMRP will not function on any ports until you enable it globally.

To enable GMRP on individual switch ports, perform this task in privileged mode:

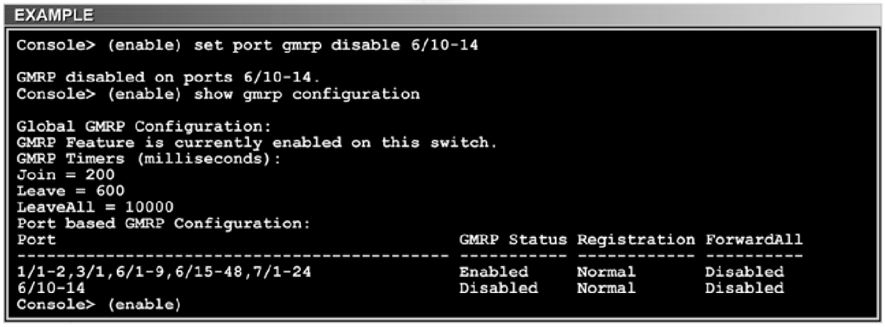
Table 3-29: GMRP Port

	Task	Command
Step 1	Enable GMRP on an individual switch port.	**`set port gmrp`** *`mod/port`* **`enable`**
Step 2	Verify the configuration.	**`show gmrp configuration`**

This example shows how to enable GMRP on port 6/12 and verify the configuration:

```
Console> (enable) set port gmrp enable 6/12
GMRP enabled on port 6/12.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
Port                                         GMRP Status Registration ForwardAll
-------------------------------------------- ----------- ------------ ----------
1/1-2,3/1,6/1-9,6/12,6/15-48,7/1-24          Enabled     Normal       Disabled
6/10-11,6/13-14                              Disabled    Normal       Disabled
Console> (enable)
```

Disabling GMRP on Individual Switch Ports



EXAMPLE

```
Console> (enable) set port gmrp disable 6/10-14

GMRP disabled on ports 6/10-14.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
Port                                         GMRP Status Registration ForwardAll
-------------------------------------------- ----------- ------------ ----------
1/1-2,3/1,6/1-9,6/15-48,7/1-24               Enabled     Normal       Disabled
6/10-14                                      Disabled    Normal       Disabled
Console> (enable)
```

Note	You can change the per-port GMRP configuration regardless of whether GMRP is enabled globally. However, GMRP will not function on any ports until you enable it globally.

To disable GMRP on individual switch ports, perform this task in privileged mode:

Table 3-30: GMRP Port

	Task	Command
Step 1	Disable GMRP on individual switch ports.	**`set port gmrp disable`** *`mod/port`*
Step 2	Verify the configuration.	**`show gmrp configuration`**

This example shows how to disable GMRP on ports 6/10-14 and verify the configuration:

```
Console> (enable) set port gmrp disable 6/10-14

GMRP disabled on ports 6/10-14.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
Port                                         GMRP Status Registration ForwardAll
-------------------------------------------- ----------- ------------ ----------
1/1-2,3/1,6/1-9,6/15-48,7/1-24               Enabled     Normal       Disabled
6/10-14                                      Disabled    Normal       Disabled
Console> (enable)
```

Enabling GMRP Forward-All Option

```
Console> (enable) set gmrp fwdall enable 1/1

GMRP Forward All groups option enabled on port 1/1.
Console> (enable)
```

When you enable the GMRP forward-all option on a port, a copy of all multicast traffic registered on the switch is forwarded to that port. Enable the forward-all option on any port connected to a router that needs to receive any multicasts (routers do not support GMRP and so cannot send GMRP join messages). The forward-all option can also be used to forward all registered multicast traffic to a port with a network analyzer or probe attached.

To enable the GMRP forward-all option on a switch port, perform this task in privileged mode:

Table 3-31: GMRP Forward-All

Task	Command
Enable the GMRP forward-all option on a switch port.	**`set gmrp fwdall enable`** *`mod/port`*

This example shows how to enable the GMRP forward-all option on port 1/1:

```
Console> (enable) set gmrp fwdall enable 1/1

GMRP Forward All groups option enabled on port 1/1.
Console> (enable)
```

Disabling GMRP Forward-All Option

EXAMPLE

```
Console> (enable) set gmrp fwdall disable 1/1

GMRP Forward All groups option disabled on port 1/1.
Console> (enable)
```

To disable the GMRP forward-all option on a port, perform this task in privileged mode:

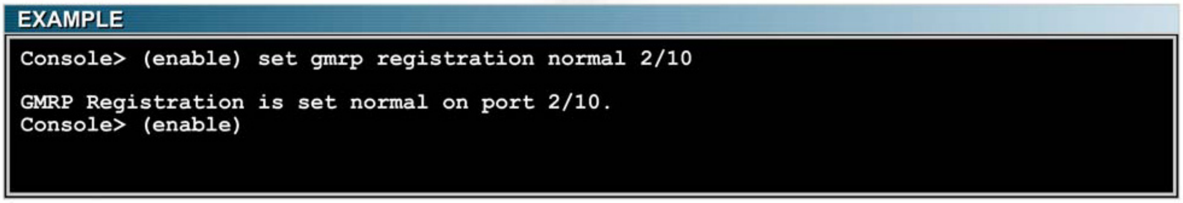
Table 3-32: Disabling GMRP Forward-All

Task	Command
Disable the GMRP forward-all option on a port.	**`set gmrp fwdall disable`** *`mod/port`*

This example shows how to disable the GMRP forward-all option on port 1/1:

```
Console> (enable) set gmrp fwdall disable 1/1

GMRP Forward All groups option disabled on port 1/1.
Console> (enable)
```

Configuring GMRP Registration



EXAMPLE

```
Console> (enable) set gmrp registration normal 2/10

GMRP Registration is set normal on port 2/10.
Console> (enable)
```

Setting Normal Registration

Configuring a port in **`normal`** registration mode allows dynamic GMRP multicast registration and deregistration on the port. Normal mode is the default on all switch ports.

To set normal registration on a port, perform this task in privileged mode:

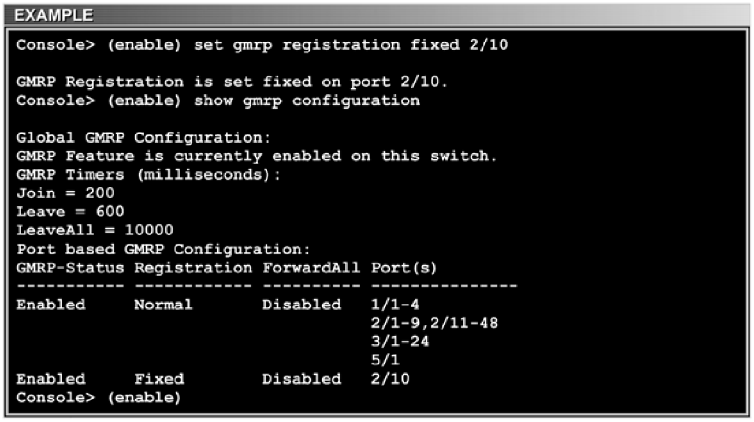
Table 3-33: GMRP Registration

	Task	Command
Step 1	Set normal registration on a port.	**set gmrp registration normal** *mod/port*
Step 2	Verify the configuration.	**show gmrp configuration**

This example shows how to set normal registration on port 2/10:

```
Console> (enable) set gmrp registration normal 2/10


GMRP Registration is set normal on port 2/10.
Console> (enable)
```

Setting Fixed Registration



EXAMPLE

```
Console> (enable) set gmrp registration fixed 2/10

GMRP Registration is set fixed on port 2/10.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
GMRP-Status Registration ForwardAll Port(s)
----------- ------------ ---------- ---------------
Enabled     Normal       Disabled   1/1-4
                                    2/1-9,2/11-48
                                    3/1-24
                                    5/1
Enabled     Fixed        Disabled   2/10
Console> (enable)
```

When you configure a port in fixed registration mode, all the multicast groups currently registered on all ports are registered on the port, but the port ignores any subsequent registrations or deregistrations on other ports. A port in **fixed** registration mode continues to register multicast groups that are specific to the port. You must return the port to **normal** registration mode to deregister multicast groups on the port.

To set fixed registration on a port, perform this task in privileged mode:

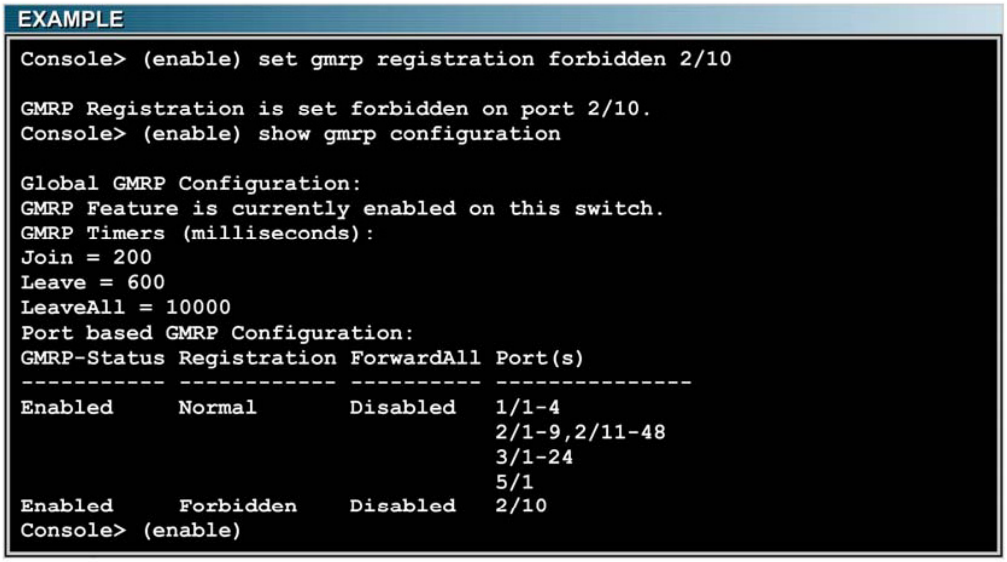
Table 3-34: GMRP Fixed Registration

	Task	Command
Step 1	Set fixed registration on a port.	**`set gmrp registration fixed`** *`mod/port`*
Step 2	Verify the configuration.	**`show gmrp configuration`**

This example shows how to set fixed registration on port 2/10 and verify the configuration:

```
Console> (enable) set gmrp registration fixed 2/10

GMRP Registration is set fixed on port 2/10.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
GMRP-Status Registration ForwardAll Port(s)
----------- ------------ ---------- --------------------------------------------
Enabled     Normal       Disabled   1/1-4
                                    2/1-9,2/11-48
                                    3/1-24
                                    5/1
Enabled     Fixed        Disabled   2/10
Console> (enable)
```

Setting Forbidden Registration



EXAMPLE

```
Console> (enable) set gmrp registration forbidden 2/10

GMRP Registration is set forbidden on port 2/10.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
GMRP-Status Registration ForwardAll Port(s)
----------- ------------ ---------- ---------------
Enabled     Normal       Disabled   1/1-4
                                    2/1-9,2/11-48
                                    3/1-24
                                    5/1
Enabled     Forbidden    Disabled   2/10
Console> (enable)
```

Setting a port in forbidden registration mode deregisters all GMRP multicasts and prevents any further GMRP multicast registration on the port.

To set forbidden registration on a port, perform this task in privileged mode:

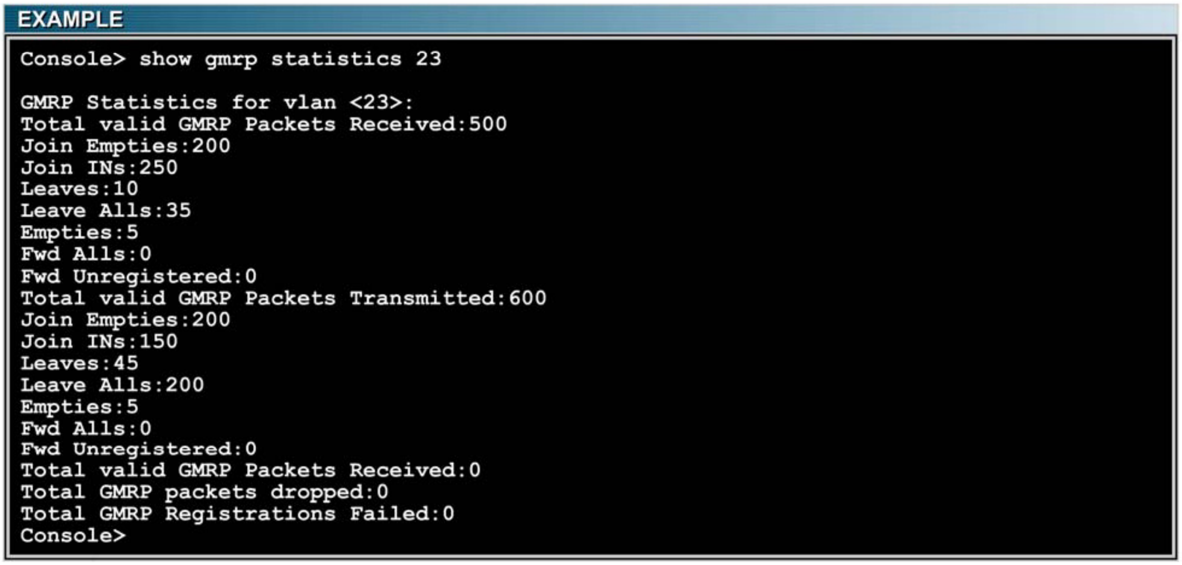
Table 3-35: GMRP Forbidden Registration

	Task	Command
Step 1	Set forbidden registration on a port.	**`set gmrp registration forbidden`** *`mod/port`*
Step 2	Verify the configuration.	**`show gmrp configuration`**

This example shows how to set forbidden registration on port 2/10 and verify the configuration:

```
Console> (enable) set gmrp registration forbidden 2/10

GMRP Registration is set forbidden on port 2/10.
Console> (enable) show gmrp configuration

Global GMRP Configuration:
GMRP Feature is currently enabled on this switch.
GMRP Timers (milliseconds):
Join = 200
Leave = 600
LeaveAll = 10000
Port based GMRP Configuration:
GMRP-Status Registration ForwardAll Port(s)
----------- ------------ ---------- --------------------------------------------
Enabled     Normal       Disabled   1/1-4
                                    2/1-9,2/11-48
                                    3/1-24
                                    5/1
Enabled     Forbidden    Disabled   2/10
Console> (enable)
```

Displaying GMRP Statistics



```
Console> show gmrp statistics 23

GMRP Statistics for vlan <23>:
Total valid GMRP Packets Received:500
Join Empties:200
Join INs:250
Leaves:10
Leave Alls:35
Empties:5
Fwd Alls:0
Fwd Unregistered:0
Total valid GMRP Packets Transmitted:600
Join Empties:200
Join INs:150
Leaves:45
Leave Alls:200
Empties:5
Fwd Alls:0
Fwd Unregistered:0
Total valid GMRP Packets Received:0
Total GMRP packets dropped:0
Total GMRP Registrations Failed:0
Console>
```

To display GMRP statistics on the switch, perform this task in privileged mode:

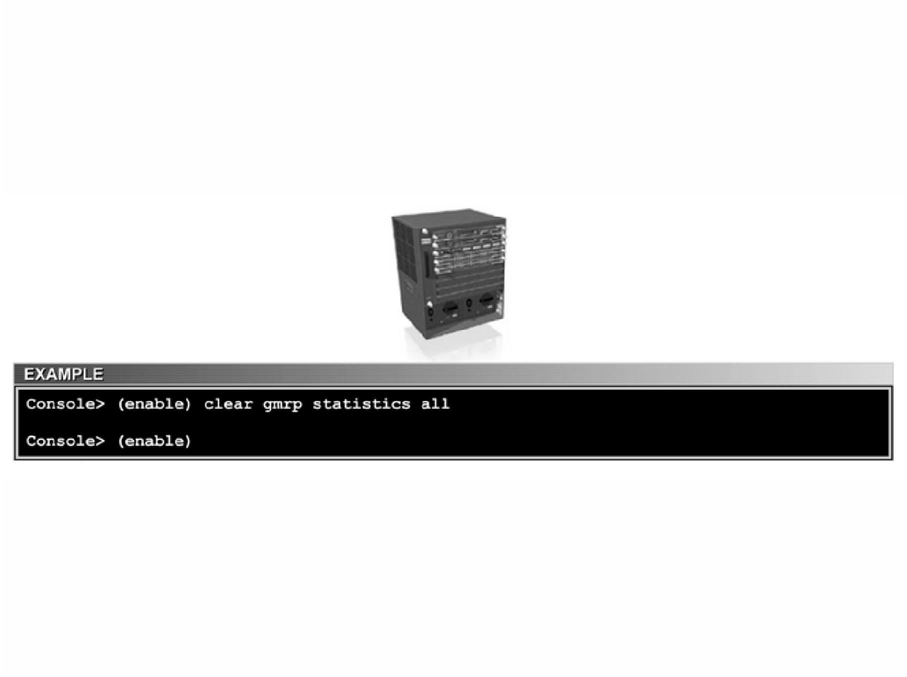
Table 3-36: show gmrp

Task	Command
Display GMRP statistics.	**show gmrp statistics** [*vlan_id*]

This example shows how to display GMRP statistics for VLAN 23:

```
Console> show gmrp statistics 23

GMRP Statistics for vlan <23>:
Total valid GMRP Packets Received:500
Join Empties:200
Join INs:250
Leaves:10
Leave Alls:35
Empties:5
Fwd Alls:0
Fwd Unregistered:0
Total valid GMRP Packets Transmitted:600
Join Empties:200
Join INs:150
Leaves:45
Leave Alls:200
Empties:5
Fwd Alls:0
Fwd Unregistered:0
Total valid GMRP Packets Received:0
Total GMRP packets dropped:0
Total GMRP Registrations Failed:0
Console>
```

Clearing GMRP Statistics



EXAMPLE

```
Console> (enable) clear gmrp statistics all

Console> (enable)
```

To clear all GMRP statistics on the switch, perform this task in privileged mode:

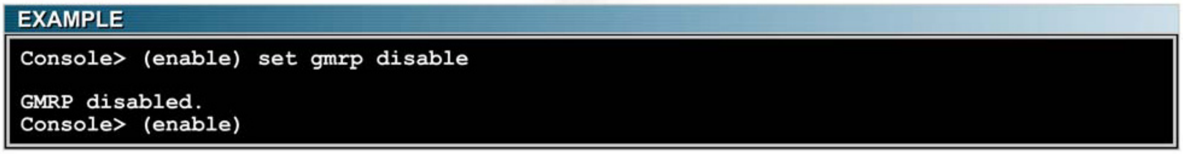
Table 3-37: clear gmrp

Task	Command
Clear GMRP statistics.	**clear gmrp statistics** {*vlan_id* | **all**}

This example shows how to clear the GMRP statistics for all VLANs:

```
Console> (enable) clear gmrp statistics all

Console> (enable)
```

Disabling GMRP Globally on the Switch



EXAMPLE

```
Console> (enable) set gmrp disable

GMRP disabled.
Console> (enable)
```

To disable GMRP globally on the switch, perform this task in privileged mode:

Table 3-38: Disable GMRP Globally

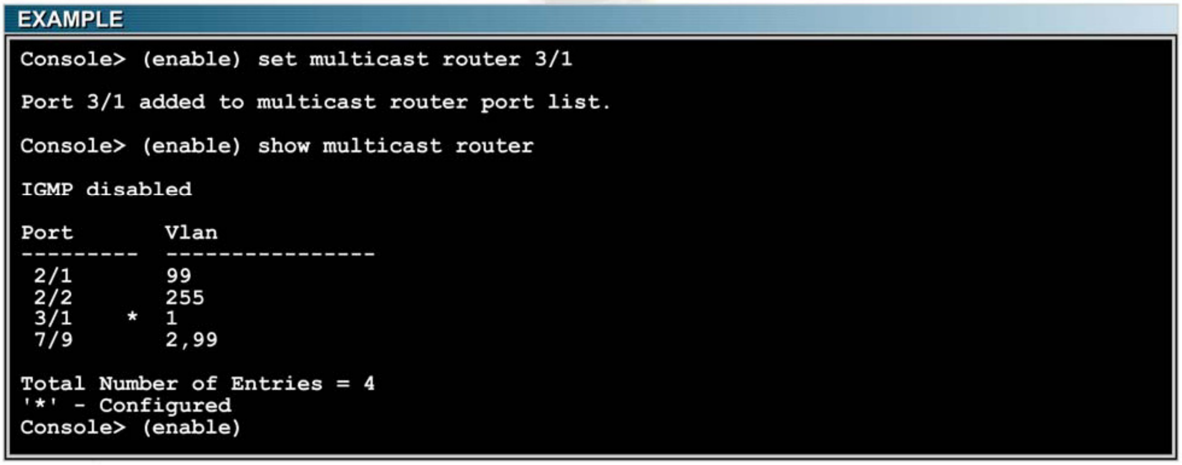
Task	Command
Disable GMRP globally on the switch.	**`set gmrp disable`**

This example shows how to disable GMRP globally on the switch:

```
Console> (enable) set gmrp disable

GMRP disabled.
Console> (enable)
```

Configuring Multicast Router Ports and Group Entries

In order for the Catalyst 6500 to effectively take advantage of IP Multicasting, you must tell the switch on which ports multicast routers reside.



EXAMPLE

```
Console> (enable) set multicast router 3/1

Port 3/1 added to multicast router port list.

Console> (enable) show multicast router

IGMP disabled

Port       Vlan
---------  ----------------
 2/1       99
 2/2       255
 3/1    *  1
 7/9       2,99

Total Number of Entries = 4
'*' - Configured
Console> (enable)
```

Specifying Multicast Router Ports

When you enable IGMP snooping, the switch automatically learns to which ports a multicast router is connected. However, if desired, you can manually specify multicast router ports.

To specify multicast router ports manually, perform this task in privileged mode:

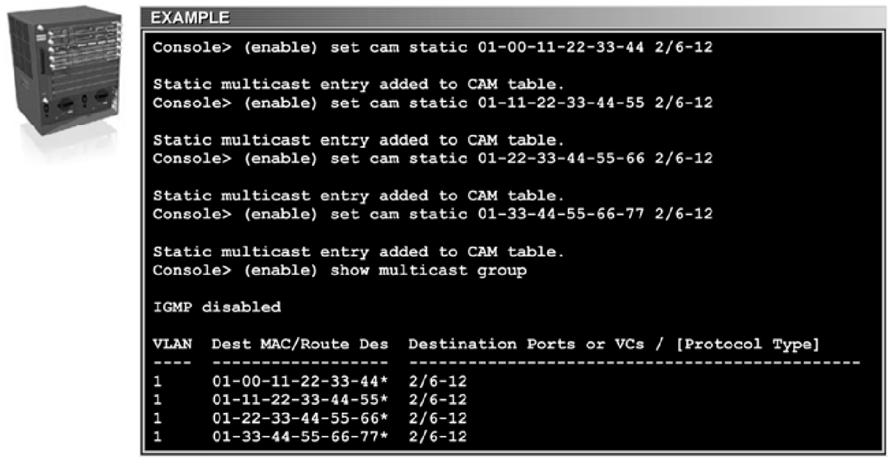
Table 3-39: set multicast router

	Task	Command
Step 1	Manually specify a multicast router port.	**`set multicast router`** *`mod/port`*
Step 2	Verify the configuration.	**`show multicast router`** [*`mod/port`*] [*`vlan_id`*]

This example shows how to specify a multicast router port manually and verify the configuration (the asterisk [*] next to the multicast router on port 3/1 indicates that the entry was configured manually):

```
Console> (enable) set multicast router 3/1

Port 3/1 added to multicast router port list.

Console> (enable) show multicast router

IGMP disabled

Port       Vlan
---------  ----------------
 2/1       99
 2/2       255
 3/1    *  1
 7/9       2,99

Total Number of Entries = 4
'*' - Configured
Console> (enable)
```

Configuring Multicast Groups



```
Console> (enable) set cam static 01-00-11-22-33-44 2/6-12

Static multicast entry added to CAM table.
Console> (enable) set cam static 01-11-22-33-44-55 2/6-12

Static multicast entry added to CAM table.
Console> (enable) set cam static 01-22-33-44-55-66 2/6-12

Static multicast entry added to CAM table.
Console> (enable) set cam static 01-33-44-55-66-77 2/6-12

Static multicast entry added to CAM table.
Console> (enable) show multicast group

IGMP disabled

VLAN  Dest MAC/Route Des  Destination Ports or VCs / [Protocol Type]
----  ------------------  --------------------------------------------
1     01-00-11-22-33-44*  2/6-12
1     01-11-22-33-44-55*  2/6-12
1     01-22-33-44-55-66*  2/6-12
1     01-33-44-55-66-77*  2/6-12
```

To configure a multicast group manually, perform this task in privileged mode:

Note	With software releases 7.1(1) and later, the maximum number of Layer 2 multicast entries is 15488.

Table 3-40: Configuring Multicast Groups

	Task	Command
Step 1	Add one or more multicast MAC addresses to the CAM table.	**set cam {static | permanent}** *multicast_mac mod/port* [*vlan*]
Step 2	Verify the multicast group configuration.	**show multicast group** [*mac_addr*] [*vlan_id*]

This example shows how to configure multicast groups manually and verify the configuration (the asterisks indicate the entry was manually configured):

Console> (enable) **set cam static 01-00-11-22-33-44 2/6-12**

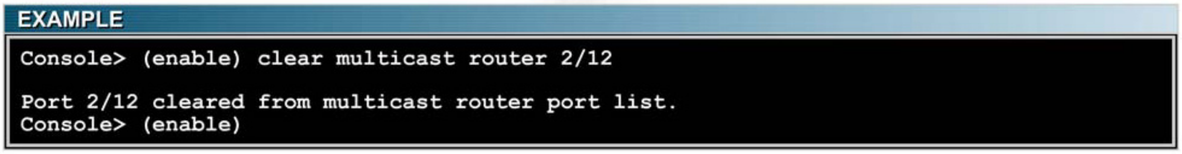
Static multicast entry added to CAM table.

Console> (enable) **set cam static 01-11-22-33-44-55 2/6-12**

Static multicast entry added to CAM table.

```
Console> (enable) set cam static 01-22-33-44-55-66 2/6-12

Static multicast entry added to CAM table.
Console> (enable) set cam static 01-33-44-55-66-77 2/6-12

Static multicast entry added to CAM table.
Console> (enable) show multicast group

IGMP disabled

VLAN  Dest MAC/Route Des  Destination Ports or VCs / [Protocol Type]
----  -----------------  ------------------------------------------------
1     01-00-11-22-33-44*  2/6-12
1     01-11-22-33-44-55*  2/6-12
1     01-22-33-44-55-66*  2/6-12
1     01-33-44-55-66-77*  2/6-12

Total Number of Entries = 4
Console> (enable)
```

Clearing Multicast Router Ports



EXAMPLE

```
Console> (enable) clear multicast router 2/12

Port 2/12 cleared from multicast router port list.
Console> (enable)
```

To clear manually configured multicast router ports, perform one of these tasks in privileged mode:

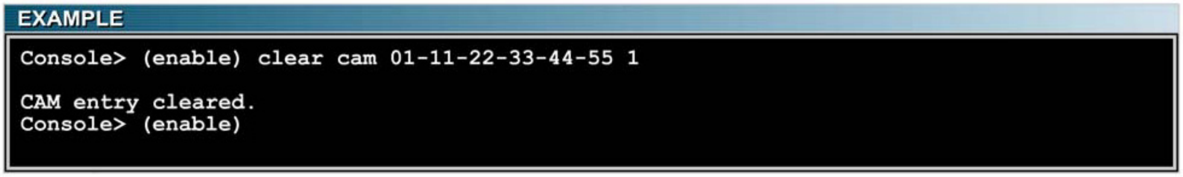
Table 3-41: clear multicast router

Task	Command
Clear specific, manually configured multicast router ports.	**`clear multicast router`** *`mod/port`*
Clear all manually configured multicast router ports.	**`clear multicast router all`**

This example shows how to clear a manually configured multicast router port entry:

```
Console> (enable) clear multicast router 2/12


Port 2/12 cleared from multicast router port list.
Console> (enable)
```

Clearing Multicast Group Entries



```
Console> (enable) clear cam 01-11-22-33-44-55 1

CAM entry cleared.
Console> (enable)
```

To clear manually configured multicast group entries, perform this task in privileged mode:

Table 3-42: clear cam

Task	Command
Clear a multicast group entry from the CAM table.	**`clear cam`** *`mac_addr`* `[`*`vlan`*`]`

This example shows how to clear a multicast group entry from the CAM table:

```
Console> (enable) clear cam 01-11-22-33-44-55 1

CAM entry cleared.
Console> (enable)
```

Displaying Multicast Protocol Status

EXAMPLE

```
Console> (enable) show multicast protocols status

IGMP disabled
IGMP fastleave enabled
RGMP enabled
GMRP disabled
```

This command displays the status (enabled or disabled) of the Layer 2 multicast protocols on the switch.

To display the multicast protocol status, perform this task in privileged mode:

Table 3-43: Displaying the Multicast Protocol Status

Task	Command
Display the multicast protocol status.	**`show multicast protocols status`**

This example shows how to display the multicast protocol status:

```
Console> (enable) show multicast protocols status

IGMP disabled
IGMP fastleave enabled
RGMP enabled
GMRP disabled
```

Lesson Summary

This lesson accomplished the following:

- Explained Multicast Routing
- Explained the differences between unicast, broadcast, and multicast routing
- Detailed the configuration of multicast routing at Layer 2 and Layer 3 on the Catalyst 6500

Next Steps

After completing this lesson, go to:

- Configuring WAN Services

References

For additional information, refer to these resources:

- IP Multicast: http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt3/index.htm

summary

- Detailed the configuration of InterVLAN routing on the Catalyst 6500 for the followin grouted protocols: IP, IPX, and AppleTalk
- Detailed the configuration of Multilayer Switching on the Catalyst 6500
- Explained the Layer 3 redundancy features available on the Catalyst 6500
- Explained the use of Hot Standby Routing Protocol (HSRP) on the MSFCs to provide default gateway redundancy to clients
- Detailed the configuration of high availability redundancy on the supervisor engines
- Explain the differences between unicast, broadcast, and multicast routing
- Detailed the configuration of multicast routing at Layer 2 and Layer 3 on the Catalyst 6500

Review Questions

Q1) Which of the following features is used to provide default gateway redundancy to clients?

A) InterVLAN routing

B) MLS

C) HSRP

D) High Availability

Answer: C

Q2) The Catalyst 6500 supports which of the following multicast protocols?

A) IGMP

B) CGMP

C) GMRP

D) DVMRP

Answer: A and C

Q3) What HSRP group does the following MAC address belong to?

0000.0c07.ac1b

Answer: 27

Q4) What command is used to enable IP MLS on an MSFC interface?

Answer: mls ip

Q5) Which of the following hardware configurations support Layer 3 redundancy?

A) A single chassis with two Supervisor 1 Engines, each with a PFC and MSFC1

B) A single chassis with two Supervisor 1 Engines, each with a PFC and MSFC2

C) A single chassis with two Supervisor 1 Engines, each with a PFC, and one with an MSFC1 and the other with an MSFC2

D) A single chassis with two Supervisor 1 Engines, each with a MSFC1 and only one with a PFC

E) Two chassis, with a Supervisor 2 Engine with a PFC2 and MSFC2 in each

Answer: A, B, D

WAN Services

Overview

This module introduces the concepts surrounding two of the Catalyst 6500's WAN interface modules, the FlexWAN Module and the ATM LANE/MPOA Module.

Outline

The module contains these lessons:

- FlexWAN Module
- ATM LANE/MPOA Module

FlexWAN Module

Overview

This lesson discusses the FlexWAN module. The FlexWAN module allows you to terminate WAN services such as T1/E1, T3/E3, and OC-3 on the Catalyst 6500 without the need for an external router. The same MSFC used for InterVLAN routing is used to configure these WAN interfaces.

Importance

This support of WAN services makes the Catalyst 6500 one of the most versatile switches on the market today.

Objectives

Upon completing this lesson, you will be able to:

- List the supported interfaces of the FlexWAN module
- Configure the FlexWAN module
- Verify the FlexWAN configuration

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

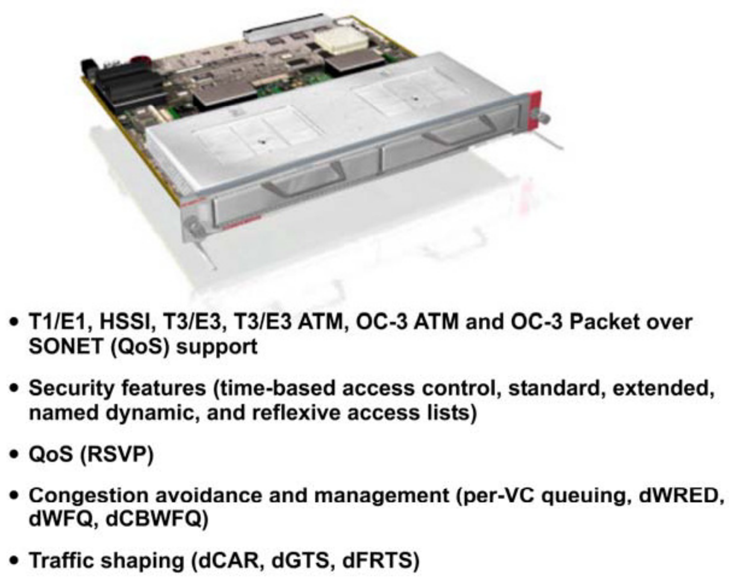
- Have a basic knowledge of WAN Technologies such as T1, Frame Relay, and ATM

Outline

This lesson includes these sections:

- Overview
- Features of the FlexWAN Module
- FlexWAN Software and Hardware
- Summary

Features of the FlexWAN Module



- **T1/E1, HSSI, T3/E3, T3/E3 ATM, OC-3 ATM and OC-3 Packet over SONET (QoS) support**
- **Security features (time-based access control, standard, extended, named dynamic, and reflexive access lists)**
- **QoS (RSVP)**
- **Congestion avoidance and management (per-VC queuing, dWRED, dWFQ, dCBWFQ)**
- **Traffic shaping (dCAR, dGTS, dFRTS)**

Originally positioned as a Distribution Layer switch, the Catalyst 6500 can connect to a variety of WAN interfaces, in addition to the selection of LAN interfaces discussed in a previous chapter. This WAN connectivity is made possible by the FlexWAN module. The FlexWAN module is installed in a Catalyst 6500 that has a Supervisor Engine with a Multilayer Switch Feature Card (MSFC) and Policy Feature Card (PFC). To leverage existing investments in WAN interfaces, the FlexWAN module can accept up to two Cisco 7200/7500 WAN port adapters. The FlexWAN module supports ATM and POS OC-3 links and channelized, multichannel, and clear channel port adapters at speeds from T1/E1 to T3/E3.

Note In a redundant MSFC configuration, the FlexWAN module interfaces appear only on the designated MSFC. To configure the FlexWAN module interfaces on the redundant MSFC, you must force a switchover. The interfaces can be configured and the configuration can be saved. The saved configuration is used on the redundant MSFC if a switchover occurs.

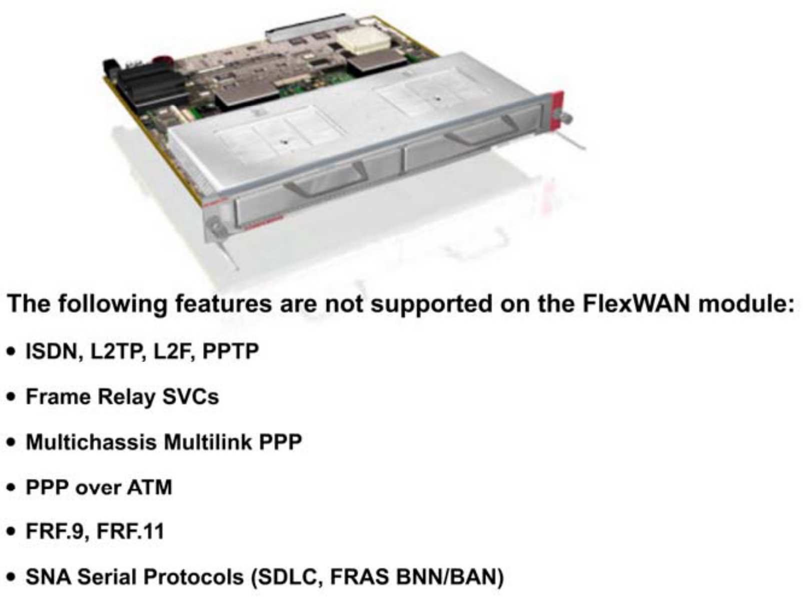
Supported Port Adapters

Table 4-1: Port Adapters

Group	Port Adapter
ATM (with traffic shaping)	PA-A3-OC3MM PA-A3-OC3SMI PA-A3-OC3SML PA-A3-T3 PA-A3-E3
T1/E1	PA-4T+ PA-8T-V35 PA-8T-X21 PA-8T-232 PA-MC-4T1 PA-MC-8T1 PA-MC-8E1/120 PA-MC-STM-1
T3/E3 (clear-channel and channelized)	PA-T3 PA-2T3 PA-T3+ PA-2T3+ PA-E3 PA-2E3 PA-MC-T3 PA-MC-2T3+ PA-MC-E3
HSSI	PA-H PA-2H
Packet over Sonet (OC-3)	PA-POS-OC3MM PA-POS-OC3SMI PA-POS-OC3SML

The Catalyst 6500 supports a wide range of 7200/7500 port adapters for connectivity to the WAN. The above table details the interface type and corresponding part numbers for these supported port adapters. Note that LAN port adapters (e.g., Ethernet and TokenRing) are not supported by the FlexWAN.

Unsupported Features



The following features are not supported on the FlexWAN module:

- **ISDN, L2TP, L2F, PPTP**
- **Frame Relay SVCs**
- **Multichassis Multilink PPP**
- **PPP over ATM**
- **FRF.9, FRF.11**
- **SNA Serial Protocols (SDLC, FRAS BNN/BAN)**

Some of the features typically supported by Cisco's IOS on WAN interfaces are not supported on the Catalyst 6500 FlexWAN. Specifically, the following features are unsupported:

- ISDN, L2TP, L2F, PPTP
- Frame Relay SVCs
- Multichassis Multilink PPP
- PPP over ATM
- FRF.9, FRF.11
- SNA Serial Protocols (SDLC, FRAS BNN/BAN)

FlexWAN Software and Hardware

Native IOS:

- **Supervisor Engine 2 and MSFC2**
  - IOS 12.1(8a)E
- **Supervisor Engine1 and MSFC1 or 2**
  - IOS 12.1(5a) E1

catOS:

- **Supervisor Engine – 5.4(2)**
- **MSFC – IOS 12.1(1)EX**

To support a FlexWAN module, the Catalyst 6500 requires an MSFC and a PFC. The configuration is performed from the MSFC's IOS prompt. Following are the specific software requirements.

- For Catalyst 6500s with Cisco IOS on both the Supervisor Engine 2 and MSFC2 - Cisco IOS Release 12.1(8a)E or later on both the Supervisor Engine and MSFC2

- For Catalyst 6500s with Cisco IOS on both the Supervisor Engine 1 and MSFC or MSFC2 - Cisco IOS Release 12.1(5a)E1 or later on both the Supervisor Engine and MSFC or MSFC2

- For Catalyst 6500s with Catalyst software on the Supervisor Engine and Cisco IOS software on the MSFC or MSFC2 - Cisco IOS Release 12.1(1)EX or later on the MSFC and 5.4(2) or later on the Supervisor Engine

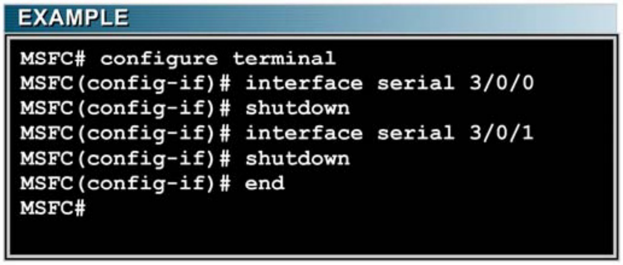
FlexWAN Module CLI Commands

Table 4-2: FlexWAN Commands

Command	Options	Description
MSFC# **show cwan**	**stats**	Displays CWAN statistics
	vlans	Displays hidden VLAN to WAN interface mapping
MSFC# **debug cwan**	**cmd-retry**	Displays FlexWAN module command retries
	cmd-timeout	Displays FlexWAN module command timeouts
	ifcom	Displays FlexWAN module interface communication
	interface	Displays FlexWAN module interface status
	love	Displays FlexWAN module love missives
	oir	Displays FlexWAN module OIR events
MSFC# **dir cwan**	**mod_num/bay-bootflash:**	Displays the FlexWAN module bootflash devices
MSFC# **show controller vip** [0-32]	**accumulator**	Displays VIP MEMD accumulators and Rx buffering statistics
	align	Displays recorded alignment data
	diagbus	Displays DBUS software interface information
	logging	Displays logging information
	proc	Displays active process statistics
	tech-support	Displays system information for tech-support

The above table provides a reference for verification and troubleshooting of the FlexWAN module. Since the FlexWAN interfaces act as routed interfaces of the MSFC, these commands are issued from the MSFC's privileged mode.

FlexWAN Interface Configuration Syntax



```
MSFC# configure terminal
MSFC(config-if)# interface serial 3/0/0
MSFC(config-if)# shutdown
MSFC(config-if)# interface serial 3/0/1
MSFC(config-if)# shutdown
MSFC(config-if)# end
MSFC#
```

After verifying that the port adapter is installed correctly (the enabled LED goes on), use the privileged-level `configure` command to configure the new interfaces. Be prepared with the following information:

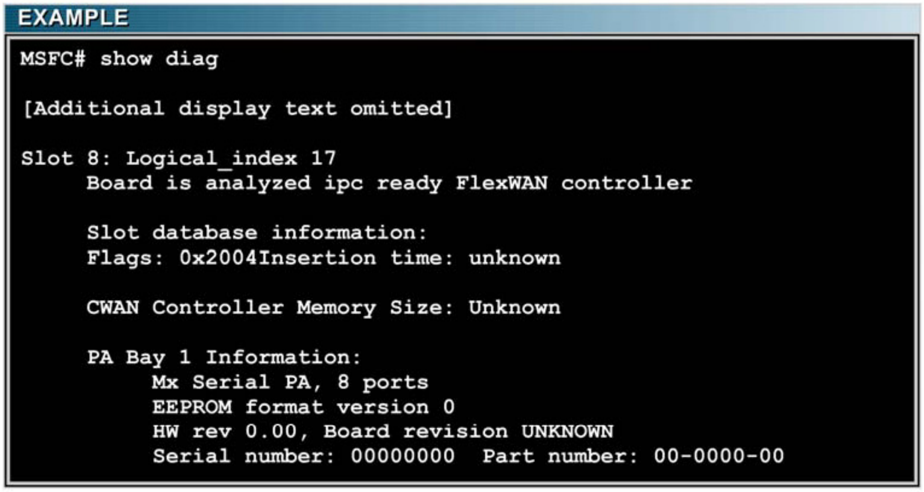
- Protocols to route on each new interface
- IP addresses for the interfaces configured for IP routing
- Whether or not the new interfaces will use bridging
- Timing source for each new interface and clock speeds for external timing

If you installed a new port adapter or want to change the configuration of an existing interface, you must enter configuration mode using the `configure` command. If you replaced a port adapter that was previously configured, the system will recognize the new port adapter interfaces and bring each of them up in their existing configuration.

Table 4-3: Interface Commands

Command	Description
interface, followed by the *type* (**serial**) and *mod_num/bay/port* (module-slot-number/port-adapter-bay-number/ interface-port-number)	The example is for interface 0 and interface 1 on a port adapter in port adapter bay 0 of a FlexWAN module installed in module slot 3. MSFC(config-if)# **interface serial 3/0/0** MSFC(config-if)# **shutdown** MSFC(config-if)# **interface serial 3/0/1** MSFC(config-if)# **shutdown** MSFC(config-if)# **end** MSFC#

Viewing FlexWAN Diagnostics



EXAMPLE

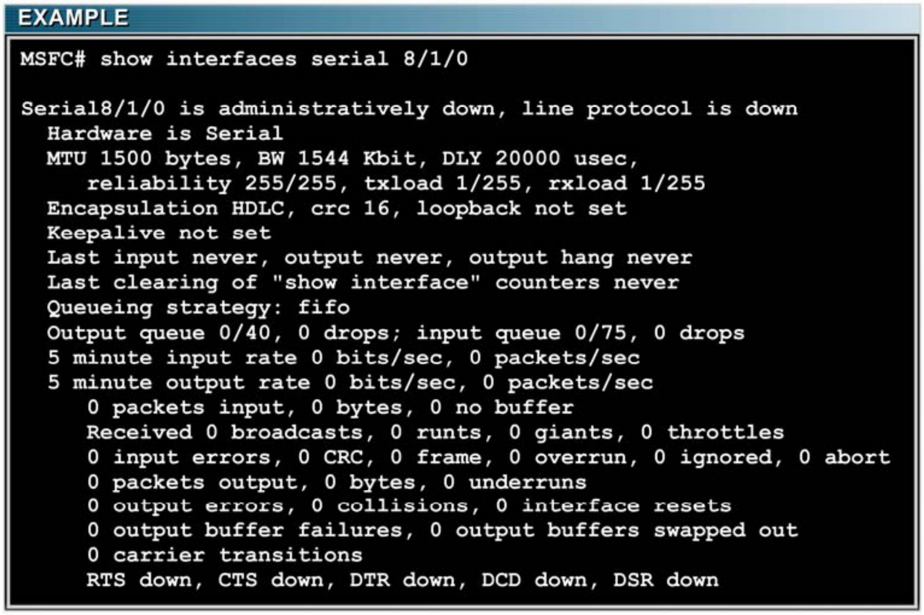
```
MSFC# show diag

[Additional display text omitted]

Slot 8: Logical_index 17
     Board is analyzed ipc ready FlexWAN controller

     Slot database information:
     Flags: 0x2004Insertion time: unknown

     CWAN Controller Memory Size: Unknown

     PA Bay 1 Information:
          Mx Serial PA, 8 ports
          EEPROM format version 0
          HW rev 0.00, Board revision UNKNOWN
          Serial number: 00000000  Part number: 00-0000-00
```

The **`show diag`** command can be used to display the types of port adapters installed in the system (and specific information about each).

Following is an example of the **`show diag`** command that shows a PA-4T+ on a Catalyst 6500 FlexWAN module:

```
MSFC# show diag

[Additional display text omitted]

Slot 8: Logical_index 17
     Board is analyzed ipc ready FlexWAN controller

     Slot database information:
     Flags: 0x2004      Insertion time: unknown

     CWAN Controller Memory Size: Unknown

     PA Bay 1 Information:
          Mx Serial PA, 8 ports
          EEPROM format version 0
          HW rev 0.00, Board revision UNKNOWN
          Serial number: 00000000  Part number: 00-0000-00
```

Viewing the FlexWAN Interfaces



```
MSFC# show interfaces serial 8/1/0

Serial8/1/0 is administratively down, line protocol is down
  Hardware is Serial
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive not set
  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 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     RTS down, CTS down, DTR down, DCD down, DSR down
```

Just as with traditional Cisco IOS routers, the **show interfaces** command can be used to view the characteristics of a port installed in a FlexWAN module. Following is an example of the **show interfaces serial** command, which shows all of the information specific to interface port 0 on a PA-4T+ installed in port adapter slot 8 (interfaces are administratively shut down until you enable them):

```
MSFC# show interfaces serial 8/1/0


Serial8/1/0 is administratively down, line protocol is down
  Hardware is Serial
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive not set
  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 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
```

```
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
RTS down, CTS down, DTR down, DCD down, DSR down
```

Lesson Summary

This lesson accomplished the following:

- List the supported interfaces of the FlexWAN module
- Configure the FlexWAN module
- Verify the FlexWAN configuration

ATM LANE/MPOA Module

Overview

The ATM LANE/MPOA module allows the Catalyst 6500 to connect to an ATM Backbone.

Importance

Support for both ATM Forum standard LAN Emulation (LANE) and Multiprotocol over ATM (MPOA) enable the Catalyst 6500 Family of switches to connect to ATM backbones, providing scalable, intelligent switching with value-added High Availability services.

Objectives

Upon completing this lesson, you will be able to:

- Describe the ATM LANE/MPOA module
- Discuss the characteristics of ATM
- Distinguish between the functions of LANE and MPOA

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Have a basic knowledge of ATM. This includes familiarity with the following terminology: UNI, NNI, PVC, and SVC.

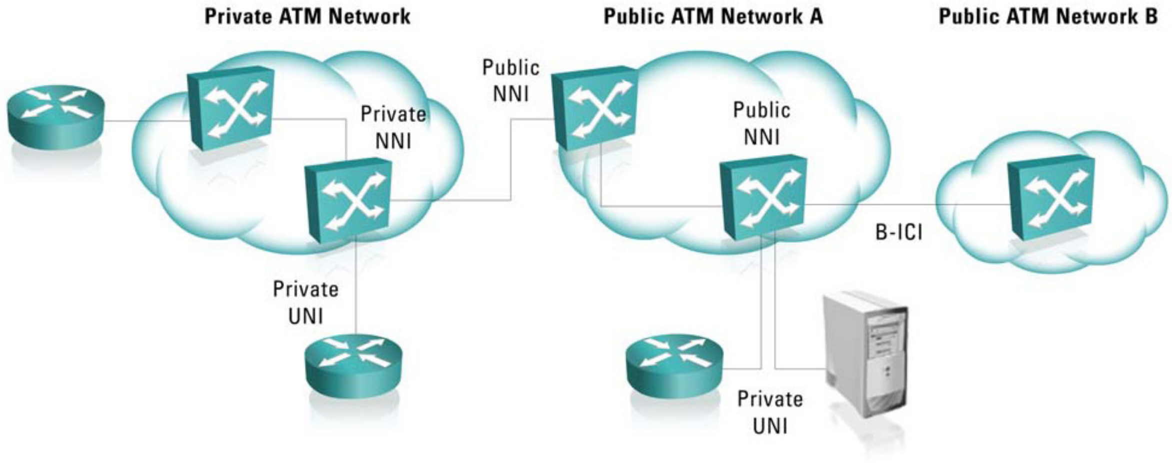
Outline

This lesson includes these sections:

- Overview
- ATM Overview
- Summary

ATM Overview

- UNI—User-to-Network Interface
- NNI—User-to-User Interface
- B-ICI Devices—Broadband ISDN Inter-Carrier Interface



The Catalyst 6500 can attach to ATM networks via the ATM LANE/MPOA module, using either singlemode or multimode fiber. This lesson provides an introduction to ATM technology and distinguishes between the features of LANE and MPOA.

Structure of an ATM Network

ATM is based on the concept of two end-point devices communicating by means of intermediate switches. As the above figure shows, an ATM network is made up of a series of switches and end-point devices. The end-point devices can be ATM-attached end stations, ATM-attached servers, or ATM-attached routers.

Components of an ATM Network

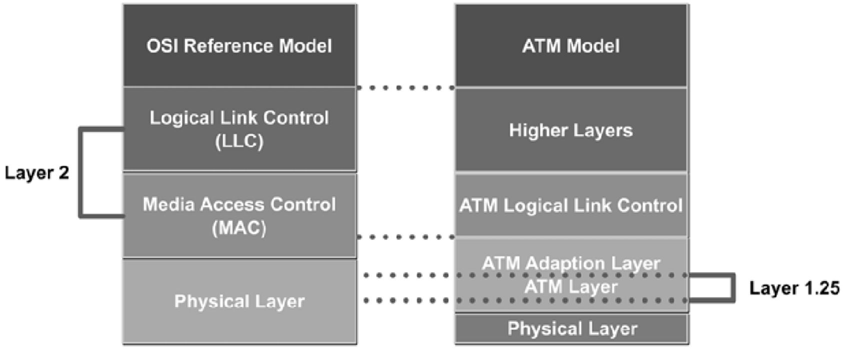
As seen in the above figure, there are two types of interfaces in an ATM network:

- User-to-Network Interface (UNI)
- Network-to-Network Interface (NNI)

The UNI connection is made up of an end-point device and a private or public ATM switch. The NNI is the connection between two ATM switches. The UNI and NNI connections can be carried by different physical connections.

In addition to the UNI and NNI protocols, the ATM Forum has defined a set of LAN Emulation (LANE) standards and a Private Network to Network Interface (PNNI) Phase 0 protocol. LANE is a technology network designers can use to internetwork legacy LANs such as Ethernet and Token Ring with ATM-attached devices. Most LANE networks consist of multiple ATM switches and typically employ the PNNI protocol.

ATM Functional Layers

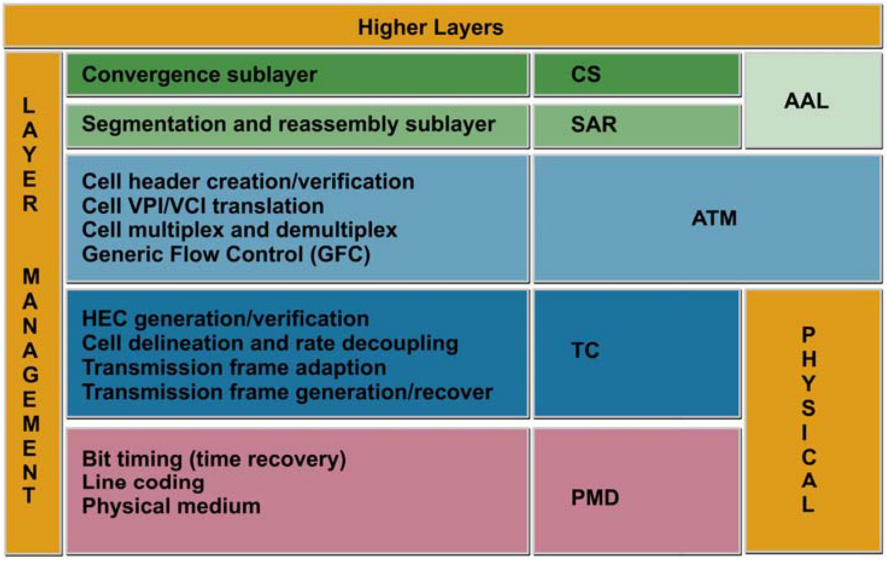


Similar to the *Open System Interconnection* (OSI) reference model, which describes how two computers communicate over a network, the ATM protocol model describes how two end systems communicate through ATM switches. The ATM protocol model consists of the following three functional layers:

- ATM physical layer
- ATM layer
- ATM adaptation layer

As the above figure shows, these three layers roughly correspond to Layer 1 and parts of Layer 2 (such as error control and data framing) of the OSI reference model.

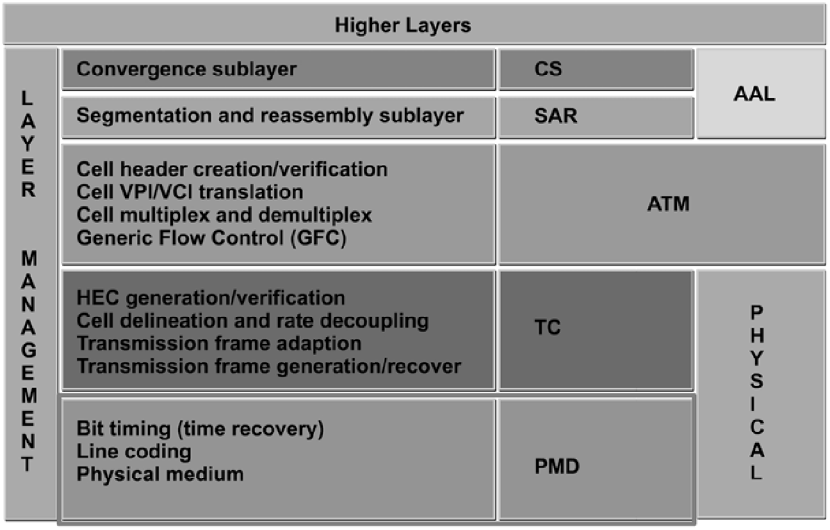
Physical Layer


Higher Layers
LAYER MANAGEMENT
Convergence sublayer
CS
Segmentation and reassembly sublayer
SAR
AAL
Cell header creation/verification
Cell VPI/VCI translation
Cell multiplex and demultiplex
Generic Flow Control (GFC)
ATM
HEC generation/verification
Cell delineation and rate decoupling
Transmission frame adaption
Transmission frame generation/recover
TC
Bit timing (time recovery)
Line coding
Physical medium
PMD
PHYSICAL

The ATM physical layer controls transmission and receipt of bits on the physical medium. It also keeps track of ATM cell boundaries and packages cells into the appropriate type of frame for the physical medium being used. The ITU model breaks the ATM physical layer into two parts:

- Physical medium sublayer
- Transmission convergence sublayer

Physical Medium Sublayer



Higher Layers

LAYER MANAGEMENT

Convergence sublayer | CS | AAL

Segmentation and reassembly sublayer | SAR

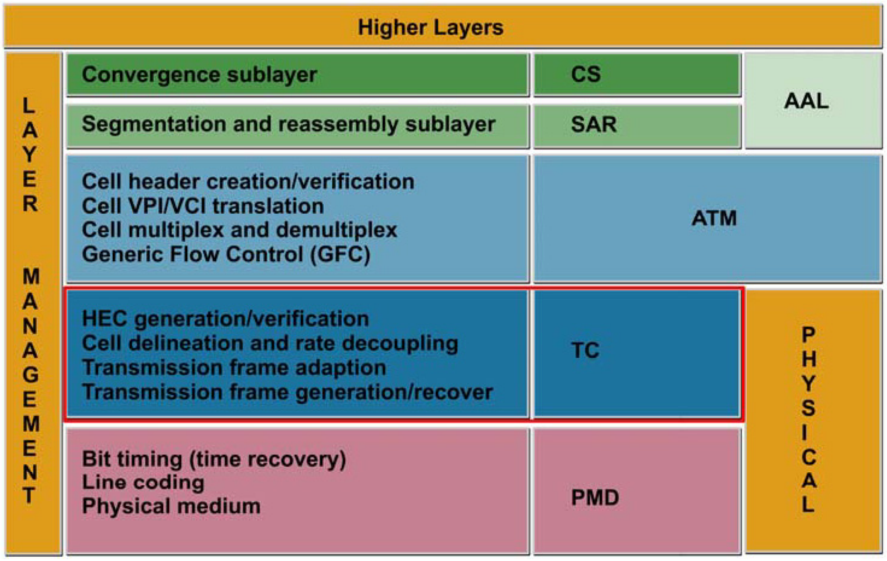
Cell header creation/verification
Cell VPI/VCI translation
Cell multiplex and demultiplex
Generic Flow Control (GFC) | ATM

HEC generation/verification
Cell delineation and rate decoupling
Transmission frame adaption
Transmission frame generation/recover | TC | PHYSICAL

Bit timing (time recovery)
Line coding
Physical medium | PMD

The type of physical transmission medium being used determines such line characteristics as line coding. The physical medium sublayer is responsible for sending and receiving a continuous flow of bits with associated timing information to synchronize transmission and reception. Because it includes only physical-medium-dependent functions, its specification depends on the physical medium used. ATM can use any physical medium capable of carrying ATM cells. Some existing standards that can carry ATM cells are SONET (Synchronous Optical Network)/SDH, DS-3/E3, 100-Mbps local fiber (Fiber Distributed Data Interface [FDDI] physical layer), and 155-Mbps local fiber (Fiber Channel physical layer). Various proposals for use over twisted-pair wire are also under consideration.

Transmission Convergence Sublayer


Higher Layers
LAYER MANAGEMENT
Convergence sublayer
CS
Segmentation and reassembly sublayer
SAR
AAL
Cell header creation/verification
Cell VPI/VCI translation
Cell multiplex and demultiplex
Generic Flow Control (GFC)
ATM
HEC generation/verification
Cell delineation and rate decoupling
Transmission frame adaption
Transmission frame generation/recover
TC
PHYSICAL
Bit timing (time recovery)
Line coding
Physical medium
PMD

The transmission convergence sublayer takes ATM cells and puts them into a bit stream for transmission. Specifically, the transmission convergence sublayer is responsible for the following:

- *Cell delineation*—Maintains ATM cell boundaries.
- *Header error control sequence generation and verification*—Generates and checks the header error control code to ensure valid data.
- *Cell rate decoupling*—Inserts or suppresses idle (unassigned) ATM cells to adapt the rate of valid ATM cells to the payload capacity of the transmission system.
- *Transmission frame adaptation*—Packages ATM cells into frames acceptable to the particular physical-layer implementation.
- *Transmission frame generation and recovery*—Generates and maintains the appropriate physical-layer frame structure.

ATM Modules

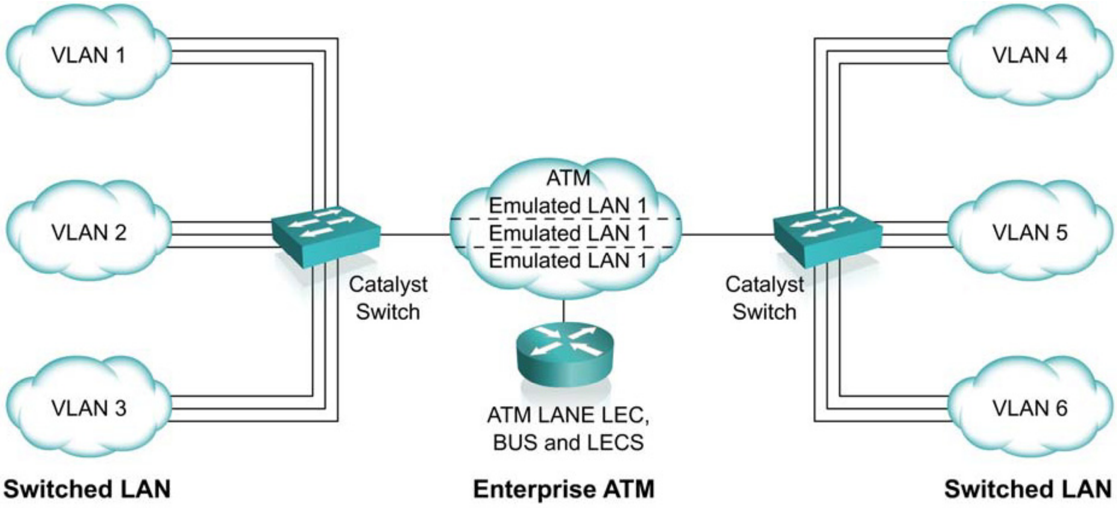
Table 4-4: ATM Module Features

Product Number	Module Description
WS-X6101-SMF	ATM Dual PHY OC-12 LANE/MPOA Module (SMF) ■ Reassembly of up to 255 buffers simultaneously; each buffer represents a packet ■ Support for up to 4096 virtual circuits ■ Support for AAL 5 ■ ATM LANE 1.0, including LEC[1], LES[2], BUS[3], and LECS[4] ■ MPOA support[5]
WS-X6101-MMF	ATM Dual PHY OC-12 LANE/MPOA Module (MMF) ■ Reassembly of up to 255 buffers simultaneously; each buffer represents a packet ■ Support for up to 4096 virtual circuits ■ Support for AAL 5 ■ ATM LANE 1.0, including LEC, LES, BUS, and LECS ■ MPOA support

[1] LEC=LAN Emulation Client
[2] LES=LAN Emulation Server
[3] BUS=broadcast and unknown server
[4] LECS=LAN Emulation Configuration Server
[5] MPOA=Multiprotocol over ATM

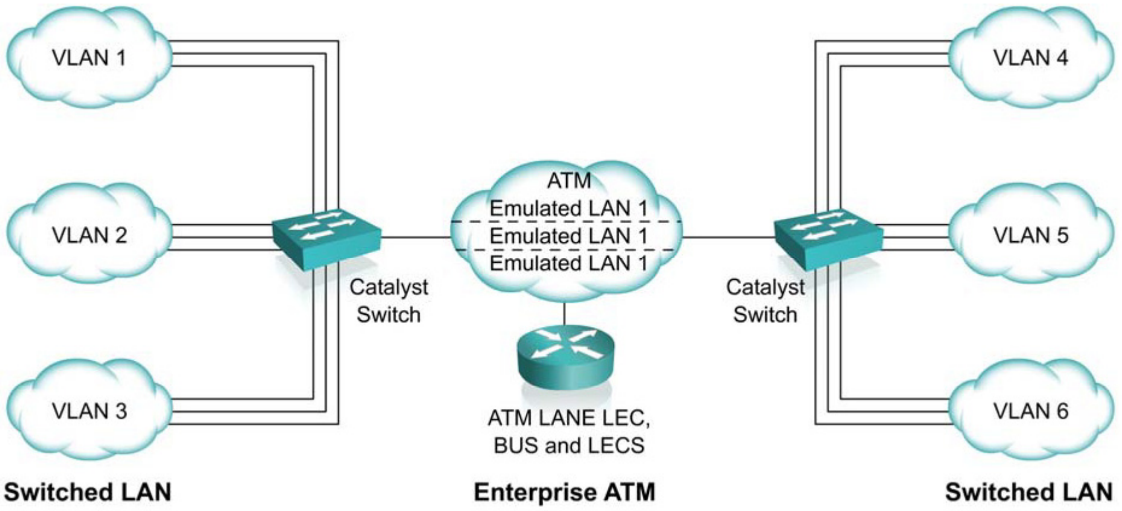
The two Catalyst 6500 ATM LANE/MPOA modules detailed in the above table both operate at OC-12 (622 Mbps) speeds. Both LANE 1.0 and MPOA functions are supported. The primary distinction between the two modules is the type of fiber supported. The WS-X6101-SFM supports singlemode fiber, and the WS-X6101-MMF support multimode fiber. An SC fiber connector is used on these modules.

ATM LANE Overview



While ATM is a non-broadcast multi-access (NBMA) technology, ATM LANE can be used to emulate a LAN (i.e., broadcast) environment, via emulated LANs (ELANs). But unlike a physical LAN server/host architecture, emulated LANs are logical. LANE is used to provide connectivity among hubs, bridges, routers, and switches to an ATM high-speed backbone. In addition, LANE allows end stations to communicate with an ATM-attached device, such as a file server, through a LAN-to-ATM switch without requiring the traffic to pass through a router. The LANE features supported in the Catalyst 6500's ATM LANE/MPOA module are LANE version 1 features. LANE version 2, which is not supported, adds redundancy and resiliency to a LANE network.

LANE Requirements



- **Connectivity between ATM-attached stations and LAN-attached stations**
- **Connectivity between LAN-attached stations across an ATM network**

The Catalyst 6500 ATM LANE/MPOA module requires connectivity to an ATM switch that supports User-Network Interface (UNI) 3.0 or 3.1 and point-to-multipoint signaling. For example, a Cisco's LightStream ATM switch could be used to connect to the Catalyst 6500.

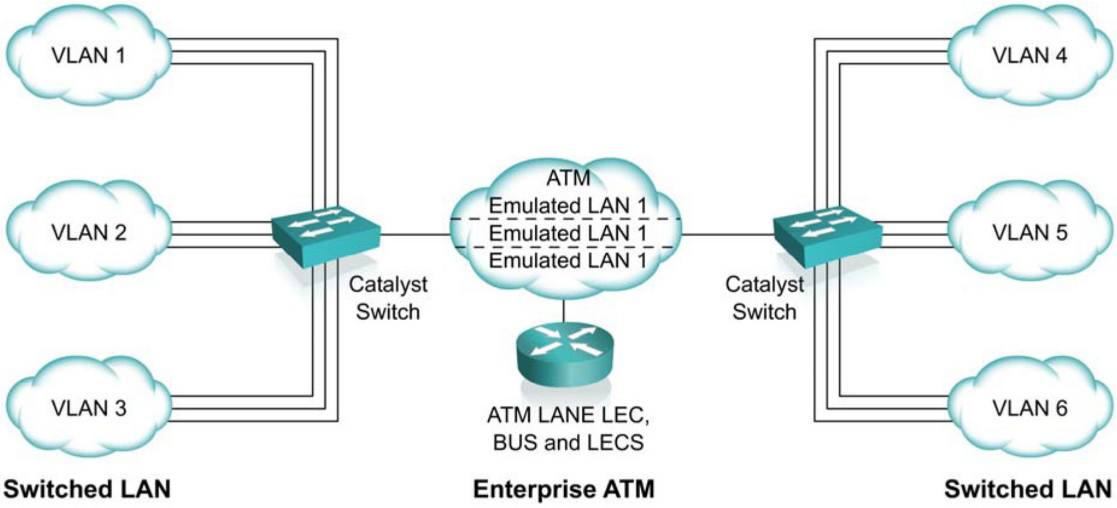
ATM LANE provides the following features:

- Connectivity between ATM-attached stations and LAN-attached stations
- Connectivity between LAN-attached stations across an ATM network

Because LANE connectivity is defined at the Media Access Control (MAC) layer, upper-layer protocol functions of LAN applications can operate unchanged when the devices join ELANs. This feature protects corporate investments in legacy LAN applications.

An ATM network can support multiple independent ELANs. End-system membership in any of the ELANs is independent of the physical location of the end system, simplifying hardware adds, moves, and changes. In addition, the end systems can move easily from one ELAN to another, whether or not the hardware moves. End stations in switched LANs are interconnected through an ATM network with ELANs that have been mapped to existing VLANs on the switched LANs. VLANs can be extended across the ATM backbone by mapping them to configured ELANs.

Extending VLANs Using ATM LANE



- **LANE Client (LEC): emulates a LAN interface to higher-layer protocols and applications**
- **LANE Server (LES): the control center for an ELAN**
- **LANE broadcast-and-unknown server (BUS): sequences and distributes multicast and broadcast packets**
- **LANE Configuration Server (LECS): contains a database of ATM addresses of LES/BUS pairs**

An unlimited number of ELANs can be created in an ATM cloud. A Catalyst 6500 ATM module can host multiple ELANs. With the Fast Simple Server Redundancy Protocol (FSSRP), each ELAN can contain multiple LES/BUS pairs for fault tolerance.

LANE is defined on a client-server LAN model and consists of these components:

- **LANE Client (LEC)**—A LEC emulates a LAN interface to higher-layer protocols and applications. LECs forward data to other LANE components and perform LANE address-resolution functions. Each LEC is a member of only one ELAN. Traffic must be routed between LECs that belong to different ELANs.

- **LANE Server (LES)**—The LES for an ELAN is the control center. It provides joining, address resolution, and address registration services to the LECs in that ELAN. LECs can register destination unicast and multicast MAC addresses with the LES. In addition, the LES handles LANE Address Resolution Protocol (LE ARP) requests and responses.

- **LANE broadcast-and-unknown server (BUS)**—The LANE BUS sequences and distributes multicast and broadcast packets and handles unicast flooding. At least one combined LES and BUS is required per ELAN.

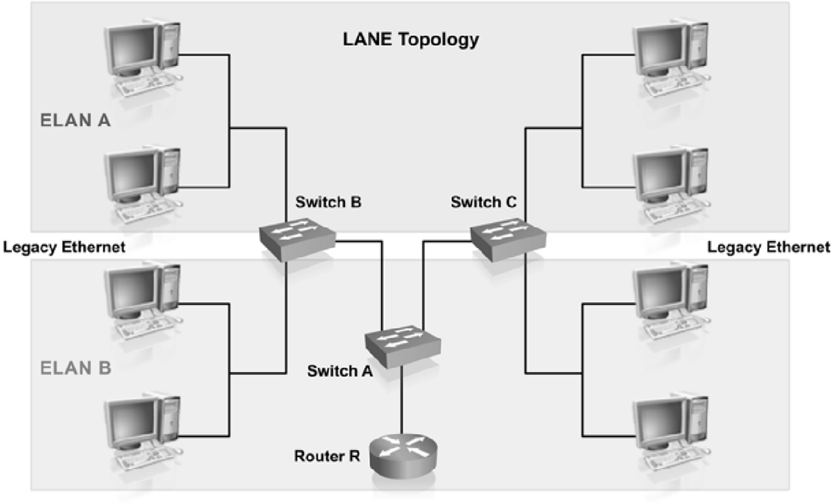
- **LANE Configuration Server (LECS)**—The LECS contains a database of ATM addresses of LES/BUS pairs for configured ELANs. A LEC consults the LECS to determine the LES's ATM address when it first joins an ELAN. The LECS returns the ATM address of the LES for that ELAN.

At least one LECS is required per ATM LANE switch cloud.

The LECS database can have the following four types of entries:

- ELAN_name, ATM_address_of_LES/BUS pairs
- LEC_MAC_address, ELAN_name pairs
- LEC_ATM_template, ELAN_name pairs
- Default_ELAN_name

Comparing VLANs and ELANs

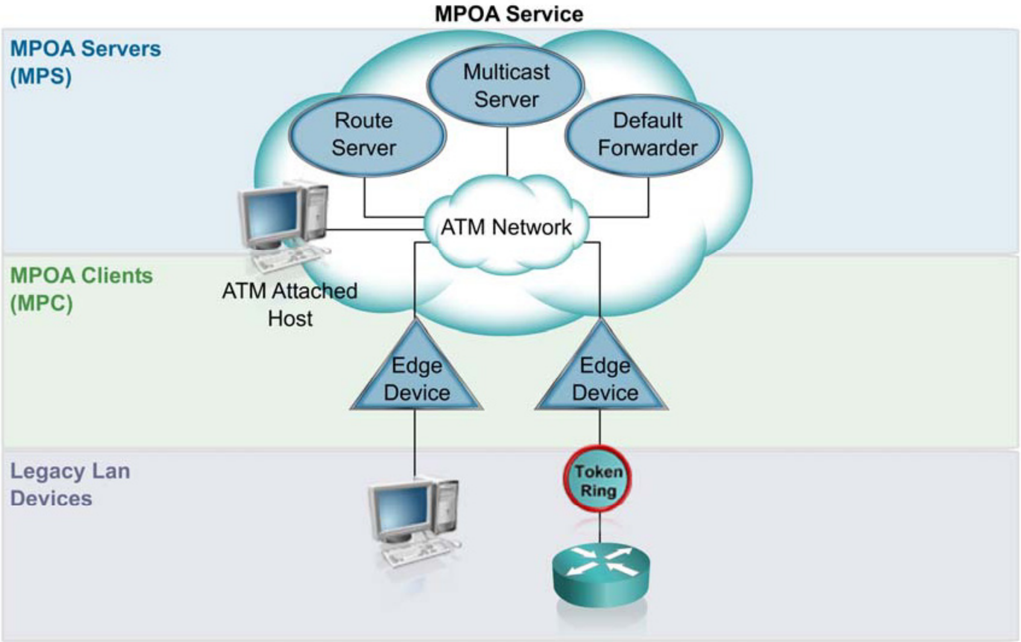


Catalyst 6500 switches support port-based VLAN configuration. An end station connected to a port belongs to the VLAN assigned to that port. The VLAN number identifies the VLAN.

On an ATM network, ELANs are designated by a name. Some ELANs can be configured from a router and some from a Catalyst 6500 switch. Additionally, some ELANs can be configured with unrestricted membership and some with restricted membership. A default ELAN, which must have unrestricted membership, can also be configured.

To create a VLAN that spans multiple Catalyst LAN switches across an ATM network, a given ATM ELAN must be mapped to the same VLAN configured on each switch. For example, if you have VLAN 10 configured on two different switches, you must map VLAN 10 to the same ATM ELAN. To communicate between two or more ELANs, you must use a router, whether the ELANs are on the same or different Catalyst switches.

MPOA Overview



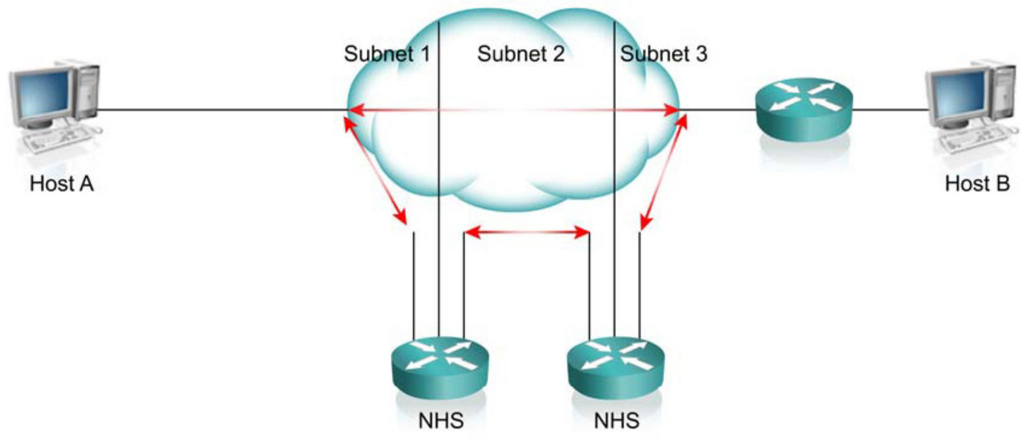
MPOA (MultiProtocol Over ATM) enhances the features provided by LANE. While a LANE environment requires a router to pass traffic between subnets, MPOA enables the routing of packets across an ATM network. MPOA replaces multihop routing with point-to-point routing using a direct virtual channel connection (VCC) between ingress and egress edge devices or hosts. An ingress edge device or host is the point at which an inbound flow enters the MPOA system; an egress edge device or host is the point at which an outbound flow exits the MPOA system.

Following are the required components for using MPOA across an NBMA network:

- MPOA Client (MPC)
- MPOA Server (MPS)
- Catalyst 6000 family ATM module
- LAN Emulation (LANE)
- Next Hop Resolution Protocol (NHRP)

MPOA Components

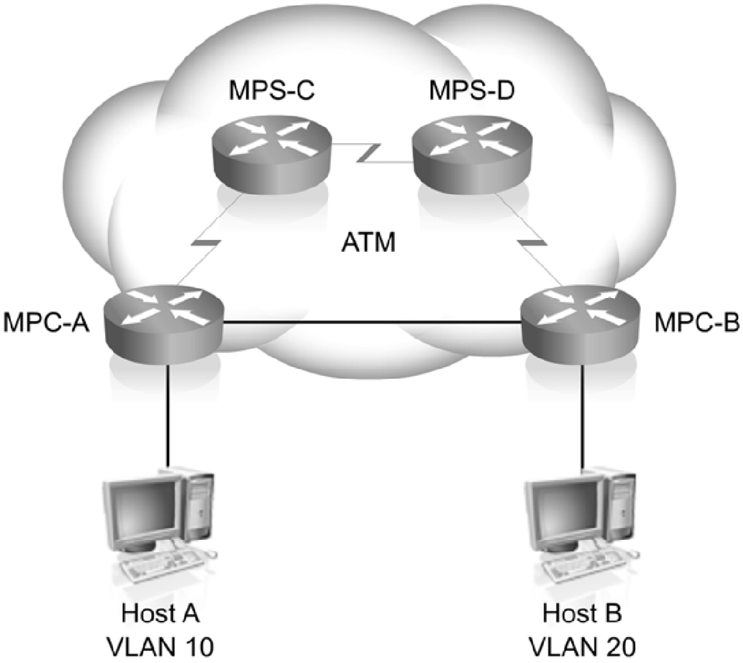
MPOA Service



To reduce the hop count for traffic between subnets over an ATM network, NHRP divides the ATM network into logical IP subnets. Although routers are still required to connect these subnets, NHRP allows intermediate routers to be bypassed by providing an extended address resolution protocol that permits Next Hop Clients (NHCs) to send queries directly between subnets. By integrating LANE and NHRP, MPOA extends the benefits of LANE by allowing intrasubnet communication over ATM VCCs without requiring routers in the data path.

Using NHRP's extended address resolution protocol, MPOA increases performance and reduces latencies by identifying the edge devices, establishing a direct VCC between the ingress and egress edge devices, and forwarding Layer-3 packets directly over this shortcut VCC, which bypasses the intermediate routers. An MPC provides the direct VCCs between the edge devices or hosts whenever possible and forwards Layer-3 packets over these shortcut VCCs. To establish shortcuts, MPCs communicate with MPSs resident on routers. The MPSs interact with their local Next Hop Servers (NHSs), which form part of the MPSs, to initiate and answer resolution requests. When an MPS receives updates from its NHS, it updates or purges relevant MPC caches as appropriate.

How MPOA Works



In an NBMA network, routing between subnets involves forwarding packets hop-by-hop through intermediate routers. MPOA can increase performance and reduce latencies by identifying the edge devices, establishing a direct VCC between the ingress and egress edge devices, and forwarding Layer-3 packets directly over this shortcut VCC, bypassing the intermediate routers.

An MPOA client (MPC) provides the direct VCCs between the edge devices or hosts whenever possible and forwards Layer-3 packets over these shortcut VCCs. The MPCs must be used with MPSs resident on routers.

The sequence of events shown in the figure is summarized as follows:

1. MPOA resolution request sent from MPC-A to MPS-C
2. NHRP resolution request sent from MPS-C to MPS-D
3. MPOA cache-imposition request sent from MPS-D to MPC-B
4. MPOA cache-imposition reply sent from MPC-B to MPS-D
5. NHRP resolution reply sent from MPS-D to MPS-C
6. MPOA resolution reply sent from MPS-C to MPC-A
7. Shortcut VCC established

The following table defines common MPOA terms:

Table 4-5: MPOA Terms

MPOA Term	Definition
MPOA resolution request	A request from an MPC to resolve a destination protocol address to an ATM address to establish a shortcut VCC to the egress device
NHRP resolution request	An MPOA resolution request, which has been converted to an NHRP resolution request
MPOA cache-imposition request	A request from an egress MPS to an egress MPC providing the MAC rewrite information for a destination protocol address
MPOA cache-imposition reply	A reply from an egress MPC acknowledging an MPOA cache-imposition request
NHRP resolution reply	An NHRP resolution reply that eventually will be converted to an MPOA resolution reply
MPOA resolution reply	A reply from the ingress MPS resolving a protocol address to an ATM address
Shortcut VCC	The path between MPCs over which Layer-3 packets are sent

Lesson Summary

This lesson accomplished the following:

- Described the ATM LANE/MPOA module
- Discussed the characteristics of ATM
- Distinguished between the functions of LANE and MPOA

summary

- Listed the supported interfaces of the FlexWAN module
- Demonstrated the configuration of the FlexWAN module
- Verified the FlexWAN configuration
- Described the ATM LANE/MPOA module
- Discussed the characteristics of ATM
- Distinguished between the functions of LANE and MPOA

Review Questions

Q1) Which of the following port adapters is NOT supported in a FlexWAN module?

A) T1/E1

B) HSSI

C) POS

D) TokenRing

Answer: D

Q2) Which of the following is true regarding the FlexWAN module?

A) The FlexWAN is only supported in Hybrid mode

B) The FlexWAN can support any 7200/7500 port adapter

C) The FlexWAN is configured via the Cat OS

D) The FlexWAN requires an MSFC and a PFC on the Supervisor Engine

Answer: D

Q3) Which of the following interfaces connects and end-user device to an ATM network?

A) UNI

B) NNI

C) PNNI

D) ILMI

Answer: A

Q4) What speed is supported on the Catalyst 6500's ATM LANE/MPOA module?

A) 45 Mbps

B) 155 Mbps

C) 622 Mbps

D) 25 Mbps

Answer: C

Q5) Which service supports cut-through routing between subnets in an ATM network?

A) LANE

B) MPOA

C) FUNI

D) LECS

Answer: B

Voice Services

Overview

This module details the voice services features available on the Catalyst 6500 Series switch, including configuration commands.

Upon completing this module, you will be able to:

- Configure Inline Power support for Cisco IP phones
- Configure Auxiliary VLANs on the Catalyst 6500 Series switch
- Describe the WS-X6624-FXS Module
- Describe the WS-X6608-T1/E1 Modules
- Configure and verify voice ports on the WS-X6624-FXS and WS-X6608-T1/E1 modules

Outline

The module contains these lessons:

- Inline Power and Auxiliary VLANs
- Overview of the WS-X6624-FXS and WS-X6608-T1/E1 Modules
- Voice Port Configuration and Verification

Inline Power and Auxiliary VLANs

Overview

A Cisco IP phone requires power to operate. The Catalyst 6500 supports a feature that provides power over the existing Ethernet cable. Additionally, the voice traffic can be separated from and prioritized over data traffic using Auxiliary VLANs. This lesson details the configuration of Inline Power and Auxiliary VLANs.

Importance

The Inline Power and Auxiliary VLAN features discussed in this lesson can optimize voice performance, while providing more reliable power to Cisco IP phones.

Objectives

Upon completing this lesson, you will be able to:

- List the three methods of powering an IP phone
- Describe how a Cisco IP phone can keep voice and data traffic separate
- Configure the Inline Power feature
- Configure Auxiliary VLANs

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have equivalent knowledge

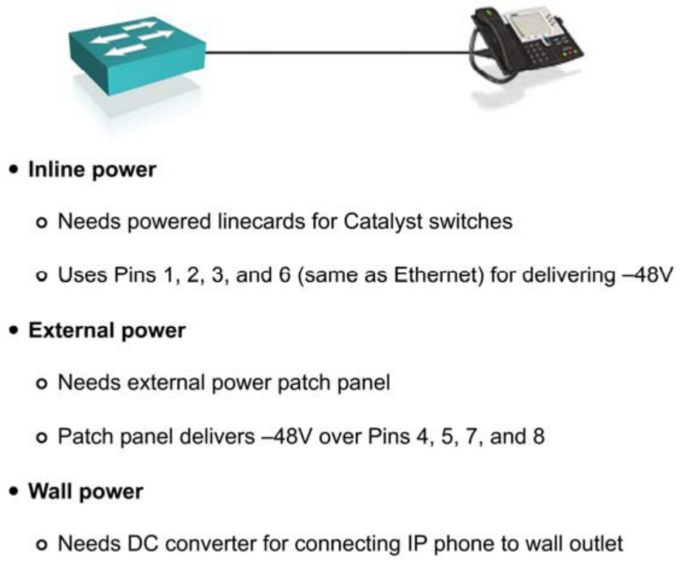
Outline

This lesson includes these sections:

- Overview
- Three Ways to Power IP Phones
- Configuring Inline Power
- Voice and Data Traffic
- Configuring an Auxiliary VLAN
- Summary

Three Ways to Power IP Phones

This section discusses how Cisco IP phones can receive power.

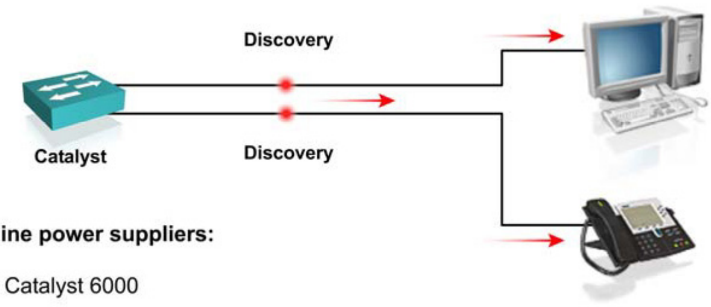


- **Inline power**
  - Needs powered linecards for Catalyst switches
  - Uses Pins 1, 2, 3, and 6 (same as Ethernet) for delivering –48V
- **External power**
  - Needs external power patch panel
  - Patch panel delivers –48V over Pins 4, 5, 7, and 8
- **Wall power**
  - Needs DC converter for connecting IP phone to wall outlet

Just like analog phones, Cisco IP phones require -48V DC to operate. There are three methods of providing power to the IP phones:

- **Inline Power:** The Catalyst 6500 with an inline Ethernet power module, and some other Catalyst models, can provide -48V DC to IP phones over a Cat 5 cable.
- **External Power:** A patch panel can be mounted next to the Catalyst 6500 to provide a centralized location from which power is supplied, preferably with a UPS backup.
- **Wall Power:** An external power transformer can be purchased and attached directly to the IP phone.

Catalyst Inline Power



- **Inline power suppliers:**
  - Catalyst 6000
  - Catalyst 4000
  - Catalyst 3524
- **Phones able to use inline power:**
  - Cisco 7960
  - Cisco 7940
  - Cisco 7910 and 7910+SW

Phone discovery helps simplify phone deployment. The three products that supply inline power (Catalyst 6000/6500, Catalyst 4000, and Catalyst 3524) are all capable of automatically sensing if the connected device is a Cisco IP Phone (7960, 7940, 7910, or 7910+SW) before sending power down the line.

The switch performs phone discovery by sending a specific tone down the wire towards the IP phone. In its un-powered state this tone is looped back by the phone (using normally closed relay contacts) and sent back to the switch. When the switch receives this tone it knows that the device connected is a Cisco IP phone, and it is safe to deliver –48V DC to the device. Only the Cisco IP phone exhibits this behavior. Other device(s) connected to the switch power do not loop the tone back and as such are safe from –48V DC being delivered to them. This hardware polling is done by the system at fixed intervals on a port by port basis until a link signal is seen or the system has been configured not to apply inline power to that port.

When the switch finds a phone using phone discovery, it applies power to the device. The Cisco IP phone powers up, energizing the relay and removing the loopback (normally closed relay becomes open) between transmit and receive pairs. It also sends a link signal to the switch. From this point it functions as a normal 10/100 Ethernet device.

Phone Discovery

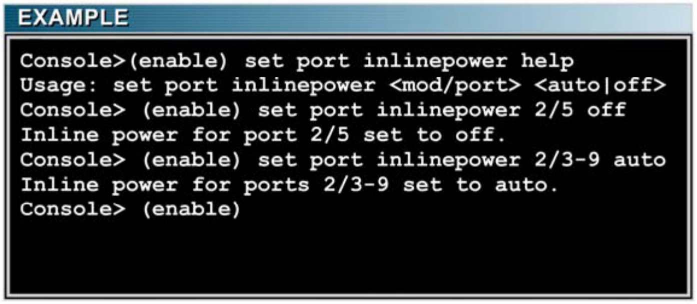
- Phone discovery
- Provide power
- Provide VLAN
- DHCP, get IP address

The following is the process used by Cisco IP phones to get inline power from the Catalyst switch:

- The unpowered phone plugs into a powered linecard port, with admin mode on the switch set to `auto`.
- The port senses the device using phone discovery mechanisms and reports it to the Supervisor.
- The Supervisor Engine checks the power budget and allocates a default amount, informing the port to apply –48V DC.
- The port turns on power to the phone and reports link up to the Supervisor.
- If the phone was powered by an external patch panel or wall power, the switch port reports link up to Supervisor.
- Phone begins CDP exchange with the switch and gets its VLAN ID (VVID) and reports actual power needed for operation.
- The phone sends a DHCP request on that VLAN for an IP address.

Configuring Inline Power

This section details how to configure the Catalyst 6500 to provide power over an existing Ethernet connection.



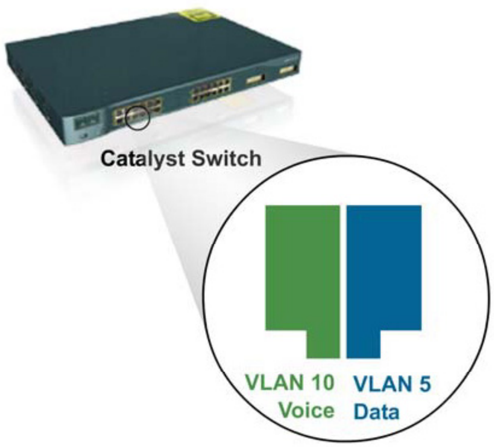
```
Console>(enable) set port inlinepower help
Usage: set port inlinepower <mod/port> <auto|off>
Console> (enable) set port inlinepower 2/5 off
Inline power for port 2/5 set to off.
Console> (enable) set port inlinepower 2/3-9 auto
Inline power for ports 2/3-9 set to auto.
Console> (enable)
```

The above example shows how to enable or disable inline power on a single port or on a range of ports with the syntax: **`set port inlinepower <`*`mod/port`*`> <auto | off>`**.

Notice that there is not an **`on`** option. This prevents power being applied to a port without verifying that the attached device is an IP phone.

Voice and Data Traffic

This section describes how voice and data traffic can coexist on the single physical connection between the IP phone and the Catalyst 6500 switch.



- **Phone and PC on 2 different VLANs**
- **Advantages:IP addressing, traffic separation, wiring**
- **Requires a single port capable of handling 2 VLANs**

The intuitive way to handle data and voice traffic on the same physical link/port is to logically separate them, using 2 different VLANs.

The following are some of the advantages of this solution:

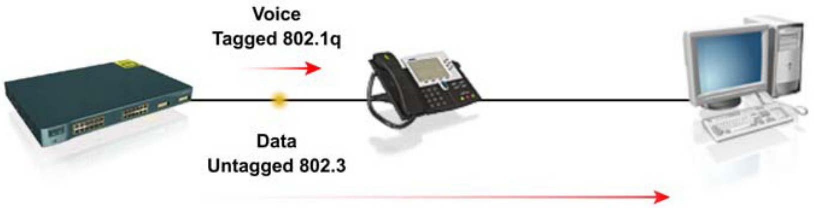
- From an addressing perspective, this solution allows scalability of the network. IP subnets have usually more than 50% (often more than 80%) of their IP addresses allocated. A separate VLAN (separate IP subnet) to carry the voice traffic allows for the introduction of a large number of new devices (the IP phones) in the network, without painful modifications of the IP addressing scheme.

- At the same time, it allows the logical separation of data and voice traffic that have different characteristics. This permits this data to be handled separately in the entire network.

- Two devices can be connected to the switch using only one physical port and one Ethernet cable between the wiring closet and the phone/PC location.

From the switch perspective, this solution requires an access port capable of handling 2 VLANs.

Auxiliary VLANs



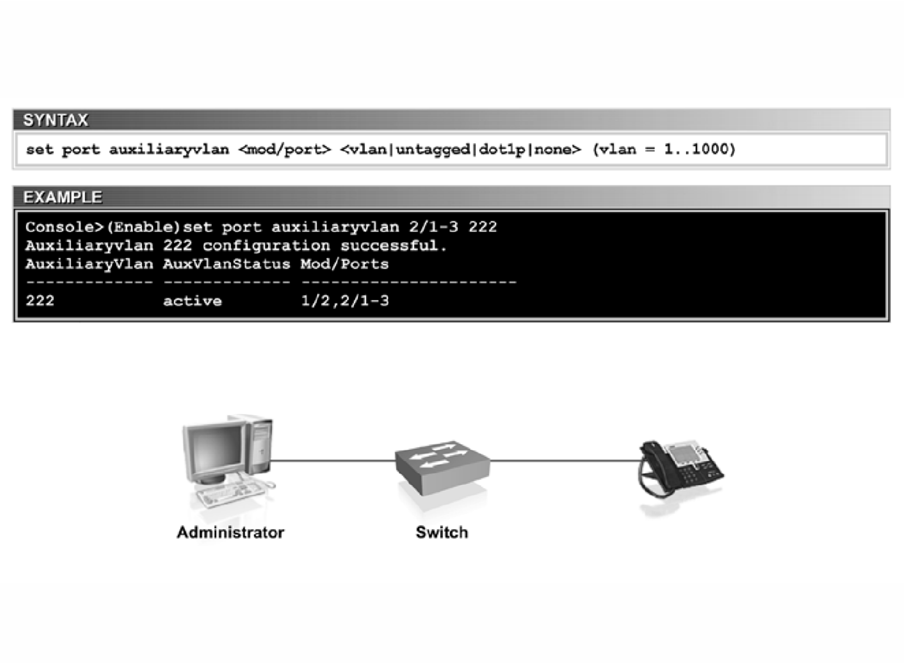
- An access port able to handle 2 VLANs
- Native VLAN (PVID) & Auxiliary VLAN (VVID)
- Hardware set to dot1q trunk

To accommodate both voice and data traffic on a port, the Catalyst can dynamically create an IEEE 802.1q trunk on a port connected to a Cisco IP phone and place data and voice traffic in different VLANs. The Voice VLAN ID (VVID) is communicated to the Cisco IP phone via CDP.

The IP phone then sends data traffic on the Port VLAN ID (PVID) and voice traffic on the VVID. Since voice traffic should typically take priority over data traffic, the phone applies Quality of Service (QoS) markings to the voice traffic. At layer 2, the voice traffic is marked with a Class of Service (CoS) value of 5, and at layer 3, the voice traffic is marked with an IP Precedence value of 5.

Configuring an Auxiliary VLAN

This section demonstrates how to configure Auxiliary VLANs on the Catalyst 6500 switch.



SYNTAX

```
set port auxiliaryvlan <mod/port> <vlan|untagged|dot1p|none> (vlan = 1..1000)
```

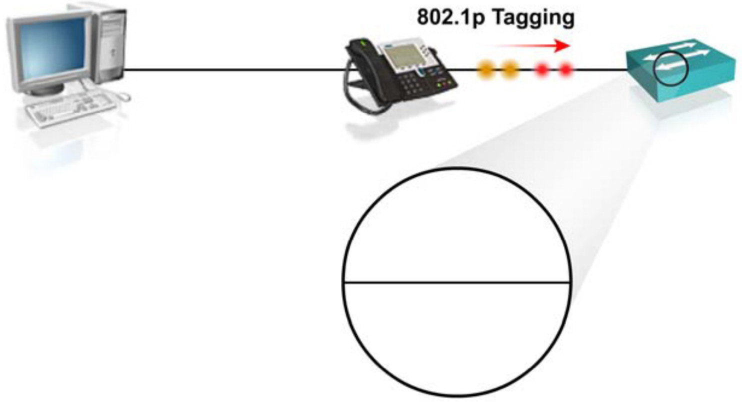
EXAMPLE

```
Console>(Enable)set port auxiliaryvlan 2/1-3 222
Auxiliaryvlan 222 configuration successful.
AuxiliaryVlan AuxVlanStatus Mod/Ports
------------- ------------- ----------------------
222           active        1/2,2/1-3
```

The VVID is configured in the Cat OS Version 5.5 and later with the command **`set port auxiliaryvlan`** ***`<mod/port>`*** ***`<vlan`*** **`| untagged | dot1p | none>`**.

In this example, the voice VLAN (VVID) has been set to a value of 222 for ports 2/1 through 2/3. When the phone powers up, the switch instructs it to be in VLAN 222. This command can be used to set the VVID on a per port basis, range of ports, or for an entire module.

Configuring an Auxiliary VLAN (cont.)

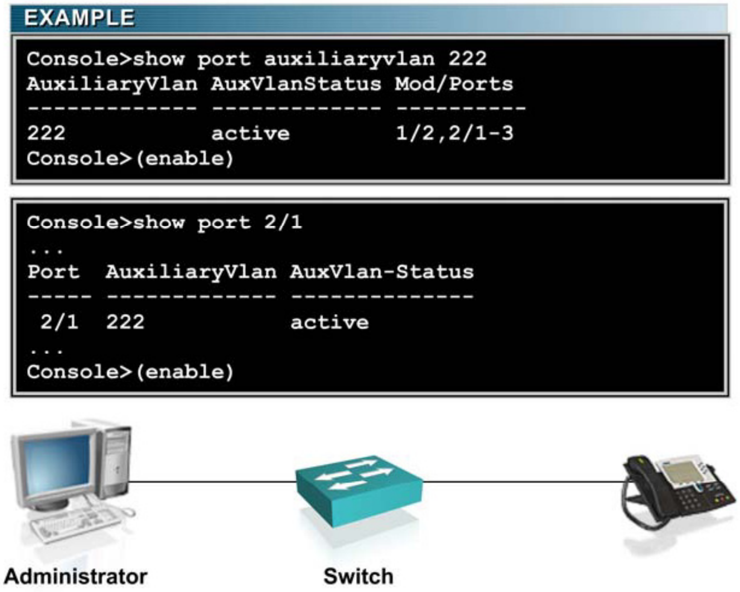


- vlan (1-1000): phone tags pkts with VLAN & priority (802.1q)
- untagged: phone sends untagged pkts
- dot1p: phone tags pkts with priority (802.1p) but no VLAN information
- none: phone uses internal configuration

The Auxiliary VLAN parameter can be set to one of the following:

- **1-1000:** Tells the IP phone that it has to send 802.1q tagged packets and specifies which VLAN they should be tagged for.
- **untagged:** Forces the IP phone to send untagged packets 802.3.
- **dot1p:** Tells the phone to send 802.1p packets. Packets are tagged with the priority information but will not carry the VLAN information.
- **none:** The switch does not to tell anything to the phone. The phone uses its internal configuration (usually its default configuration).

Verifying Auxiliary VLAN Configuration



```
Console>show port auxiliaryvlan 222
AuxiliaryVlan AuxVlanStatus Mod/Ports
------------- ------------- ----------
222           active        1/2,2/1-3
Console>(enable)
```

```
Console>show port 2/1
...
Port  AuxiliaryVlan AuxVlan-Status
----- ------------- --------------
 2/1  222           active
...
Console>(enable)
```

There are two ways to check the status of the Auxiliary VLAN on a port or module. In the first example, the command **show port auxiliaryvlan** <***vlan id***> shows the status of the Auxiliary VLAN and the module and ports it is active on.

The second example shows the output of the **show port** <***module/port***> command, which displays the module and port, the Auxiliary VLAN, and the status of the port.

Lesson Summary

This lesson accomplished the following:

- Listed the three methods of powering an IP phone
- Described how a Cisco IP phone can keep voice and data traffic separate
- Demonstrated the configuration of the Inline Power feature
- Demonstrated the configuration of Auxiliary VLANs

Overview of the WS-X6624-FXS and WS-X6608-T1/E1 Modules

Overview

The Catalyst 6500 can act as a gateway for both analog and digital voice circuits. This lesson examines the WS-X6624-FXS analog gateway interface module and the WS-X6608-T1/E1 digital interface modules.

Importance

The WS-X6624-FXS and WS-X6608-T1/E1 modules can be used to gain high port density for analog and digital voice connections within the Catalyst 6500.

Objectives

Upon completing this lesson, you will be able to:

- Describe the features of the WS-X6624-FXS and WS-X6608-T1/E1 modules
- List the applications of the WS-X6624-FXS and WS-X6608-T1/E1 modules

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Cisco Voice over Frame Relay, ATM, and IP (CVOICE) course or have equivalent knowledge

Outline

This lesson includes these sections:

- Overview
- The Catalyst WS-X6624-FXS Module
- The Catalyst WS-X6608-T1/E1 Module
- Summary

The Catalyst WS-X6624-FXS Module

- Connects analog telephony equipment into Cisco Catalyst 6000/6500 switches
- Analog phones participate in H.323 conferences
- Analog phones communicate with IP phones

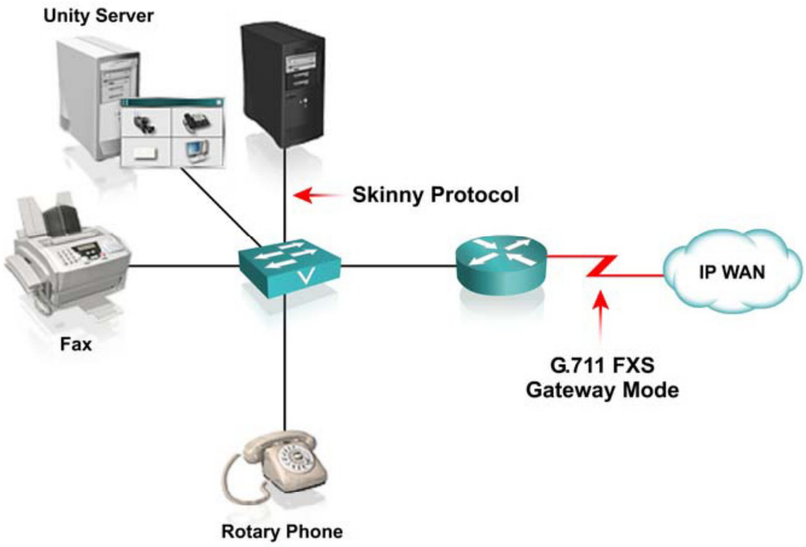
This section discusses the features and uses of the WS-X6624-FXS module.

The Catalyst 6000/6500 FXS Analog Interface Module provides 24 Foreign Exchange Station (FXS) ports for analog telephony devices. Like a PBX, an FXS port provides:

- -48 V DC
- Ringing voltage
- Dial tone
- Busy signals
- Recognition of dialed digits, either DTMF or pulse

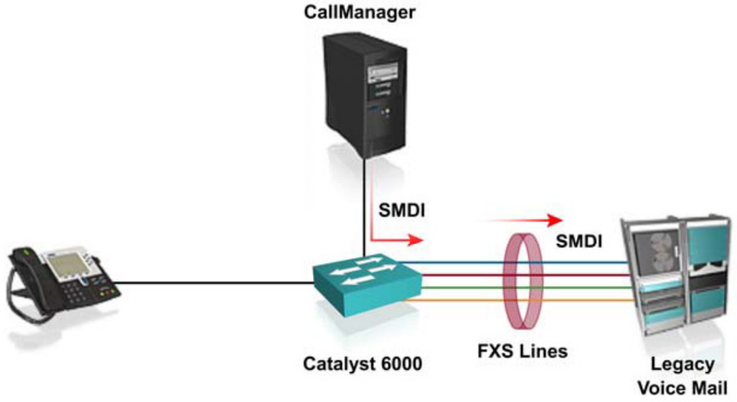
The analog devices do not plug directly into the front of the FXS module. Instead an RJ-21 connector connects to the front of the Catalyst 6500 chassis, and the other end is terminated on a cross-connect block.

Applications for the Catalyst WS-X6624-FXS Module



Most any type of analog telephony device can connect to the 6500's FXS module, just as it would connect to a PBX or the Public Switched Telephone Network (PSTN). Since the Catalyst 6500 can communicate with the Cisco CallManager, via the Skinny Station Protocol, these analog devices can participate in VoIP conversations.

Using the WS-X6624-FXS for Legacy Voice Mail Integration



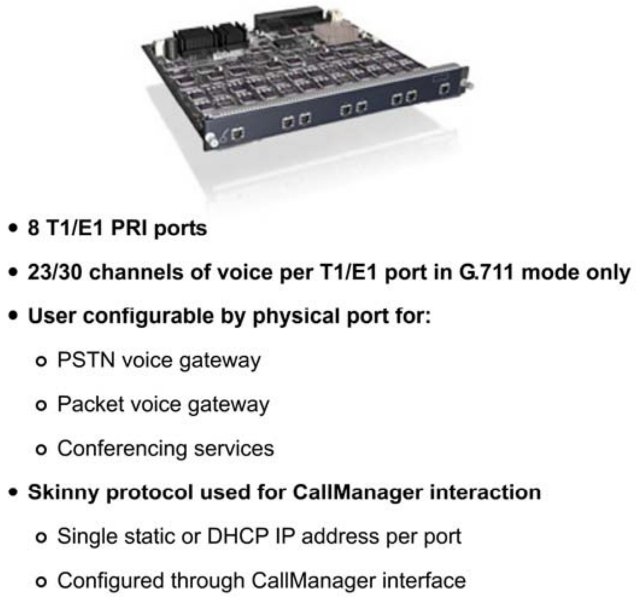
Analog FXS Catalyst 6000 Module:

- **Skinny controlled by CallManager**
- **Ports can be associated with voice mail extensions**

A common use of the WS-X6624-FXS module is to connect to legacy voice mail systems. These voice mail systems may need analog trunk connections. Using the Simplified Message Desk Interface (SMDI) protocol, these FXS ports can be associated with voice mail extensions known to CallManager, and CallManager can perform call control using the Skinny Station protocol.

The Catalyst WS-X6608-T1/E1 Modules

This section discusses the features and uses of the WS-X6608-T1/E1 module.



- **8 T1/E1 PRI ports**
- **23/30 channels of voice per T1/E1 port in G.711 mode only**
- **User configurable by physical port for:**
  - PSTN voice gateway
  - Packet voice gateway
  - Conferencing services
- **Skinny protocol used for CallManager interaction**
  - Single static or DHCP IP address per port
  - Configured through CallManager interface

The Voice T1 and Services module provides 8 T1 ports (192 channels or DS0 voice trunks) for connections to the Public Switched Telephone Network (PSTN) or PBX. As a gateway to the PSTN or legacy PBX, this module provides voice packetization services for delivery to/from the IP network. The module supports voice trunk protocols such as the ISDN Primary Rate Interface (PRI).

In addition to providing T1 or E1 ports, the module's digital signal processor (DSP) modules can be used as a resource by the CallManager. These DSPs can be used for such functions as conferencing or transcoding.

Lesson Summary

This lesson accomplished the following:

- Described the features of the WS-X6624-FXS and WS-X6608-T1/E1 modules

- Listed the applications of the WS-X6624-FXS and WS-X6608-T1/E1 modules

Voice Port Configuration and Verification

Overview

Both the WS-X6624-FXS and the WS-X6608-T1/E1 modules have their ports configured from the Cisco CallManager. This lesson details how to gain the information about these modules required to enter in CallManager.

Importance

The ability to determine a module's or a port's MAC address is essential for configuration within Cisco's CallManager.

Objectives

Upon completing this lesson, you will be able to:

- Configure voice ports on both the WS-X6624-FXS and WS-X6608-T1/E1 modules
- Verify voice module configuration
- Verify voice port configuration

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Cisco IP Telephony (CIPT) course or have equivalent knowledge

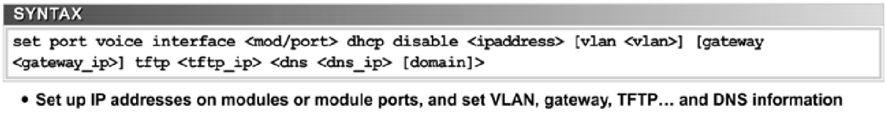
Outline

This lesson includes these sections:

- Overview
- Voice Port Configuration
- Displaying the Port Voice Interface Configuration
- Summary

Voice Port Configuration

This section discusses how to enable a voice port.



SYNTAX

```
set port voice interface <mod/port> dhcp disable <ipaddress> [vlan <vlan>] [gateway
<gateway_ip>] tftp <tftp_ip> <dns <dns_ip> [domain]>
```

- Set up IP addresses on modules or module ports, and set VLAN, gateway, TFTP... and DNS information

The voice port commands given in this lesson apply to both the WS-X6624-FXS and the WS-X6608-T1/E1 modules. It is critical to note that most of the voice port parameters are not configured from the Catalyst 6500's prompt but from Cisco CallManager (CCM).

If DHCP is enabled for a port, the port obtains all other configuration information from the TFTP server. When disabling DHCP on a port, you must specify some mandatory parameters as follows.

If you do not specify DNS parameters, the software uses the system DNS configuration on the Supervisor Engine to configure the port.

For 8-port T1/E1 PSTN interface modules only, you cannot specify more than one port at a time because a unique IP address must be set for each port.

To configure the port voice interface for DHCP, TFTP, and DNS servers, perform this task in privileged mode:

Table 5-1: Enabling a Voice Port

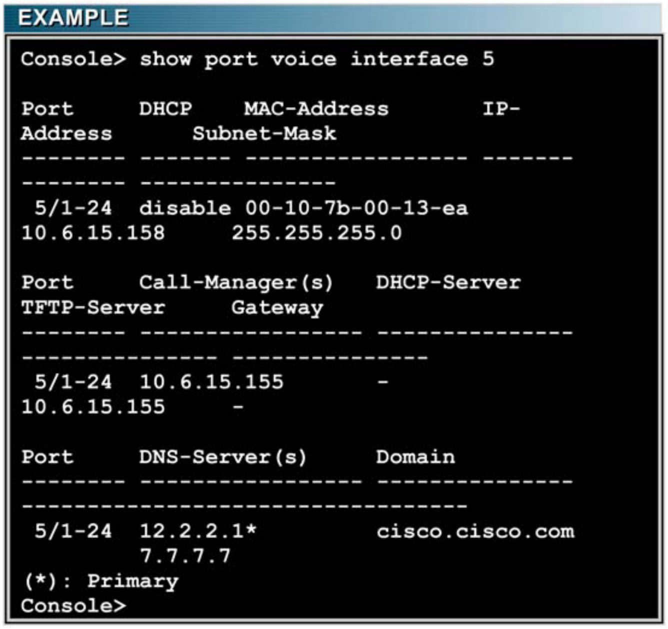
Task	Command
Configure port voice interface for DHCP, TFTP, and DNS servers.	**set port voice interface** *mod/port* **dhcp enable [vlan** *vlan***]** **set port voice interface** *mod/port* **dhcp disable {***ipaddrspec***} {tftp** *ipaddr***} [vlan** *vlan***] [gateway** *ipaddr***] [dns [***ipaddr***] [***domain_name***]]**

These examples show how to configure the port voice interface for DHCP, TFTP, and DNS servers:

```
Console> (enable) set port voice interface 7/1 dhcp enable
Port 7/1 DHCP enabled.

Console> (enable) set port voice interface 7/3 dhcp disable 171.68.111.41/24
tftp
173.32.43.11 dns 172.20.34.204 cisco.com
Port 7/3 dhcp disabled.
System DNS configurations applied.

Console> (enable) set port voice interface 7/4-6 dhcp enable vlan 3
Vlan 3 configuration successful
Ports 7/4-6 DHCP enabled.
Console> (enable)
```

Displaying the Port Voice Interface Configuration

This sections demonstrates how to obtain the MAC address(es) of a module for use by Cisco CallManager.



EXAMPLE

```
Console> show port voice interface 5

Port     DHCP    MAC-Address       IP-
Address      Subnet-Mask
-------- ------- ----------------- -------
-------- ---------------
 5/1-24  disable 00-10-7b-00-13-ea
10.6.15.158     255.255.255.0

Port     Call-Manager(s)   DHCP-Server
TFTP-Server     Gateway
-------- ----------------- ---------------
-------------- ---------------
 5/1-24  10.6.15.155       -
10.6.15.155     -

Port     DNS-Server(s)     Domain
-------- ----------------- ---------------
---------------------------------
 5/1-24  12.2.2.1*         cisco.cisco.com
         7.7.7.7
(*): Primary
Console>
```

To display the port voice interface configuration, perform this task in privileged mode:

Table 5-2: Verifying a Voice Port

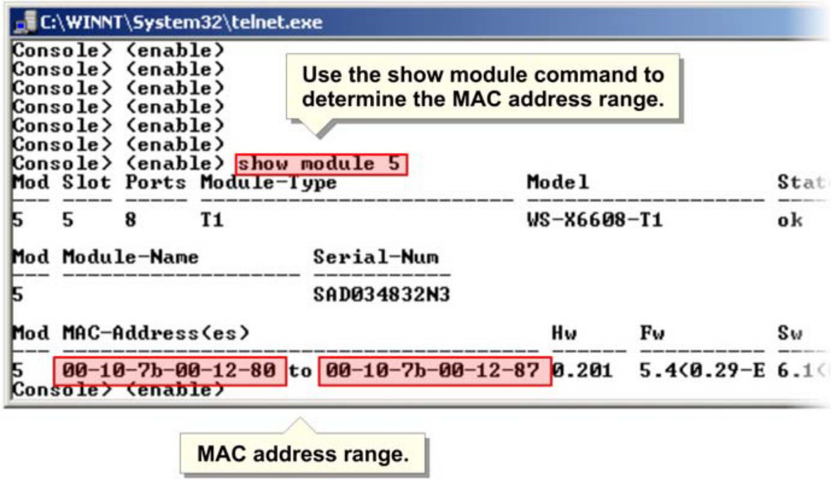
Task	Command
Display the port voice interface configuration.	`show port voice interface [mod[/port]]`

This example shows how to display the port voice interface configuration (this display is from the 24-port FXS analog interface module):

```
Console> show port voice interface 5
Port     DHCP    MAC-Address       IP-Address      Subnet-Mask
-------- ------- ----------------- --------------- ---------------
 5/1-24  disable 00-10-7b-00-13-ea 10.6.15.158     255.255.255.0
 Port     Call-Manager(s)   DHCP-Server     TFTP-Server     Gateway
-------- ----------------- --------------- --------------- ---------------
 5/1-24  10.6.15.155       -               10.6.15.155     -
 Port     DNS-Server(s)     Domain
-------- ----------------- -------------------------------------------------
 5/1-24  12.2.2.1*         cisco.cisco.com
         7.7.7.7
(*): Primary
```

```
Console> (enable)
```

Determining a Voice Module’s MAC Address



```
Console> (enable)
Console> (enable)
Console> (enable)
Console> (enable)
Console> (enable)
Console> (enable)
Console> (enable) show module 5
Mod Slot Ports Module-Type                   Model                Stat
--- ---- ----- ----------------------------- -------------------- ----
5   5    8     T1                            WS-X6608-T1          ok

Mod Module-Name          Serial-Num
--- -------------------- -----------
5                        SAD034832N3

Mod MAC-Address(es)                          Hw     Fw         Sw
--- ---------------------------------------- ------ ---------- ----
5   00-10-7b-00-12-80 to 00-10-7b-00-12-87 0.201  5.4(0.29-E 6.1(
Console> (enable)
```

Use the show module command to determine the MAC address range.

MAC address range.

In order for the FXS and T1/Service modules to be used by Cisco CallManager, the CCM must be configured with the MAC address(es) of the module. The MAC address(es) may be determined by issuing the command **`show module <module_number>`**.

In the above example, the **`show module`** command is being executed for a WS-X6608-T1 module. Notice that instead of a single MAC address (as would be displayed for a WS-X6624-FXS module) a range of eight MAC addresses is displayed. Since each of the T1 module’s ports and/or DSPs can be used for different roles, each port has a unique MAC address. Therefore, each port can be uniquely tasked by the CallManager.

Lesson Summary

This lesson accomplished the following:

- Illustrated the configuration of voice ports for the WS-X6624-FXS and WS-X6608-T1/E1 modules

- Verified voice module configuration

- Verified voice port configuration

summary

- Described the configuration of Inline Power support for Cisco IP phones
- Described the configuration of Auxiliary VLANs on the Catalyst 6500 Series switch
- Described the WS-X6624-FXS Module
- Described the WS-X6608-T1/E1 Modules

Review Questions

Q1) Which of the following is used by the Catalyst 6500 to communicate VVID information to a Cisco IP phone?

A) 802.1p

B) CDP

C) VTP

D) PAgP

Answer: B

Q2) When inline power is supplied by the Catalyst 6500's Ethernet module, which pins are used?

A) 1, 2, 3, 4 (The first four pins)

B) 4, 5, 6, 7 (All pins not used by Ethernet)

C) 5, 6, 7, 8 (The last four pins)

D) 1, 2, 3, 6 (The same pins used by Ethernet)

Answer: D

Q3) Which of the following is NOT provided by an FXS port?

A) Dial tone

B) Ringing voltage

C) Digit dialing

D) –48 V DC

Answer: C (While the FXS port can interpret dialed digits, it does not generate dialed digits.)

Q4) In order for Cisco CallManager to use the DSPs associated with a port on a WS-X6608-T1 module, what information must the CallManager be given?

A) MAC address of the entire module

B) MAC address of a specific port

C) The slot number of the module

D) The serial number of the module

Answer: B

Q5) What information is NOT returned from the command **`show port voice interface 5`**?

A) Serial number of module 5

B) IP address(es) of associated CallManager(s)

C) IP address of associated TFTP server

D) IP address(es) of associated DNS server(s)

Answer: A

Configuring Security Services

Overview

Each network should have a security policy stating what types of traffic are allowed. The focus of this module is to provide additional tools, above and beyond the security fundamentals of user and enable passwords, on the Catalyst 6500 series switch. Several security options will be explored including IP permit lists, Layer 3 protocol filtering and VLAN access control lists (VACLs).

Upon completing this module, you will be able to:

- Protect access to the management sc0 interface via telnet and other protocols
- Restrict specific Layer 3 protocols from specific switch ports
- Understand the concept of a VACL, and how it is used in the switch

Outline

The module contains these lessons:

- IP Permit Lists
- Layer 3 Protocol Filtering
- VLAN Access Control Lists (VACLs)

IP Permit Lists

Overview

Once the Catalyst switch has been configured with an IP address for sc0, and has been given a default gateway, telnet access is possible allowing remote management. The problem, however, is that the switch is vulnerable to connections from unauthorized hosts, as well as the authorized ones.

Importance

IP permit lists on the Catalyst switch are particularly important when the VLAN used for management also contains users. Using the IP permit lists will help defend the sc0 from unauthorized access.

Objectives

Upon completing this lesson, you will be able to:

- Understand the function of an IP permit list
- Configure an IP Permit List on a 6500 series Catalyst switch
- Enable and Disable an IP Permit List

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Be familiar with the initial setup of a Catalyst 6500 series switch including password setup for user and enable mode.
- Know how to configure the Catalyst sc0 interface with an IP address for remote management
- Understand how to setup a password for telnet access to the switch

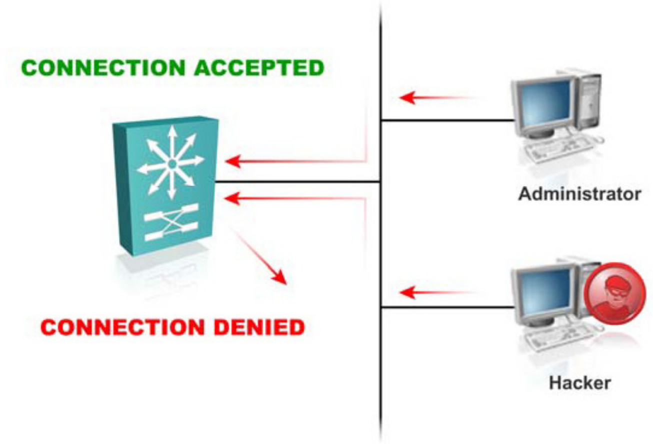
Outline

This lesson includes these sections:

- Overview
- Understanding IP Permit Lists
- Configuring IP Permit Lists
- Enabling IP Permit Lists
- Summary
- Assessment (Case Study)

Overview

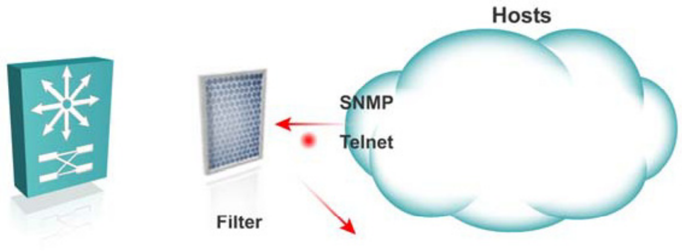
It is important to have remote access to the switch for management, but it is also important to restrict which IP hosts can telnet to the switch for security reasons. Although IP source addresses may be spoofed, the permit list is an excellent first line of defense against unauthorized access to sc0.



The IP Permit list is the warden to sc0 on your switch. Before connections are allowed, the permit list is checked before access is allowed.

Understanding IP Permit Lists

This section details the purpose of an IP permit list, including which TCP/IP services are not affected by the IP permit list.



IP permit lists prevent inbound Telnet and SNMP access to the switch from unauthorized source IP addresses. All other TCP/IP services (such as IP traceroute and IP ping) continue to work normally when you enable the IP permit list. Outbound Telnet, TFTP, and other IP-based services are unaffected by the IP permit list.

When a Telnet session is attempted from an unauthorized source IP address, the connection is denied by the switch. SNMP requests from unauthorized IP addresses receive no response.

If you want to log unauthorized access attempts to the console or a syslog server, you must change the logging severity level for IP. If you want to generate SNMP traps when unauthorized access attempts are made, you must enable IP permit list (ippermit) SNMP traps. Both of these concepts will be covered in this lesson.

Multiple access attempts from the same unauthorized host only trigger notifications every ten minutes.

You can configure up to 100 entries in the permit list. Each entry includes whether the IP address is part of the SNMP permit list, Telnet permit list, or both lists.

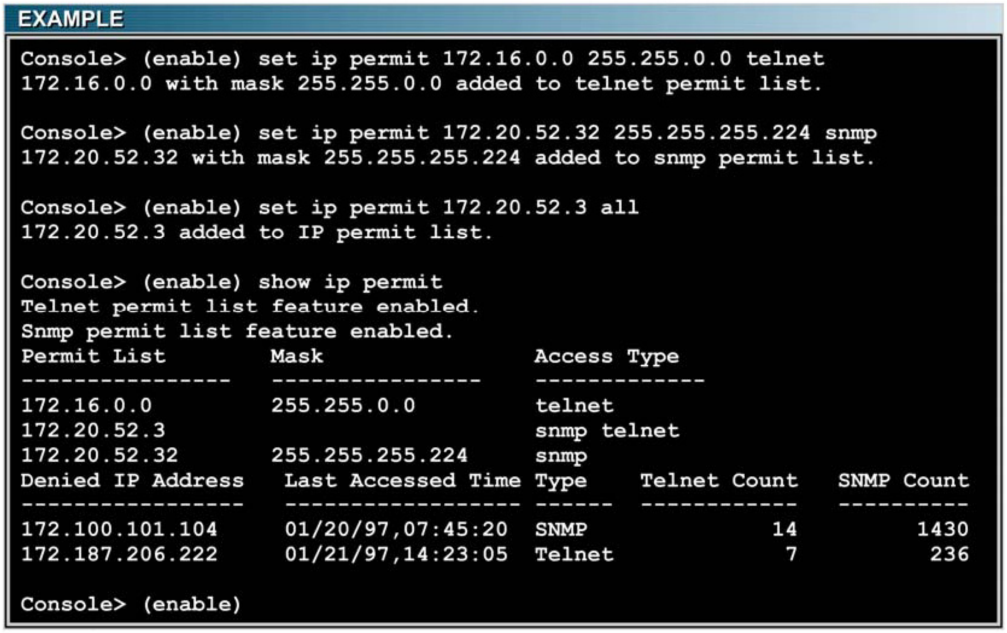
If you do not specify the mask for an IP permit list entry, or if you enter a host name instead of an IP address, the mask has an implicit value of all bits set to one (255.255.255.255 or 0xffffffff), which matches only the IP address of that host.

If you do not specify SNMP or Telnet for the type of permit list for the IP address, the IP address is added to both the SNMP and Telnet permit lists.

You can specify the same IP address in more than one entry in the permit list if the masks are different. The mask is applied to the address before it is stored in NVRAM, so that entries that have the same effect (but different addresses) are not stored. When you add such an address to the IP permit list, the system displays the address after the mask is applied.

Configuring IP Permit Lists

This section will cover the commands behind creating an IP permit list.



```
Console> (enable) set ip permit 172.16.0.0 255.255.0.0 telnet
172.16.0.0 with mask 255.255.0.0 added to telnet permit list.

Console> (enable) set ip permit 172.20.52.32 255.255.255.224 snmp
172.20.52.32 with mask 255.255.255.224 added to snmp permit list.

Console> (enable) set ip permit 172.20.52.3 all
172.20.52.3 added to IP permit list.

Console> (enable) show ip permit
Telnet permit list feature enabled.
Snmp permit list feature enabled.
Permit List        Mask                Access Type
----------------   ----------------    -------------
172.16.0.0         255.255.0.0         telnet
172.20.52.3                            snmp telnet
172.20.52.32       255.255.255.224     snmp
Denied IP Address   Last Accessed Time Type    Telnet Count   SNMP Count
-----------------   ------------------ ------  ------------   ----------
172.100.101.104     01/20/97,07:45:20  SNMP              14         1430
172.187.206.222     01/21/97,14:23:05  Telnet             7          236

Console> (enable)
```

An IP address can be added to the SNMP permit list, the Telnet permit list, or both lists.

To add IP addresses to an IP permit list, perform this task in privileged mode:

Table 6-1: Permit List Configuration and Verification Commands

	Task	Command
Step 1	Specify the IP addresses to add to IP permit list.	**set ip permit** *ip_address* **[***mask***] [all | snmp | telnet]**
Step 2	Verify the IP permit list configuration.	**show ip permit**

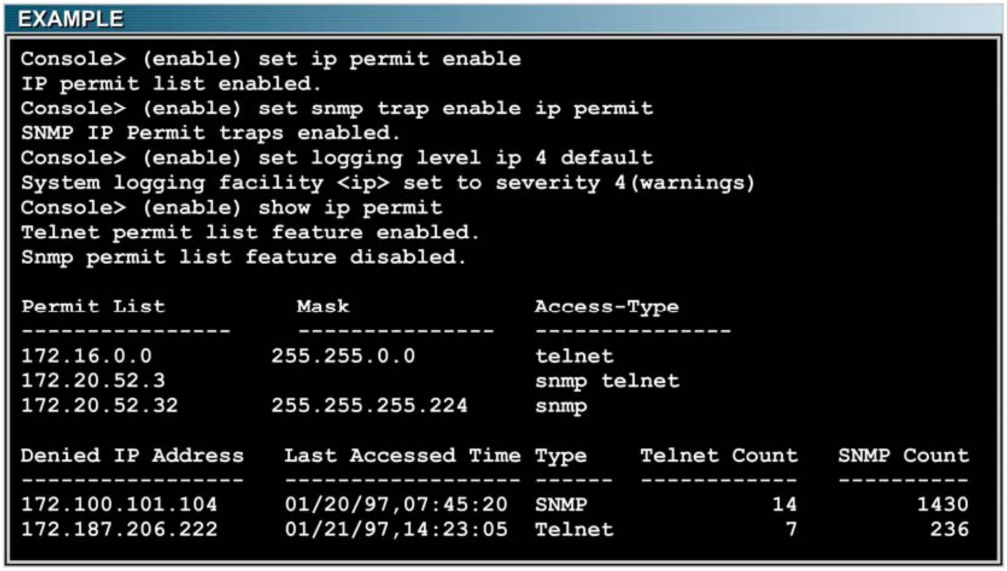
Note You can also use the **set security acl** command to set permit lists. We will discuss this option later in this chapter.

This example shows how to add IP addresses to an IP permit list and verify the configuration:

```
Console> (enable) set ip permit 172.16.0.0 255.255.0.0 telnet
172.16.0.0 with mask 255.255.0.0 added to telnet permit list.

Console> (enable) set ip permit 172.20.52.32 255.255.255.224 snmp
172.20.52.32 with mask 255.255.255.224 added to snmp permit list.
Console> (enable) set ip permit 172.20.52.3 all
172.20.52.3 added to IP permit list.
Console> (enable) show ip permit
Telnet permit list feature enabled.
Snmp permit list feature enabled.
Permit List        Mask                 Access Type
----------------   ----------------    -------------
172.16.0.0         255.255.0.0         telnet
172.20.52.3                            snmp telnet
172.20.52.32       255.255.255.224     snmp
Denied IP Address   Last Accessed Time Type    Telnet Count   SNMP Count
-----------------   ------------------ ------  ------------   ----------
172.100.101.104     01/20/97,07:45:20  SNMP              14         1430
172.187.206.222     01/21/97,14:23:05  Telnet             7          236
 Console> (enable)
```

Enabling IP Permit Lists

After the IP permit list is created, it needs to be enabled in order to function.



EXAMPLE

```
Console> (enable) set ip permit enable
IP permit list enabled.
Console> (enable) set snmp trap enable ip permit
SNMP IP Permit traps enabled.
Console> (enable) set logging level ip 4 default
System logging facility <ip> set to severity 4(warnings)
Console> (enable) show ip permit
Telnet permit list feature enabled.
Snmp permit list feature disabled.

Permit List          Mask              Access-Type
----------------     ---------------   ---------------
172.16.0.0         255.255.0.0         telnet
172.20.52.3                            snmp telnet
172.20.52.32       255.255.255.224     snmp

Denied IP Address   Last Accessed Time Type    Telnet Count   SNMP Count
-----------------   ------------------ ------  ------------   ----------
172.100.101.104     01/20/97,07:45:20  SNMP              14         1430
172.187.206.222     01/21/97,14:23:05  Telnet             7          236
```

It is possible to enable either the SNMP permit list, the Telnet permit list, or both lists. If a permit list is not specified, both the SNMP and Telnet permit lists are enabled.

Warning Before enabling IP permit list, make sure the IP address of the network management system is added to the permit list, especially when configuring through SNMP. Failure to do so could result in the Telnet connection being dropped by the switch once the configuration change is made. It is recommended that the permit list is disabled before clearing IP permit entries or host addresses.

To enable an IP permit list on the switch, perform this task in privileged mode:

Table 6-2: Permit List Commands

	Task	Command
Step 1	Enable IP permit list.	**`set ip permit enable [all	snmp	telnet]`**
Step 2	If desired, enable the IP permit trap to generate traps for unauthorized access attempts.	**`set snmp trap enable ippermit`**
Step 3	If desired, configure the logging level to see syslog messages for unauthorized access attempts.	**`set logging level ip 4 default`**

Step 4	Verify the IP permit list configuration.	**`show ip permit`**
		`show snmp`

This example shows how to enable an IP permit list and verify the configuration:

```
Console> (enable) set ip permit enable
IP permit list enabled.
Console> (enable) set snmp trap enable ip permit
SNMP IP Permit traps enabled.
Console> (enable) set logging level ip 4 default
System logging facility <ip> set to severity 4(warnings)
Console> (enable) show ip permit
Telnet permit list feature enabled.
Snmp permit list feature disabled.

Permit List          Mask              Access-Type
----------------     ---------------   ---------------
172.16.0.0           255.255.0.0       telnet
172.20.52.3                            snmp telnet
172.20.52.32         255.255.255.224   snmp

Denied IP Address   Last Accessed Time Type    Telnet Count   SNMP Count
-----------------   ------------------ ------  ------------   ----------
172.100.101.104     01/20/97,07:45:20  SNMP              14         1430
172.187.206.222     01/21/97,14:23:05  Telnet             7          236

Console> (enable) show snmp
RMON:                       Disabled
Extended Rmon:              Extended RMON module is not present
Traps Enabled:
ippermit
Port Traps Enabled: None

Community-Access     Community-String
----------------     --------------------
read-only            public
read-write           private
read-write-all       secret

Trap-Rec-Address                           Trap-Rec-Community
----------------------------------------   --------------------
Console> (enable)
```

To disable the IP permit list on the switch, perform this task in privileged mode:

Table 6-3: Disabling a Permit List

	Task	Command
Step 1	Disable IP permit list on the switch.	**set ip permit disable [all │ snmp │ telnet]**
Step 2	Verify the IP permit list configuration.	**show ip permit**

This example shows how to disable the IP permit list:

```
Console> (enable) set ip permit disable
IP permit list disabled.
Console> (enable)
```

Clearing an IP Permit List Entry

An IP address can be cleared from the SNMP permit list, the Telnet permit list, or both lists. If permit list is unspecified when an address is cleared, the IP address is deleted from both permit lists.

Warning Disable IP permit list before clearing IP permit entries or host addresses to prevent the management connection from being dropped by the switch.

To clear an IP permit list entry, perform this task in privileged mode:

Table 6-4: Clearing an IP Permit List Entry

	Command	Description
Step 1	Disable IP permit list.	**set ip permit disable [all │ snmp │ telnet]**
Step 2	Specify the IP address to remove from the IP permit list.	**clear ip permit** {*ip_address* [*mask*] **│ all**} **[all │ snmp │ telnet]**
Step 3	Verify the IP permit list configuration.	**show ip permit**

This example shows how to clear an IP permit list entry:

```
Console> (enable) set ip permit disable all
Console> (enable) clear ip permit 172.100.101.102
172.100.101.102 cleared from IP permit list.
Console> (enable) clear ip permit 172.160.161.0 255.255.192.0 snmp
172.160.128.0 with mask 255.255.192.0 cleared from snmp permit list.
Console> (enable) clear ip permit 172.100.101.102 telnet
172.100.101.102 cleared from telnet permit list.
Console> (enable) clear ip permit all
IP permit list cleared.
Console> (enable)
```

Lesson Summary

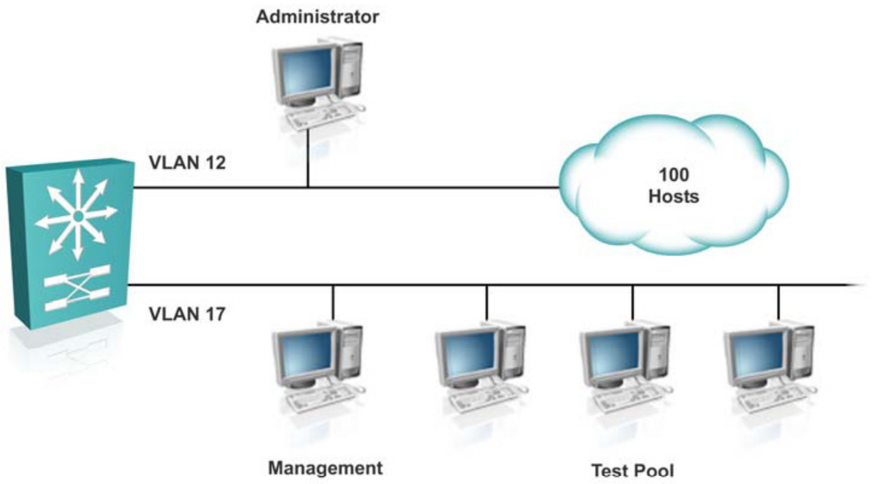
This lesson accomplished the following:

- Explained the function of an IP permit list
- Configured an IP Permit List on a 6500 series Catalyst switch
- Enabled and Disabled an IP Permit List

This lesson discussed how to restrict access to the sc0 interface on the Catalyst switch using the IP Permit List feature.

Case Study

Look at the management stations in the network below, and notice their VLAN numbers. Answer the questions that follow.



Would an IP Permit List that allowed telnet access from the entire subnet be appropriate?

What is the risk of allowing SNMP access from the entire subnet?

Layer 3 Protocol Filtering

Overview

Layer 3 Protocol Filtering prevents certain protocol traffic from being forwarded out switch ports. Broadcast and unicast flood traffic is filtered based on the membership of ports in different protocol groups. This lesson focuses on configuring a Catalyst switch to perform protocol filtering on a port-by-port basis.

Importance

This filtering is in addition to the filtering provided by port-VLAN membership. By adding this additional filtering, excess traffic is reduced, and the security of the switched environment is increased.

Objectives

Upon completing this lesson, you will be able to:

- Define Layer 3 Protocol Filtering
- Configure Layer 3 Protocol Filtering

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Basic configuration and management of a Catalyst 6000 series switch

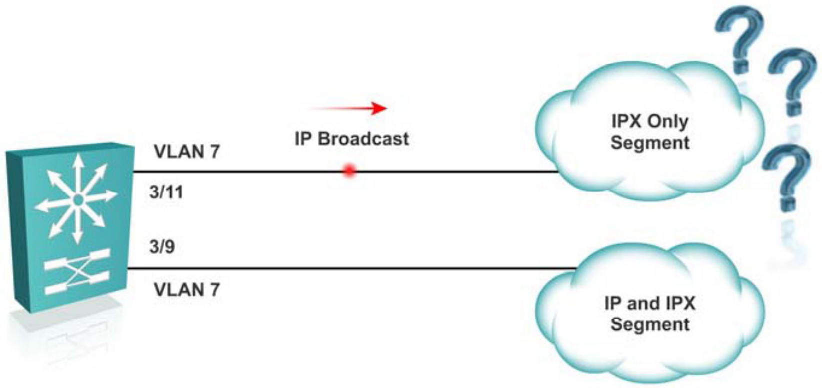
Outline

This lesson includes these sections:

- Overview
- Understanding Layer 3 Protocol Filtering
- Configuring Layer 3 Protocol Filtering
- Summary
- Assessment (Case Study): ACME, Inc.

Overview

This section introduces the concept of Layer 3 Protocol Filtering, which improves security as well as performance.



When a host that supports only Internetwork Packet Exchange (IPX) is connected to a switch port, the host will receive broadcast traffic for IPX as well as IP. By training the switch to allow only specific Layer 3 protocols to certain ports, the number of useless broadcasts is reduced on the port. This also hides the non-required traffic from unauthorized users.

Understanding Layer 3 Protocol Filtering

This section details the operations of Layer 3 Protocol Filtering within the Catalyst switch.

Layer 3 Filtering Supported On:

- **Nontrunking Ethernet**
- **Fast Ethernet**
- **Gigabit Ethernet**

Trunking Ports:

- **Always members of all protocol groups**
- **Cannot be protocol filtered**

Layer 2 Protocols:

- **Always members of all protocol groups**
- **Cannot be protocol filtered**

Protocol Group Modes:

- **On – port receives all flood traffic for specific protocol**
- **Off - port does not receive any flood traffic for specific protocol**
- **Auto – port is added if protocol traffic received on specific port**

Layer 3 Protocol Filtering prevents certain protocol traffic from being forwarded out specific switch ports. By default, broadcast and unicast flood traffic is filtered based on the membership of ports in different protocol groups. This filtering is in addition to the filtering provided by port-VLAN membership. Layer 3 Protocol Filtering is supported only on nontrunking Ethernet, Fast Ethernet, and Gigabit Ethernet ports

Trunking ports are always members of all protocol groups. Filtering is not performed on trunk ports. Layer 2 protocols, such as Spanning Tree Protocol (STP) and Cisco Discovery Protocol (CDP), are not affected by Layer 3 Protocol Filtering. Dynamic ports and ports that have port security enabled are members of all protocol groups.

You can configure a port with any one of these modes for each protocol group: **`on`**, **`off`**, or **`auto`**. If the configuration is set to **`on`**, the port receives all the flood traffic for that protocol. If the configuration is set to **`off`**, the port does not receive any flood traffic for that protocol.

If the configuration is set to **`auto`**, the port is added to the group only after packets of the specific protocol are received on that port. With autolearning, ports become members of the protocol group only after receiving packets of the corresponding protocol from the device attached to that port. Autoconfigured ports are removed from the protocol group if no packets are received for that protocol within 60 minutes. Ports are also removed from the protocol group when the Supervisor Engine detects that the link is down on the port.

For example, if a host that supports both IP and Internetwork Packet Exchange (IPX) is connected to a switch port configured as **`auto`** for IPX, but the host is transmitting only IP traffic, the port to which the host is connected will not forward any IPX flood traffic to the host. However, if the host sends an IPX packet, the Supervisor Engine software detects the protocol traffic and the port is added to the IPX group, allowing the port to receive IPX flood traffic. If the host stops sending IPX traffic for more than 60 minutes, the port is removed from the IPX protocol group.

By default, ports are configured to **`on`** for the IP protocol group. Typically, you should only configure a port to **`auto`** for IP if there is a directly connected end station out the port. The default port configuration for IPX and Group is **`auto`**.

With Layer 3 Protocol Filtering enabled, ports are identified on a protocol basis. A port can be a member of one or more of the protocol groups. Flood traffic for each protocol group is forwarded out a port only if that port belongs to the appropriate protocol group.

Packets are classified into the following protocol groups:

- IP
- IPX
- AppleTalk, DECnet, and Banyan VINES ("group")
- Packets not belonging to any of these protocols

Default Layer 3 Protocol Filtering Configuration

Table 6-5 shows the default Layer 3 Protocol Filtering configuration.

Table 6-5: Layer 3 Protocol Filtering Default Configuration Feature

Feature	Default Value
Layer 3 protocol filtering	**`Disabled`**
ip mode	**`on`**
ipx mode	**`auto`**
group mode	**`auto`**

Configuring Layer 3 Protocol Filtering

This section will discuss the process of configuring Layer 3 Protocol Filtering on the Catalyst switch.

To configure Layer 3 Protocol Filtering on Ethernet ports, perform this task in privileged mode:

Table 6-6: Layer 3 Protocol Filtering Commands

	Command	Description
Step 1	Enable Layer 3 protocol filtering on the switch.	**`set protocolfilter enable`**
Step 2	Set the protocol membership of the desired ports.	**set port protocol** *mod_num/port_num* **{ip | ipx | group} {on | off | auto}**
Step 3	Verify the port filtering configuration.	**show port protocol** [*mod_num*[/*port_num*]]

This example shows how to enable Layer 3 Protocol Filtering, set the protocol membership of ports, and verify the configuration:

```
Console> (enable) set protocolfilter enable
Protocol filtering enabled on this switch.
Console> (enable) set port protocol 7/1-4 ip on
IP protocol set to on mode on ports 7/1-4.
Console> (enable) set port protocol 7/1-4 ipx off
IPX protocol disabled on ports 7/1-4.
Console> (enable) set port protocol 7/1-4 group auto
Group protocol set to auto mode on ports 7/1-4.
```

```
Console> (enable) show port protocol 7/1-4
Port     Vlan       IP       IP Hosts IPX      IPX Hosts Group    Group Hosts
-------- ---------- -------- -------- -------- --------- -------- -----------
7/1      4           on       1        off     0         auto-off        0
7/2      5           on       1        off     0         auto-on         1
7/3      2           on       1        off     0         auto-off        0
7/4      4           on       1        off     0         auto-on         1
Console> (enable)
```

Disabling Layer 3 Protocol Filtering

To disable Layer 3 Protocol Filtering, perform this task in privileged mode:

Table 6-7: Disabling Protocol Filtering Commands

Command	Description
Disable Layer 3 protocol filtering on the switch.	**`set protocolfilter disable`**

Summary

This lesson accomplished the following:

- Define Layer 3 Protocol Filtering
- Configure Layer 3 Protocol Filtering

This lesson discussed how Layer 3 Protocol Filtering operates within a Catalyst switched environment, as well as how to configure, enable and disable Layer 3 protocol filters on the switch.

Case Study: ACME, Inc.

ACME Inc. tracks all of its injury liability claims on 3 Novell 3.12 servers. There are approximately 20 users who need access to these Novell servers. The entire company, which includes 450 people, needs access to corporate email and Internet access using the IP protocol. The company is in one building, and is using two 6509 switches for the core and distribution layer.

Could ACME, Inc. benefit from Layer 3 Protocol Filtering?

If so, what would be the appropriate way to implement it so that Layer 3 Protocol Filtering reduces broadcast traffic for most of the company's workstations?

VLAN Access Control Lists (VACLs)

Overview

Router-based IOS access control lists (ACLs) restrict or allow routed traffic *between* VLANs. VLAN ACLs (VACLs) control *all* packets, even within the same VLAN. This lesson will show how the VACLs operate, and how they are applied on the switch.

Importance

In order to control intra-VLAN traffic, or to forward specific VLAN traffic to an Intrusion Detection System (IDS), the power of VACLs can be leveraged.

Objectives

Upon completing this lesson, you will be able to:

- Describe how a VACL operates
- Learn how to configure a VACL
- Understand situations where a VACL can be effective

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Sound understanding and working knowledge of VLANs, IOS Access Lists
- Working knowledge of Catalyst switching functions

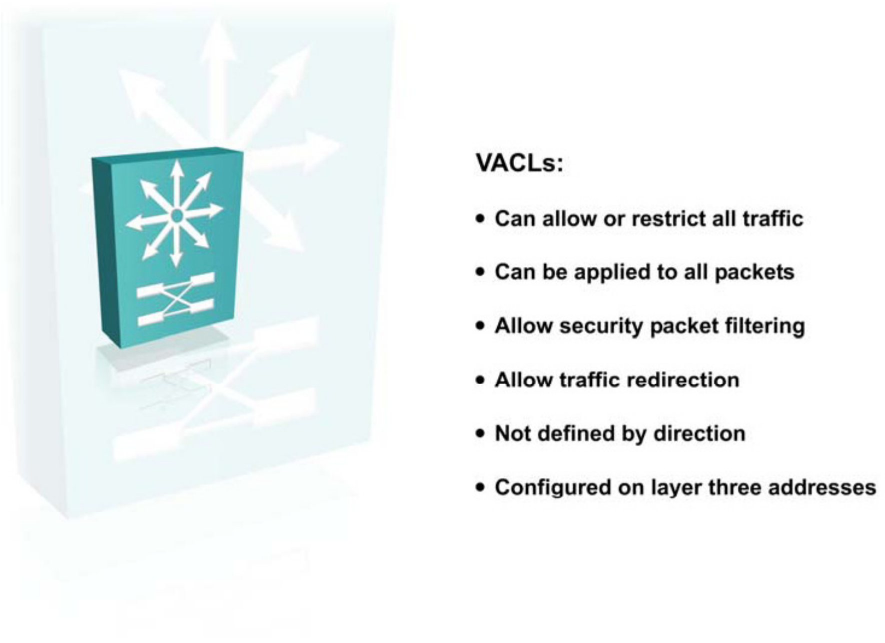
Outline

This lesson includes these sections:

- Overview
- Using VACLs in Your Network
- Configuring VACLs
- Showing VACL-to-VLAN Mapping
- Redirecting Broadcast Traffic
- Restricting DHCP Response
- Forwarding Traffic for Intrusion Detection
- Summary
- Assessment (Case Study): The Tech Expo Floor

Overview

Using a Policy Feature Card (PFC) there are additional tools for limiting and controlling traffic within a switch. The lesson discusses one of the features, namely VACLs.



VACLs:

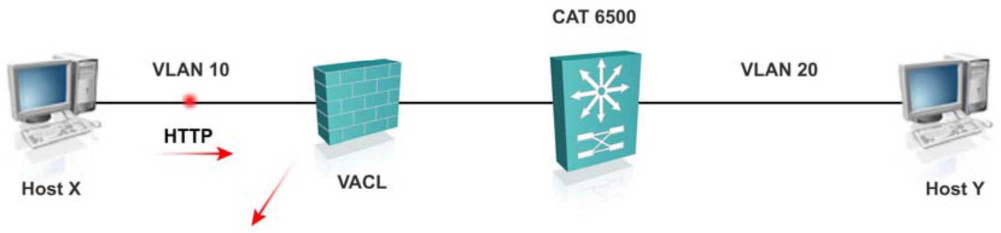
- **Can allow or restrict all traffic**
- **Can be applied to all packets**
- **Allow security packet filtering**
- **Allow traffic redirection**
- **Not defined by direction**
- **Configured on layer three addresses**

VACLs can allow or restrict *all* traffic. VACLs can be configured on the switch to apply to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VACLs allow security packet filtering and redirecting traffic to specific physical switch ports. Unlike IOS ACLs, VACLs are not defined by direction (input or output).

VACLs can be used to control Layer 3 addresses such as IP and IPX. All other protocols are access controlled through MAC addresses and Ethertype using MAC VACLs.

Using VACLs in Your Network

Often times there may be a need to restrict traffic between two hosts in the same VLAN (broadcast domain). Using VACLs gives us this ability within the Catalyst switch. Additionally, VACLs can be used to limit traffic between two hosts on different VLANs by using VACLs. This section discusses where VACLs might be appropriate.



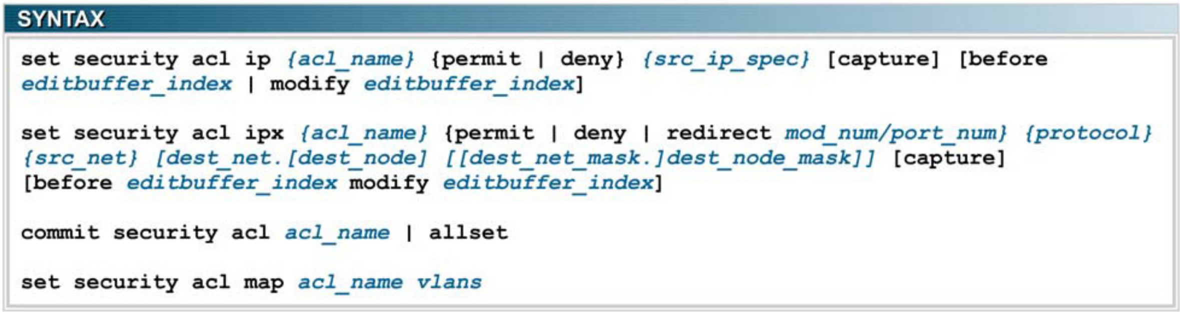
Wiring Closet Configuration

In a wiring closet configuration, Catalyst 6000 family switches might not be equipped with MSFCs (routers). In this situation, the switch can *still* support a VACL. Suppose Host X and Host Y are in different VLANs and are connected to CAT6500. Traffic from Host X to Host Y is eventually being routed by the switch equipped with the MSFC. Traffic from Host X to Host Y can be access controlled at the traffic entry point, CAT6500.

If HTTP traffic should not be switched from Host X to Host Y, a VACL can be configured on CAT6500. All HTTP traffic from Host X to Host Y would be dropped at CAT6500 and not be bridged.

Configuring VACLs

This section describes how to create and activate VACLs on the Catalyst 6000 family switches.



SYNTAX

```
set security acl ip {acl_name} {permit | deny} {src_ip_spec} [capture] [before
editbuffer_index | modify editbuffer_index]

set security acl ipx {acl_name} {permit | deny | redirect mod_num/port_num} {protocol}
{src_net} [dest_net.[dest_node] [[dest_net_mask.]dest_node_mask]] [capture]
[before editbuffer_index modify editbuffer_index]

commit security acl acl_name | allset

set security acl map acl_name vlans
```

VACL Configuration Summary

To create a VACL and map it to a VLAN, perform these steps:

Step 1 Enter the **`set security acl ip`** command to create a VACL and add ACEs.

Step 2 Enter the **`commit`** command to commit the VACL and its associated ACEs to NVRAM.

Step 3 Enter the **`set security acl map`** command to map the VACL to a VLAN.

VACLs have an implicit deny feature at the end of the list; a packet is denied if it does not match any VACL ACE.

This example shows how to create an IP VACL (we will name it ACL1) to allow traffic from source address 172.20.53.4:

```
Console> (enable) set security acl ip ACL1 permit host 172.20.53.4 0.0.0.0
ACL1 editbuffer modified. Use ‘commit’ command to apply changes.
Console> (enable)
```

Note Since VACLs have an implicit deny feature at the end of the list, in the above example, *all* other traffic would be denied.

This example shows how to add to ACL1 to allow traffic from all source addresses:

```
Console> (enable) set security acl ip ACL1 permit any
ACL1 editbuffer modified. Use ‘commit’ command to apply changes.
Console> (enable)
```

This example shows how to display the contents of the edit buffer:

```
Console> (enable) show security acl info ACL1 editbuffer
set security acl ip  ACL1
-----------------------------------------------------------------
1. permit ip host 172.20.53.4 any
2. permit ip any any
Console> (enable)
```

This example shows how to commit the VACL to NVRAM:

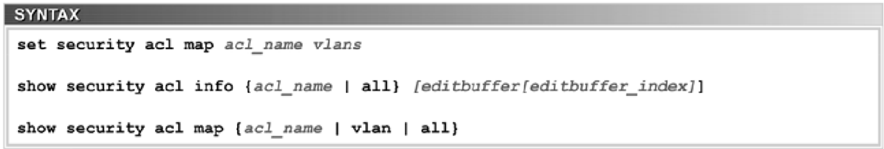
```
Console> (enable) commit security acl all
ACL commit in progress.
ACL ACL1 is committed to hardware.
Console> (enable)
```

Enter the `show security acl info ACL1` command to verify that the changes were committed. If this VACL has not been mapped to a VLAN, enter the `set security acl map` command to map it to a VLAN.

This example shows how to map the VACL to a VLAN:

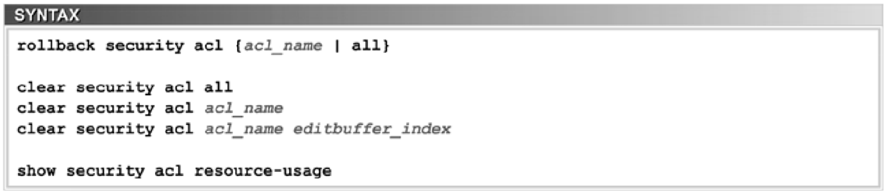
```
Console> (enable) set security acl map ACL1 10

ACL IPACL1 mapped to vlan 10

Console> (enable)
```

Showing VACL-to-VLAN Mapping



SYNTAX

```
set security acl map acl_name vlans

show security acl info {acl_name | all} [editbuffer[editbuffer_index]]

show security acl map {acl_name | vlan | all}
```



SYNTAX

```
rollback security acl {acl_name | all}

clear security acl all
clear security acl acl_name
clear security acl acl_name editbuffer_index

show security acl resource-usage
```

VACL-to-VLAN mapping can be displayed for a specified ACL or VLAN with the **`show security acl map`** command.

To show VACL-to-VLAN mapping, perform this task in privileged mode:

Table 6-8: Displaying VACL-to-VLAN mapping

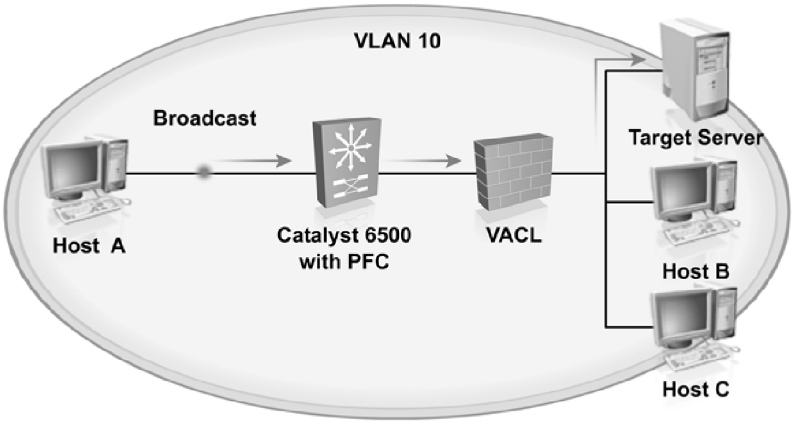
Task	Command
Show VACL-to-VLAN mapping.	**`show security acl map`** {*acl_name* | *vlan* | **all**}

This example shows how to show the mappings of a specific VACL:

```
Console> (enable) show security acl map ACL1
ACL IPACL1 is mapped to VLANs:
10
Console> (enable)
```

Redirecting Broadcast Traffic

Broadcast packets, by default, reach every host in a VLAN. With VACLs, these broadcast packets can be redirected to the intended application server port.



Using broadcast redirection, a broadcast packet from a host can be redirected to the target application server port, preventing other ports from receiving the packet. This reduces process time for the client, which does not need to see the broadcast.

To redirect broadcast traffic to a specific server port, perform this task in privileged mode:

Note In this example, TCP port 1234 is the intended server application port.

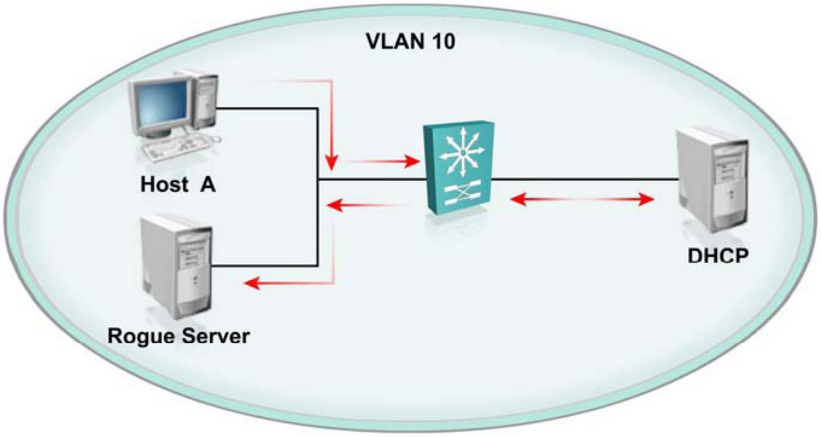
Table 6-9: Redirecting Broadcast Traffic

Task	Command (example)
Step 1. Redirect the broadcast packets.	`set security acl ip SERVER redirect 4/1 tcp any host 255.255.255.255 eq 1234`
Step 2. Permit all other traffic.	`set security acl ip SERVER permit ip any any`
Step 3. Commit the VACL.	`commit security acl SERVER`
Step 4. Map to VLAN 10.	`set security acl map SERVER 10`

Note The same concept can be applied to direct broadcast traffic to a multicast destination by redirecting the traffic to a group of ports.

Restricting DHCP Response

When DHCP requests are broadcasted, they reach every DHCP server in the VLAN. A rogue DHCP server would hear the request of a client and be able to respond, (with inappropriate IP address, DNS, etc). VACLs can limit the response so that only the *authorized* DHCP server can respond, and the switch will drop the other responses.



To restrict DHCP responses for a specific server, perform this task in privileged mode (the target DHCP server IP address is 1.2.3.4):

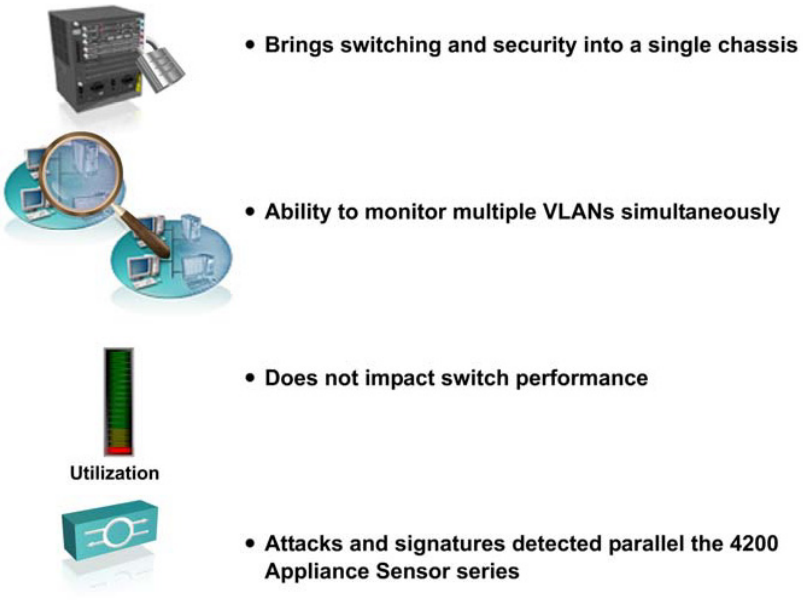
Table 6-10: Restricting DHCP Responses

Task	Command (example)
Step 1. Permit DHCP response from host 1.2.3.4.	`set security acl ip SERVER permit udp host 1.2.3.4 any eq 68`
Step 2. Deny DHCP responses from any other host, including the rouge DHCP server.	`set security acl ip SERVER deny udp any any eq 68`
Step 3. Permit other IP traffic.	`set security acl ip SERVER permit any any`
Step 4. Commit the VACL.	`commit security acl SERVER`
Step 5. Map the VACL to VLAN 10.	`set security acl map SERVER 10`

Only the target server returns a DHCP response from the DHCP request.

Forwarding Traffic for Intrusion Detection

The Intrusion Detection System Module (IDSM) is a module in the Catalyst 6000 family switch. It is part of the Cisco Secure Intrusion Detection System (Cisco Secure IDS). The IDSM performs network sensing—real-time monitoring of network packets through packet capture and analysis. But the problem is that the IDSM can only monitor traffic that is sees. In a switched network, only unicast, broadcasts and unknown layer 2 frames will arrive, by default, at the IDSM. This section will show how VACLs can be used to direct specific traffic to the IDSM for analysis.



- Brings switching and security into a single chassis
- Ability to monitor multiple VLANs simultaneously
- Does not impact switch performance
- Attacks and signatures detected parallel the 4200 Appliance Sensor series

The IDSM captures packets, and then reassembles and compares this data against signatures indicating typical intrusion activity. An effective way to forward the network traffic to the IDSM is to use the VLAN access control lists (VACLs). Port 1 on the IDSM is the monitoring interface for the IDSM. If the IDSM module is in slot 3 of the switch, we would need the VACL to copy the required data to port 3/1, so that the IDSM could analyze the data.

The four basic steps to direct the desired traffic to the IDSM port 1 are:

1. Create VACL to capture interesting traffic

2. Commit VACL to memory

3. Map the VACL to the desired VLAN(s)

4. Assign the IDSM monitoring port, (port 1), as the VACL capture port

Following are the examples for each of these steps.

Example 1:

```
Switch>(enable) set security acl ip WEBONLY permit tcp any host 172.30.1.50 eq
80 capture
Switch>(enable) set security acl ip WEBONLY permit ip any any
```

The new option here is the word “capture” at the end of the first command. This word can be interpreted as meaning “copy”.

This sets the WEBONLY VACL to capture all **HTTP** traffic for IDS analysis.

Note The second command line is *required* for other traffic to flow through the VLAN, even though we are not capturing it for the IDSM.

Example 2:

```
Switch>(enable) commit security acl WEBONLY
```

This commits the VACL to memory.

Example 3:

```
Switch>(enable) set security acl map WEBONLY 10
```

This maps the VACL to VLAN 10.

Example 4:

```
Switch>(enable) set security acl capture-ports 3/1
```

This defines what port (the IDSM is in slot 3; the monitoring interface is always port 1) the captured data is forwarded to. This command takes the packets that were captured or “copied” from example 1, and directs this data to the monitoring interface of the IDSM.

Lesson Summary

This lesson accomplished the following:

- Described how a VACL operates
- Explained how to configure a VACL
- Described situations where a VACL can be effective

Next Steps

After completing this lesson, go to:

- QoS on the 6000 series Catalyst switch

References

For additional information, refer to these resources:

- For complete syntax and usage information for the commands used in this chapter, refer to the *Catalyst 6000 Family Command Reference* publication

summary

- Verified the function of an IP Permit List
- Demonstrated and configure an IP Permit List on a 6500 series Catalyst switch
- Demonstrated how to enable and disable an IP Permit List
- Defined Layer 3 protocol Filtering
- Configured Layer 3 Protocol Filtering
- Described how a VACL operates
- Demonstrated how to configure a VACL
- Understood situations where a VACL could be effective

Case Study: The Tech Expo Floor

Many technical conferences or shows now allow access to wireless Internet while at the show. One of the concerns from the planners is DHCP. The concern is that one of the laptop devices, with a wireless network adapter, could run a DHCP server, and hand out incorrect DHCP address and/or DHCP options such as WINS, Default Gateway, DNS and other information.

Is it possible that an attendee could become a DHCP server for other attendees, either on accident or on purpose?

What threats are posed by an unauthorized DHCP server roaming on the wireless network?

What could be done to prevent the DHCP server, running on the attendee's laptop, from successfully responding to DHCP? (Remember the client is wireless, and not directly connected to the Catalyst switch.)

Quality of Service

Overview

This module details Quality of Service (QoS) features supported on the Catalyst 6500 Series switch.

Outline

The module contains these lessons:

- The Need for QoS
- Layer 2 Marking
- Layer 3 Marking
- Layer 2 to Layer 3 Remarking
- Congestion Management
- Congestion Avoidance
- Policing

The Need for QoS

Overview

With the convergence of voice, video, and data networks, Quality of Service (QoS) becomes a significant design issue. This lesson discusses quality issues that can arise in converged networks and lists the categories of Cisco's QOS tools.

Importance

An understanding of available QoS tools is critical to servicing latency-sensitive applications on the Catalyst 6500.

Objectives

Upon completing this lesson, you will be able to:

- List problems addressed by QoS
- Describe Cisco's QoS tools

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Remote Access Networks (BCRAN) course or have equivalent knowledge

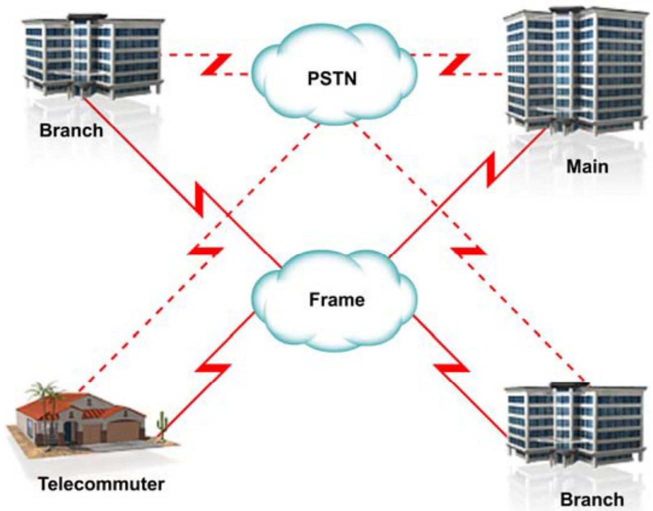
Outline

This lesson includes these sections:

- Overview
- Networks Before and After Convergence
- Cisco's QoS Tools
- Summary

Networks Before and After Convergence

This section discusses the quality problems that can arise when voice, video, and data networks are consolidated in a single network.



Traditional data traffic characteristics:

- Bursty data flow
- First-come, first-served access
- Data rate adaptive to network conditions
- Brief outages are survivable

Before Convergence:

Traditionally, voice, data, and video networks have been separate. Since the latency-sensitive voice and video traffic was separate from the data traffic, the bursty nature of the data traffic was not an issue.

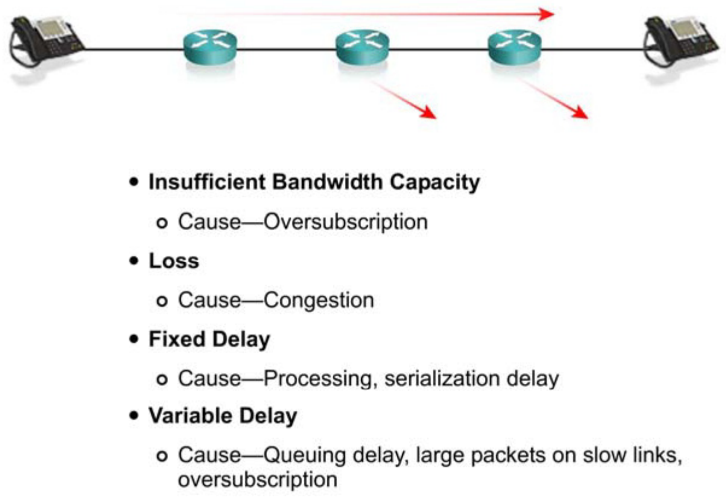
After Convergence:

To leverage the investment in the data WAN, companies are migrating to a converged network, where voice, video, and data share the WAN bandwidth. While the Public Switched Telephone Network (PSTN) is still present in the network, for backup and for handling WAN overflow, a converged network design gives the company "toll-bypass," which can recover the additional WAN costs.

The Challenge:

Since multiple applications now share a common network, latency-sensitive applications need preferential treatment. Providing this preferential treatment is the goal of Cisco's QoS tools.

Quality Issues



- **Insufficient Bandwidth Capacity**
  - Cause—Oversubscription
- **Loss**
  - Cause—Congestion
- **Fixed Delay**
  - Cause—Processing, serialization delay
- **Variable Delay**
  - Cause—Queuing delay, large packets on slow links, oversubscription

Insufficient Bandwidth Capacity: When services (e.g., voice and video) are added to an existing network, there is a potential for oversubscribing the existing bandwidth.

Loss: If a buffer fills to capacity, additional packets will be discarded. This is called a "Tail Drop."

Fixed Delay: There are two categories of delay that need to be factored into a multiservice design, fixed delay and variable delay. Examples of fixed delay include processing delay and serialization delay.

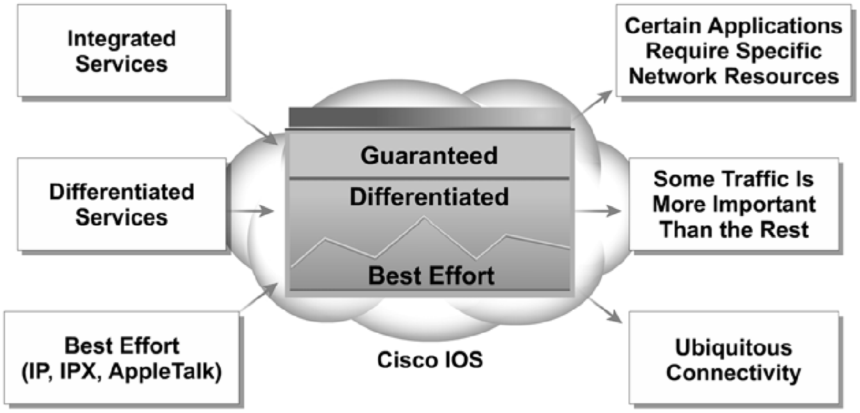
- **Processing Delay:** A switch or router's processor speed and/or caching method affect how quickly a frame or packet is forwarded out of an interface.

- **Serialization Delay:** The slower a serial interface, the longer it takes to completely forward a packet.

Variable Delay: Some components of delay can be manipulated. For example, queuing can expedite the forwarding of particular packets, and large packets can be fragmented to minimize serialization delay.

Cisco’s QoS Tools

This section details the three categories of Cisco’s QoS tools.


Integrated Services
Differentiated Services
Best Effort (IP, IPX, AppleTalk)
Guaranteed
Differentiated
Best Effort
Cisco IOS
Certain Applications Require Specific Network Resources
Some Traffic Is More Important Than the Rest
Ubiquitous Connectivity

By default, on high-speed circuits (> 2 Mbps), Cisco handles traffic on a first come, first serve basis. This type of treatment is referred to as “**Best Effort**” or FIFO (First In First Out).

When multiple traffic types are present on the network, we may need to treat specific traffic flows with higher priority. This requires differentiating (i.e., classifying) the various traffic flows. This is referred to as “**Differentiated Services**.” Generally, the various traffic types are marked at Layer 2 and/or at Layer 3. Once the traffic is marked, congestion avoidance tools or congestion management tools are applied to the traffic.

Congestion management seeks to queue traffic based on the traffic’s priority. Following are the congestion management tools that will be addressed in this module:

- Class-Based Weighted Fair Queuing
- Low Latency Queuing

Congestion avoidance prevents an output buffer from filling to capacity by discarding packets before they enter the queue. The following congestion avoidance tool will be addressed in this module:

- Weighted Random Early Detection

Some applications, such as Voice over IP phone calls, can reserve an amount of bandwidth for the duration of the application. This reserved bandwidth cannot be used by other applications. This type of treatment is called “**Integrated Services**.” The following integrated services tool will be addressed in this module:

- RSVP

In some cases, the amount of traffic entering or exiting an interface may need to be limited. This is called “**policing**.” The following policing tools will be addressed in this module:

- QoS ACLs
- Committed Access Rate (CAR)

Lesson Summary

This lesson accomplished the following:

- Listed problems addressed by QoS
- Described Cisco's QoS tools

Next Steps

After completing this lesson, go to:

- Layer 2 Marking

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sft_6_1/configgd/qos.htm

Layer 2 Marking

Overview

Traffic flowing through the Catalyst 6500 can be marked with a priority. At Layer 2, this priority is called Class of Service (CoS). This lesson will detail the options and the syntax for Layer 2 frame marking.

Importance

To provide preferential treatment to latency-sensitive traffic in a switched infrastructure, Layer 2 marking is a fundamental tool.

Objectives

Upon completing this lesson, you will be able to:

- List the options for Layer 2 frame marking
- Configure a Catalyst 6500 port to mark traffic with a CoS value

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Multilayer Switched Networks (BCMSN) course or have equivalent knowledge

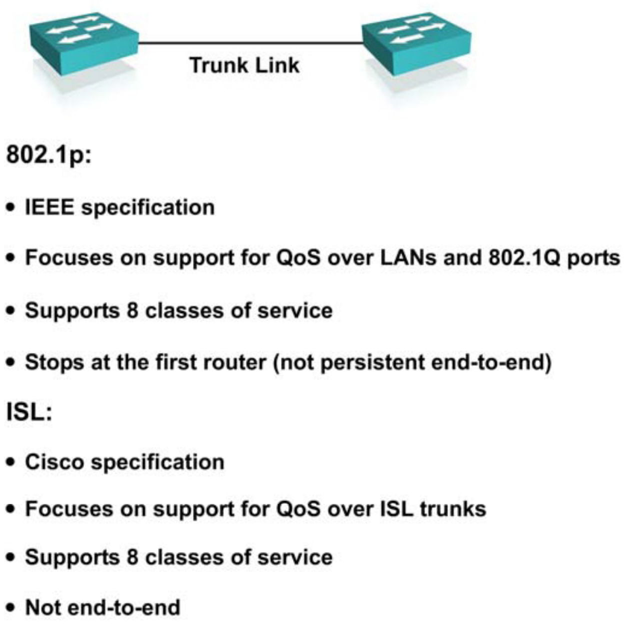
Outline

This lesson includes these sections:

- Overview
- Layer 2 Class of Service
- Summary

Layer 2 Class of Service

This section identifies the options for Layer 2 CoS marking.



802.1p:

- IEEE specification
- Focuses on support for QoS over LANs and 802.1Q ports
- Supports 8 classes of service
- Stops at the first router (not persistent end-to-end)

ISL:

- Cisco specification
- Focuses on support for QoS over ISL trunks
- Supports 8 classes of service
- Not end-to-end

Frames can be marked at Layer 2 as they traverse a trunk link. The Catalyst 6500 supports Layer 2 marking on both ISL and IEEE 802.1q trunks. Although the trunk frames are marked differently, they both can encode a 3-bit field, with 8 possible values, as a Class of Service (CoS) value.

The IEEE 802.1Q standard has a 802.1p specification that designates 3-bits out of a frame's tag control bytes that can be used for marking a frame's priority. Cisco's ISL trunk format specifies 3 bits in the header to be used for priority information.

Catalyst 6500 Port Trust States

- **untrusted** (default)
- **trust-ipprec** (Layer 3 Switching Engine only)
- **trust-dscp** (Layer 3 Switching Engine only)
- **trust-cos**

The trust state of an Ethernet port determines how it marks, schedules, and classifies received traffic, and whether or not congestion avoidance is implemented. You can configure the trust state of each port with one of these keywords:

- `untrusted` (default)
- `trust-ipprec` (Layer 3 Switching Engine only)
- `trust-dscp` (Layer 3 Switching Engine only)
- **trust-cos**

Marking at Untrusted Ports

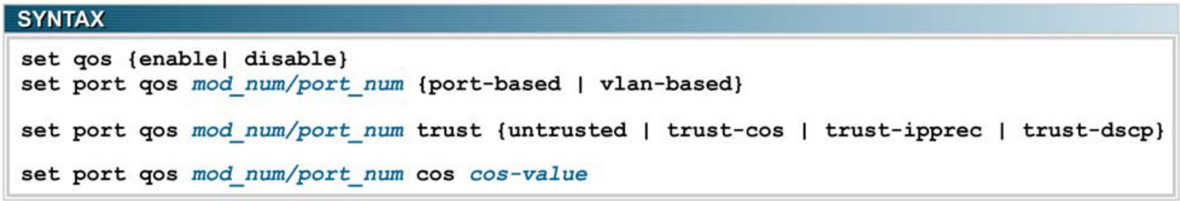
With either a Layer 2 Switching Engine or a Layer 3 Switching Engine, QoS marks all frames received through untrusted ports with the port CoS value (the default is zero). QoS does not implement ingress port congestion avoidance on untrusted ports; the traffic goes directly to the Switching Engine.

Marking at Trusted Ports

When an ISL frame enters the switch through a trusted port, QoS accepts the three least significant bits in the User field as a CoS value. When an 802.1Q frame enters the switch through a trusted port, QoS accepts the User Priority bits as a CoS value. QoS marks all traffic received in other frame types with the port CoS value.

Configuring Layer 2 QoS

This section demonstrates how to enable Layer 2 QoS on the Catalyst 6500.



SYNTAX

```
set qos {enable| disable}
set port qos mod_num/port_num {port-based | vlan-based}

set port qos mod_num/port_num trust {untrusted | trust-cos | trust-ipprec | trust-dscp}

set port qos mod_num/port_num cos cos-value
```

Enabling QoS

To enable QoS on the Catalyst 6500, perform this task in privileged mode:

Table 7-1: Enabling QoS

Task	Command
Enable QoS on the switch.	**set qos {enable | disable}**

This example shows how to enable QoS:

```
Console> (enable) set qos enable
QoS is enabled.
Console> (enable)
```

Enabling Port-Based or VLAN-Based QoS

Note	QoS only supports the commands in this section with an MSFC.

By default, QoS uses ACLs attached to ports. On a per-port basis, you can configure QoS to use ACLs attached to a VLAN. To enable VLAN-based QoS on a port, perform this task in privileged mode:

Table 7-2: Configuring Port QoS

Task	Command
Step 1 Enable VLAN-based QoS on a port.	**set port qos** *mod_num/port*_**num {port-based | vlan-based}**
Step 2 Verify the configuration.	**show port qos** *mod_num/port_num*

This example shows how to enable VLAN-based QoS on a port:

```
Console> (enable) set port qos 1/1-2 vlan-based
Hardware programming in progress...
QoS interface is set to vlan-based for ports 1/1-2.
Console> (enable)
```

Note Changing a port from port-based to VLAN-based QoS detaches all ACLs from the port. Any ACLs attached to the VLAN apply to the port immediately.

Configuring the Trust State of a Port

This command configures the trust state of a port. By default, all ports are untrusted.

To configure the trust state of a port, perform this task in privileged mode:

Table 7-3: Configuring a Port's Trust State

Task	Command
Step 1 Configure the trust state of a port.	**set port qos** *mod_num/port_num* **trust {untrusted | trust-cos | trust-ipprec | trust-dscp}**
Step 2 Verify the configuration.	**show port qos** *mod_num/port_num*

QoS only supports the **trust-ipprec** and **trust-dscp** keywords with an MSFC.

This example shows how to configure port 1/1 with the **trust-cos** keyword:

```
Console> (enable) set port qos 1/1 trust trust-cos
Port 1/1 qos set to trust-cos
Console> (enable)
```

Note Only ISL or 802.1Q frames carry CoS values. Configure ports with the **trust-cos** keyword only when the received traffic is ISL or 802.1Q frames carrying CoS values that you know to be consistent with network policy.

Configuring the CoS Value for a Port

Unmarked frames from ports configured as trusted, and all frames from ports configured as untrusted, are assigned the CoS value specified with this command.

To configure the CoS value for a port, perform this task in privileged mode:

Table 7-4: Assigning a CoS Value to a Port

Task	Command
Step 1 Configure the CoS value for a port.	**set port qos** *mod_num/port_num* **cos** *cos-value*
Step 2 Verify the configuration.	**show port qos** *mod_num/port_num*

This example shows how to configure the port CoS value to 3 for port 1/1:

```
Console> (enable) set port qos 1/1 cos 3
Port 1/1 qos cos set to 3
Console> (enable)
```

To revert to the default CoS value for a port, perform this task in privileged mode:

Table 7-5: Clearing a CoS Value from a Port

Task	Command
Step 1 Revert to the default CoS value for a port.	**clear port qos** *mod_num/port_num* **cos**
Step 2 Verify the configuration.	**show port qos** *mod_num/port_num*

This example shows how to revert to the default CoS value for port 1/1:

```
Console> (enable) clear port qos 1/1 cos
Port 1/1 qos cos setting cleared.
Console> (enable)
```

Lesson Summary

This lesson accomplished the following:

- Listed the options for Layer 2 frame marking
- Demonstrated how to mark traffic on a Catalyst 6500 port with a CoS value

Next Steps

After completing this lesson, go to:

- Layer 3 Marking

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sft_6_1/configgd/qos.htm

Layer 3 Marking

Overview

In addition to CoS markings, traffic can also have prioritization markings applied at Layer 3. This lesson will explain the two approaches for Layer 3 marking, in addition to demonstrating a modular approach for creating policies, which can mark traffic.

Importance

As latency-sensitive traffic travels between subnets, Layer 2 markings are removed, while Layer 3 markings remain intact. Therefore, the ability to mark traffic at Layer 3 is essential to preserving priority for routed traffic.

Objectives

Upon completing this lesson, you will be able to:

- List the options for Layer 3 packet marking
- Describe the Modular QoS Command Line Interface
- Use MQC on an MSFC to mark traffic with an IP Precedence, DSCP value, or CoS value

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- A basic understanding of the IP Version 4 frame format.

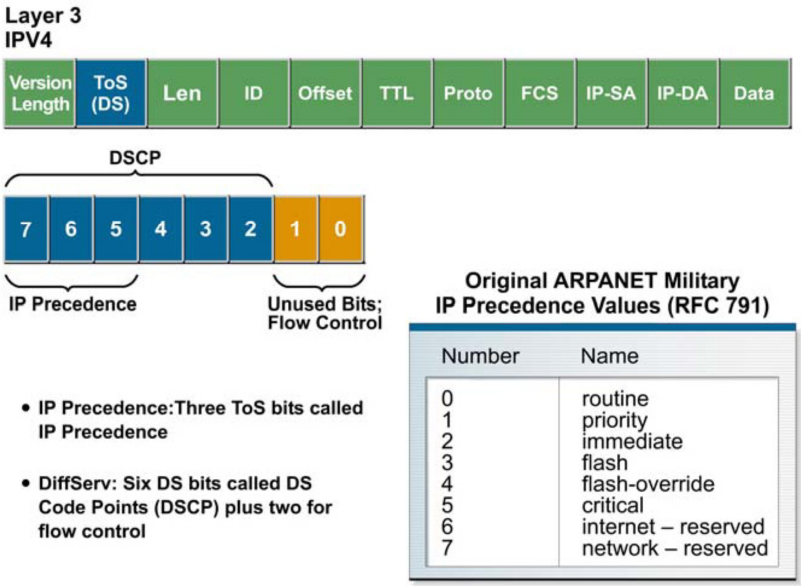
Outline

This lesson includes these sections:

- Overview
- Layer 3 Type of Service
- Marking at Layer 3 Using MQC
- Summary

Layer 3 Type of Service

This section identifies the options for Layer 3 ToS marking.



- IP Precedence:Three ToS bits called IP Precedence
- DiffServ: Six DS bits called DS Code Points (DSCP) plus two for flow control

Original ARPANET Military IP Precedence Values (RFC 791)

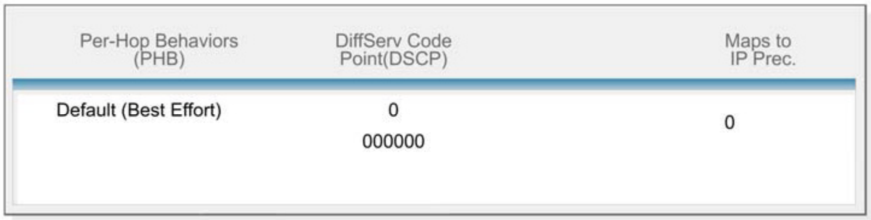
Number	Name
0	routine
1	priority
2	immediate
3	flash
4	flash-override
5	critical
6	internet – reserved
7	network – reserved

Inside an IP Version 4 header, there are 8 bits called the Type of Service (ToS) Byte. Bits within this field can be used to provide Layer 3 priority markings for packets. The two options for Layer 3 marking are IP Precedence and DSCP.

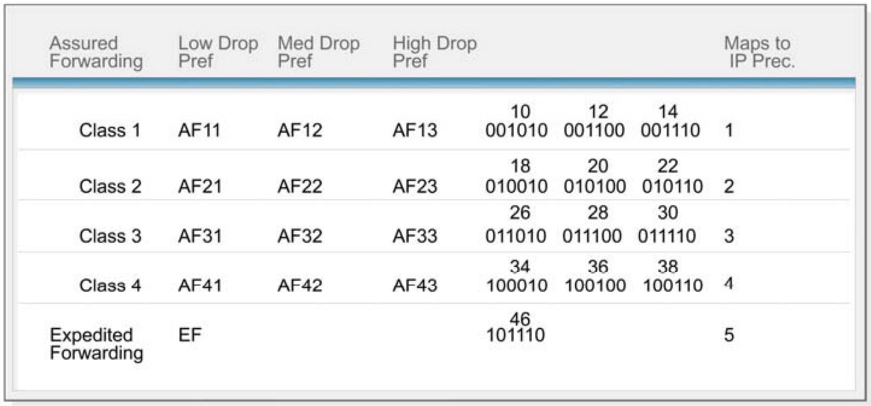
IP Precedence uses the 3 leftmost bits in the ToS Byte. These bits provide eight possible IP Precedence values (0 – 7). However, the network reserves values 6 and 7. So, IP Precedence has five usable levels of priority.

DSCP (Differentiated Services Code Point) uses the 6 leftmost bits in the ToS Byte. Therefore, there are 64 distinct levels of priority that can be set using DSCP. IP Precedence and DSCP are compatible. If a switch or router is expecting a DSCP value, and receives an IP Precedence value, it will use the IP Precedence value as the first three bits of the DSCP value, and the last three DSCP bits will be zero.

DiffServ Per Hop Behaviors



Per-Hop Behaviors (PHB)	DiffServ Code Point(DSCP)	Maps to IP Prec.
Default (Best Effort)	0 000000	0



Assured Forwarding	Low Drop Pref	Med Drop Pref	High Drop Pref				Maps to IP Prec.
Class 1	AF11	AF12	AF13	10 001010	12 001100	14 001110	1
Class 2	AF21	AF22	AF23	18 010010	20 010100	22 010110	2
Class 3	AF31	AF32	AF33	26 011010	28 011100	30 011110	3
Class 4	AF41	AF42	AF43	34 100010	36 100100	38 100110	4
Expedited Forwarding	EF			46 101110			5

Even though DSCP has 64 levels of granularity, Cisco recommends that only a subset of those values be used. The IETF has defined four models that specify how traffic marked with particular DSCP values behaves as they leave a router (i.e., a hop). These models are called Per Hop Behaviors (PHBs). When configuring DSCP values later in this module either the numerical value or the PHB name can be used.

Default

The Default PHB is referred to as "best-effort," because the DSCP value is set to zero (i.e., 000000 in binary).

Class Selector

The Class Selector PHB is meant to map directly to IP Precedence values, which only use the three leftmost bits in the ToS Byte. The Class Selector (CS) PHB accomplishes this by setting the last three DSCP bits to zero.

Assured Forwarding

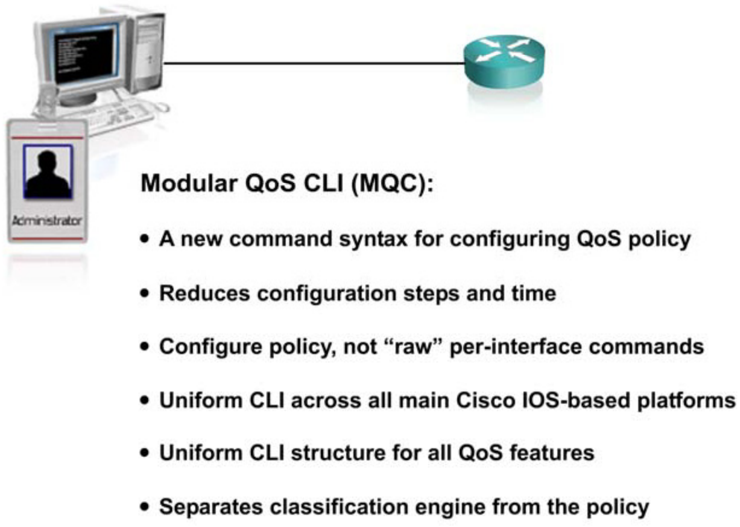
The Assured Forwarding (AF) PHB maps to the IP Precedence values of 1 – 4. Additionally, within each of those IP Precedence values, there are three AF values, which determine the drop probability of a packet. This drop preference will be used by Weighted Random Early Detection (WRED), discussed later in this module, to determine which packets should be discarded when the output queue begins to fill.

Expedited Forwarding

The Expedited Forwarding (EF) PHB is meant for traffic that needs very high priority treatment in the network. In binary, the EF DSCP value is 101110, which is a decimal value of 46. Notice that if we just consider the three leftmost bits, we have a corresponding IP Precedence value of 5, which is the highest recommended value we can assign to our traffic.

Marking at Layer 3 Using MQC

This section introduces a modular approach to Layer 3 QoS marking.



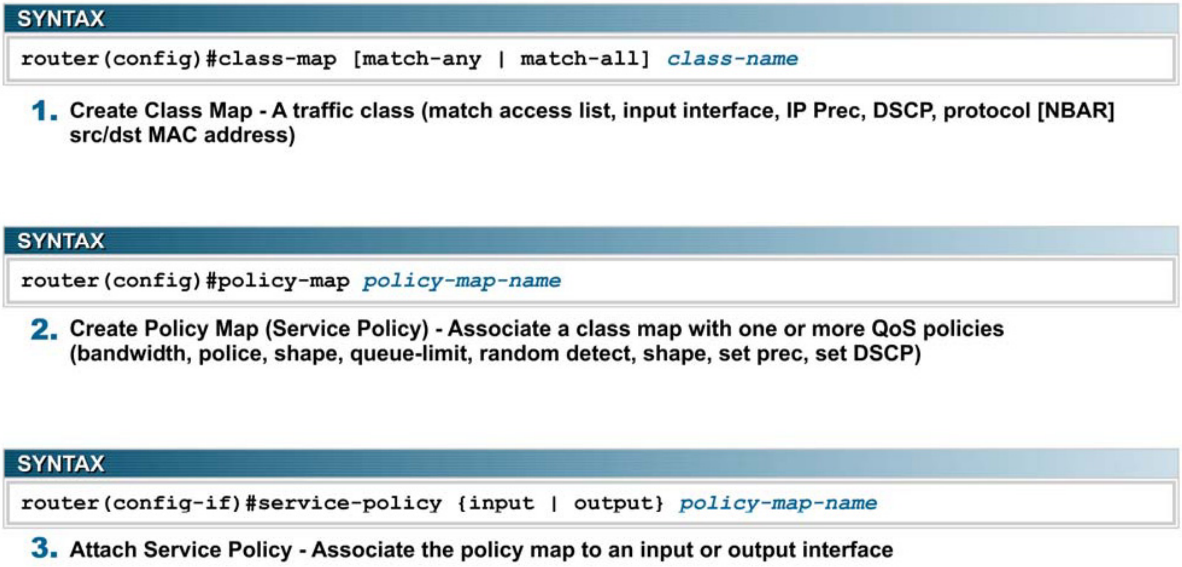
Modular QoS CLI (MQC):

- A new command syntax for configuring QoS policy
- Reduces configuration steps and time
- Configure policy, not “raw” per-interface commands
- Uniform CLI across all main Cisco IOS-based platforms
- Uniform CLI structure for all QoS features
- Separates classification engine from the policy

The Modular Quality of Service Command Line Interface (MQC) is a modular and efficient way to apply policy to an MSFC’s VLAN interfaces. MQC classifies traffic using CLASS-MAPs. A POLICY-MAP then specifies characteristics for each CLASS-MAP. Once this policy has been created, it can be applied to multiple VLAN interfaces, thus making it much more efficient to apply the same policy multiple times.

Note	MQC is a tool that is applied to the MSFC. It does not apply to the Cat OS.

Basic MQC Commands



SYNTAX

```
router(config)#class-map [match-any | match-all] class-name
```

1. **Create Class Map - A traffic class (match access list, input interface, IP Prec, DSCP, protocol [NBAR] src/dst MAC address)**

SYNTAX

```
router(config)#policy-map policy-map-name
```

2. **Create Policy Map (Service Policy) - Associate a class map with one or more QoS policies (bandwidth, police, shape, queue-limit, random detect, shape, set prec, set DSCP)**

SYNTAX

```
router(config-if)#service-policy {input | output} policy-map-name
```

3. **Attach Service Policy - Associate the policy map to an input or output interface**

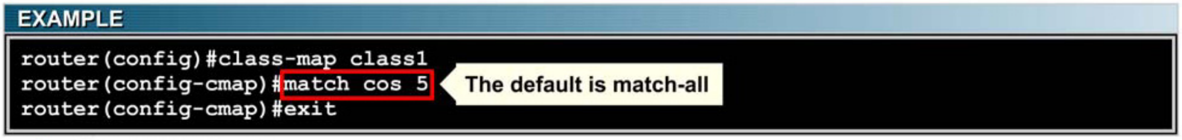
MQC involves three basic steps:

Step 1 A CLASS-MAP is configured, and as will be seen later, a variety of MATCH criteria can be used to identify what traffic should be placed in what class.

Step 2 Once multiple classes have been defined, a POLICY-MAP is created to set characteristics for the traffic in each class. For example, a POLICY-MAP can reserve an amount of bandwidth for each class of traffic. Also, the POLICY-MAP can be used to mark classified traffic at Layer 2 (i.e., CoS) or at Layer 3 (e.g., IP Precedence or DSCP).

Step 3 Once the policy has been created, it can be applied to a VLAN interface, using the **`service-policy`** command.

MQC Classification Example

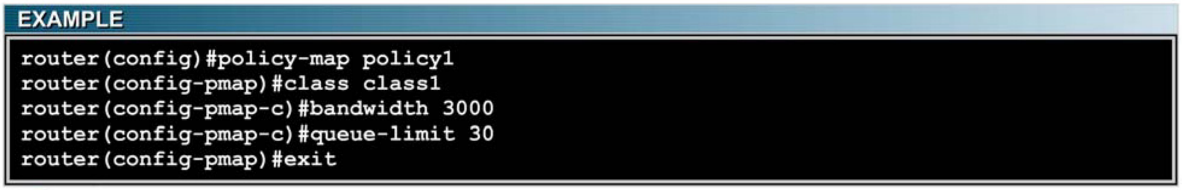


EXAMPLE

```
router(config)#class-map class1
router(config-cmap)#match cos 5
router(config-cmap)#exit
```

The default is match-all

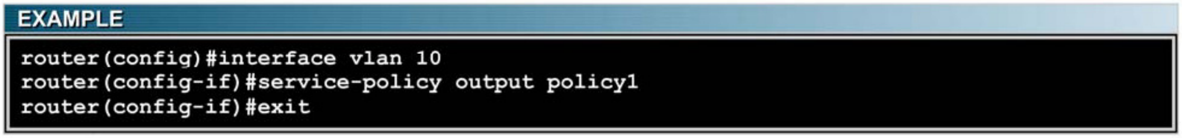
1. Create Class Map



EXAMPLE

```
router(config)#policy-map policy1
router(config-pmap)#class class1
router(config-pmap-c)#bandwidth 3000
router(config-pmap-c)#queue-limit 30
router(config-pmap)#exit
```

2. Create Policy Map



EXAMPLE

```
router(config)#interface vlan 10
router(config-if)#service-policy output policy1
router(config-if)#exit
```

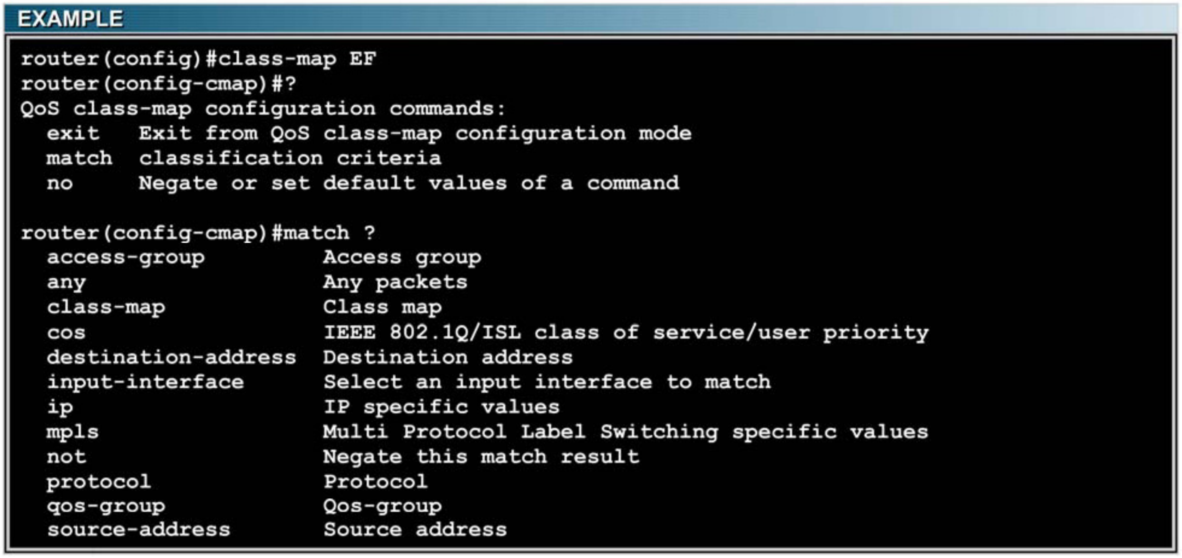
3. Attach Service Policy

Step 1 In the above example, traffic that has been marked at Layer 2 with a CoS of 5 is put into a class named “class1.”

Step 2 A policy named “policy1” is created that reserves 3 Mbps of bandwidth for traffic in this class, and the queue that holds traffic for this class is limited to 30 packets.

Step 3 The “policy1” policy is then applied outbound to the VLAN 10 interface.

Classification Using MQC



```
router(config)#class-map EF
router(config-cmap)#?
QoS class-map configuration commands:
  exit   Exit from QoS class-map configuration mode
  match  classification criteria
  no     Negate or set default values of a command

router(config-cmap)#match ?
  access-group         Access group
  any                  Any packets
  class-map            Class map
  cos                  IEEE 802.1Q/ISL class of service/user priority
  destination-address  Destination address
  input-interface      Select an input interface to match
  ip                   IP specific values
  mpls                 Multi Protocol Label Switching specific values
  not                  Negate this match result
  protocol             Protocol
  qos-group            Qos-group
  source-address       Source address
```

When creating a CLASS-MAP, multiple match conditions may be specified. Following are some of the traffic characteristics that can be matched:

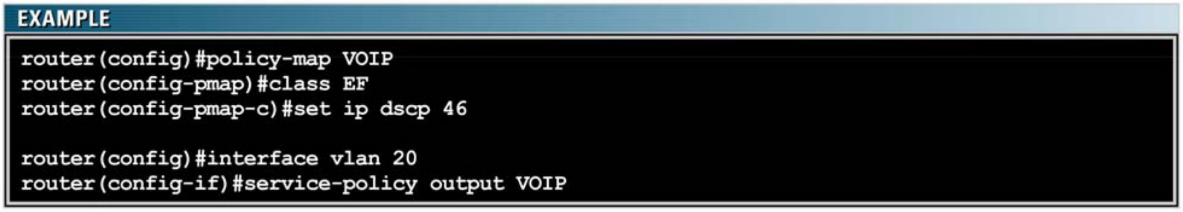
`match access-group` ***`access-list-number:`*** Traffic that is specified by a numbered access-list is matched by the CLASS-MAP.

`match cos` ***`cos_value`*****`:`** Traffic that has been marked at Layer 2 with a particular CoS value is matched by the CLASS-MAP. Up to four CoS values may be specified in a single match statement.

`match ip precedence number:` Traffic that has been marked at Layer 3 with a particular IP Precedence value is matched by the CLASS-MAP. Up to eight IP Precedence values may be specified in a single match statement.

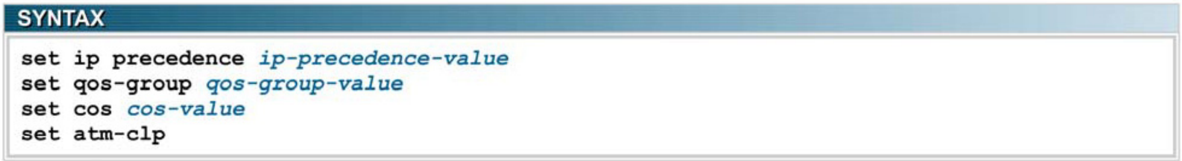
`match ip dscp number:` Traffic that has been marked at Layer 3 with a particular DSCP value is matched by the CLASS-MAP. Up to eight DSCP values may be specified in a single match statement.

Class-Based Marking Example



EXAMPLE

```
router(config)#policy-map VOIP
router(config-pmap)#class EF
router(config-pmap-c)#set ip dscp 46

router(config)#interface vlan 20
router(config-if)#service-policy output VOIP
```



SYNTAX

```
set ip precedence ip-precedence-value
set qos-group qos-group-value
set cos cos-value
set atm-clp
```

In this example, traffic that has been matched by a class named “EF” is marked with a DSCP value of 46. The policy named “VOIP” is then applied as an output policy on interface VLAN 20. Therefore, traffic belonging to the “EF” class leaves the VLAN 20 interface with a DSCP marking of 46.

A POLICY-MAP can also be used to mark traffic with an IP Precedence value, a locally-significant QoS Group value, a CoS value, or it can mark an ATM cell’s Cell Loss Priority(CLP) bit.

Lesson Summary

This lesson accomplished the following:

- Listed the options for Layer 3 packet marking
- Described the Modular QoS Command Line Interface
- Used MQC on an MSFC to mark traffic with an IP Precedence, DSCP value, or CoS value

Next Steps

After completing this lesson, go to:

- Layer 2 to Layer 3 Remarking

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mqc/mcli.htm

Layer 2 to Layer 3 Remarking

Overview

When traffic passes through a router, Layer 2 markings (i.e., CoS values) are removed, however Layer 3 markings (i.e., ToS values) remain. This lesson will demonstrate how to assign Layer 3 ToS values based on Layer 2 CoS values.

Importance

Traffic marked with Layer 2 CoS values must be remarked with Layer 3 ToS values to preserve the priority marking as the traffic passes through a router, or in the case of a Catalyst 6500, an MSFC.

Objectives

Upon completing this lesson, you will be able to:

- Define the limitation of passing CoS information through the MSFC
- Remark packets with CoS markings with ToS markings

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- A basic understanding of CoS and ToS markings as presented in the two previous lessons

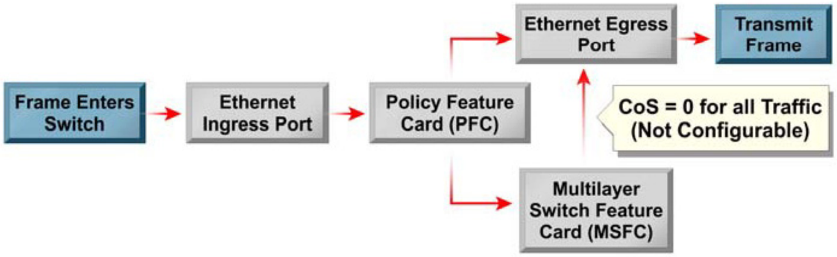
Outline

This lesson includes these sections:

- Overview
- CoS Limitation
- Mapping CoS Values to DSCP Values
- Summary

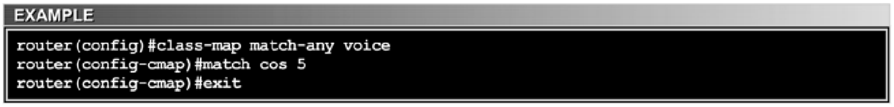
CoS Limitation

This section identifies the need for Layer 2 to Layer 3 remarking and demonstrates this remarking using the MQC approach.


Frame Enters Switch
Ethernet Ingress Port
Policy Feature Card (PFC)
Ethernet Egress Port
Transmit Frame
Multilayer Switch Feature Card (MSFC)
CoS = 0 for all Traffic (Not Configurable)

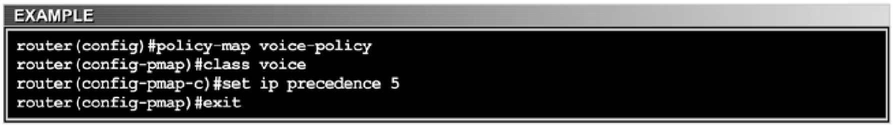
Just as the Layer 2 information on a frame is stripped and rewritten as a packet passes through a router, a packet with Layer 2 (i.e., CoS) markings will have those markings stripped as it passes through the MSFC. For example, a packet enters a port on the Catalyst 6500 on VLAN 10 with a CoS value of 5. The packet is then routed to VLAN 20. As the Layer 2 information is rewritten the CoS value is removed, and the packet exits the Catalyst 6500 with a CoS value of 0. To preserve CoS priority as a packet passes through the MSFC, the CoS values must be rewritten (i.e., remarked) to ToS values (i.e., IP Precedence or DSCP).

Remark Layer 2 to Layer 3 Example



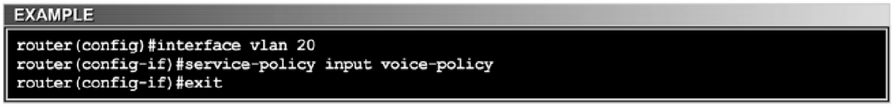
```
router(config)#class-map match-any voice
router(config-cmap)#match cos 5
router(config-cmap)#exit
```

1. Class Map - Match CoS value (Layer 2)



```
router(config)#policy-map voice-policy
router(config-pmap)#class voice
router(config-pmap-c)#set ip precedence 5
router(config-pmap)#exit
```

2. Policy Map - Set matched traffic to IP Precedence (Layer 3)

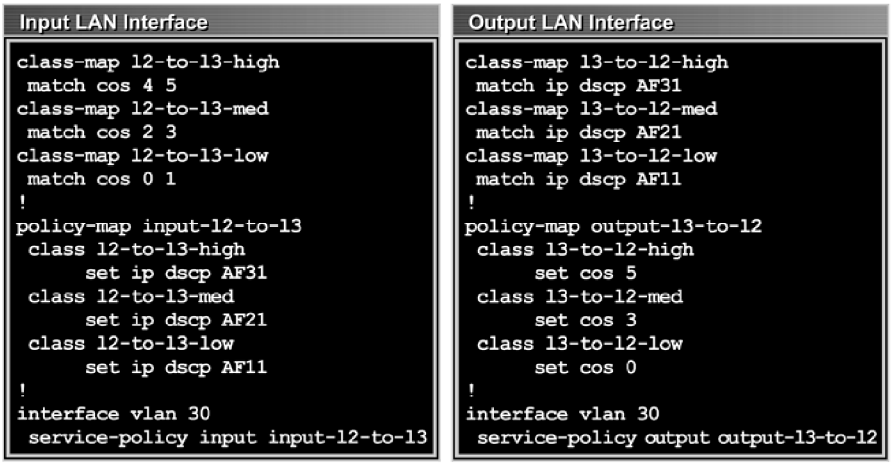


```
router(config)#interface vlan 20
router(config-if)#service-policy input voice-policy
router(config-if)#exit
```

3. Service Map - Associate service policy with an interface

In this example, a CLASS-MAP named "voice" is matching all traffic that has a CoS value of 5. A POLICY-MAP named "voice-policy" is marking all traffic in the "voice" class with an IP Precedence value of 5. The "voice-policy" policy is then applied inbound on interface VLAN 20. Therefore, as a packet with a CoS marking of 5 enters the Catalyst 6500, the input VLAN 20 interface on the MSFC marks that packet with an IP Precedence value of 5. Since the IP Precedence is part of the IP Version 4 header, it is not stripped as the packet passes through the MSFC. This procedure preserves the relative preference level of packets that only had a Layer 2 CoS marking.

Bi-directional Layer 2 and Layer 3 Remarking Example

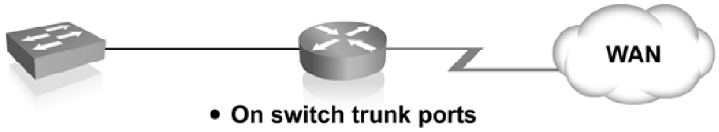


Input LAN Interface

```
class-map l2-to-l3-high
 match cos 4 5
class-map l2-to-l3-med
 match cos 2 3
class-map l2-to-l3-low
 match cos 0 1
!
policy-map input-l2-to-l3
 class l2-to-l3-high
      set ip dscp AF31
 class l2-to-l3-med
      set ip dscp AF21
 class l2-to-l3-low
      set ip dscp AF11
!
interface vlan 30
 service-policy input input-l2-to-l3
```

Output LAN Interface

```
class-map l3-to-l2-high
 match ip dscp AF31
class-map l3-to-l2-med
 match ip dscp AF21
class-map l3-to-l2-low
 match ip dscp AF11
!
policy-map output-l3-to-l2
 class l3-to-l2-high
      set cos 5
 class l3-to-l2-med
      set cos 3
 class l3-to-l2-low
      set cos 0
!
interface vlan 30
 service-policy output output-l3-to-l2
```



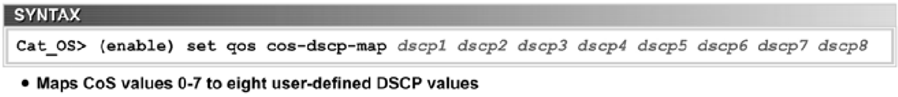
- **On switch trunk ports**
- **802.1p or ISL Interfaces**

Not only do we want to convert Layer 2 markings to Layer 3 markings, as in the previous example, we also may want to convert Layer 3 markings to Layer 2 markings. For example, if traffic coming in from the WAN has been marked with DSCP values, we may want to mark those packets with CoS markings that can be interpreted by our Layer 2 switched infrastructure.

Therefore, the above example demonstrates how to mark both inbound and outbound traffic on the VLAN 30 interface. In this example, input traffic (i.e., from the LAN) is assigned a DSCP value, and output traffic (i.e., to the LAN) is assigned a CoS value.

Mapping CoS Values to DSCP Values

This section demonstrates how to map Layer 2 CoS values to Layer 3 ToS values via the Cat OS.



SYNTAX

```
Cat_OS> (enable) set qos cos-dscp-map dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dscp8
```

- Maps CoS values 0-7 to eight user-defined DSCP values

In addition to the MQC approach of remarking Layer 2 CoS values to Layer 3 DSCP values, the Cat OS can be used as well. To map received CoS values to the DSCP value that QoS uses internally on a Layer 3 Switching Engine, perform this task in privileged mode:

Table 7-6: Mapping CoS Values to DSCP Values

Task	Command
Step 1 Map CoS values to DSCP values.	**set qos cos-dscp-map** *dscp1 dscp2 dscp3 dscp4 dscp5 dscp6 dscp7 dsc***p8**
Step 2 Verify the configuration.	**show qos maps [cos-dscp-map | ipprec-dscp-map | dscp-cos-map | policed-dscp-map]**

Enter 8 DSCP values to which QoS maps CoS values 0 through 7. This example shows how to map CoS values to DSCP values:

```
Console> (enable) set qos cos-dscp-map 20 30 1 43 63 12 13 8
QoS cos-dscp-map set successfully.
Console> (enable)
```

To revert to default CoS to DSCP value mapping, perform this task in privileged mode:

Table 7-7: Clearing CoS to DSCP Mappings

Task	Command
Step 1 Revert to CoS value/DSCP value map defaults.	**clear qos cos-dscp-map**
Step 2 Verify the configuration.	**show qos maps [cos-dscp-map | ipprec-dscp-map | dscp-cos-map | policed-dscp-map]**

This example shows how to revert to CoS-DSCP map defaults:

```
Console> (enable) clear qos cos-dscp-map
QoS cos-dscp-map setting restored to default.
Console> (enable)
```

Lesson Summary

This lesson accomplished the following:

- Defined the limitation of passing CoS information through the MSFC
- Remarked a packet's CoS markings with ToS markings

Next Steps

After completing this lesson, go to:

- Congestion Management

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mqc/mcli.htm

Congestion Management

Overview

When congestion occurs in the queue of a router, or an MSFC, various queuing techniques can be used to influence which packets get forwarded out of the queue before other packets. In this lesson, two queuing methods, Class-Based Weighted Fair Queuing (CB-WFQ) and Low Latency Queuing (LLQ) will be discussed. Additionally, a method of reserving bandwidth for an application, RSVP, will be demonstrated.

Importance

In a network carrying latency-sensitive traffic, it is critical that such traffic be forwarded ahead of traffic that can tolerate delays.

Objectives

Upon completing this lesson, you will be able to:

- Configure Class-Based Weighted Fair Queuing (CB-WFQ)
- Configure Low Latency Queuing (LLQ)
- Reserve bandwidth with RSVP

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Remote Access Networks (BCRAN) course or have equivalent knowledge

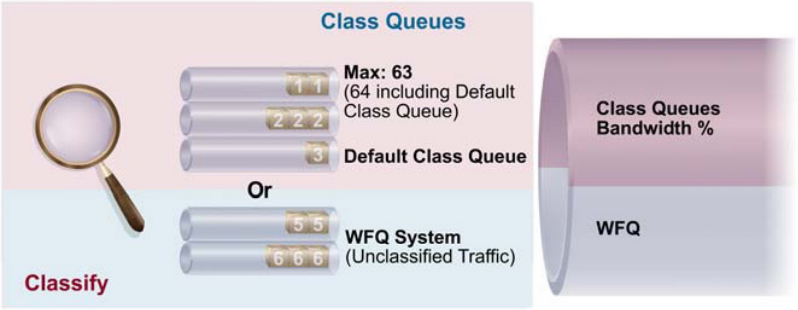
Outline

This lesson includes these sections:

- Overview
- Class-Based Weighted Fair Queuing (CB-WFQ)
- Low Latency Queuing (LLQ)
- RSVP
- Summary

Class-Based Weighted Fair Queuing (CB-WFQ)

This section introduces and demonstrates Class-Based Weighted Fair Queuing as a congestion management tool useful for data traffic requiring a minimum bandwidth.

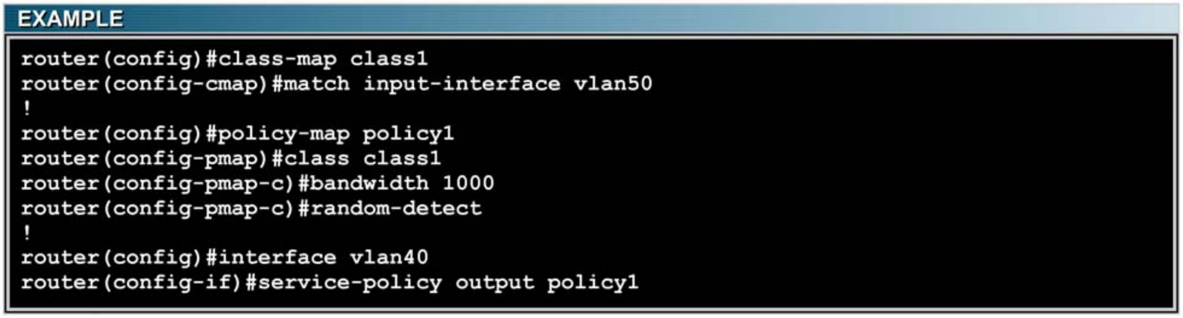


While Cisco supports multiple congestion management tools (e.g., Custom Queuing (CQ), Priority Queuing (PQ), and Weighted Fair Queuing (WFQ)), this module will focus on two of the higher-end queuing techniques, Class-Based Weighted Fair Queuing (CB-WFQ) and Low Latency Queuing (LLQ). Both of these congestion management tools work with the MQC approach discussed in the *Layer 3 Marking* lesson in this module.

Note	The Congestion Management tools presented in this lesson apply to MSFC configuration. They do not apply to the Cat OS.

CB-WFQ uses MQC to define specific classes of traffic and reserves an amount of bandwidth for each of the classes. This prevents any one type of traffic from starving out any other traffic, while still providing minimum bandwidth levels for specific traffic types. Up to 64 queues can be defined for handling the configured classes. Traffic within each class is handled in a Fair Queuing fashion, which gives preferential treatment to lower bandwidth flows. If a default class is not created, the remaining unclassified traffic is handled with Weighted Fair Queuing (WFQ). WFQ, like Fair Queuing, also gives preferential treatment to low bandwidth flows. However, WFQ uses the IP Precedence value of the packets, in combination with the packets' size, when determining the weight that will be used for sequencing the packets.

CB-WFQ Configuration Example



EXAMPLE

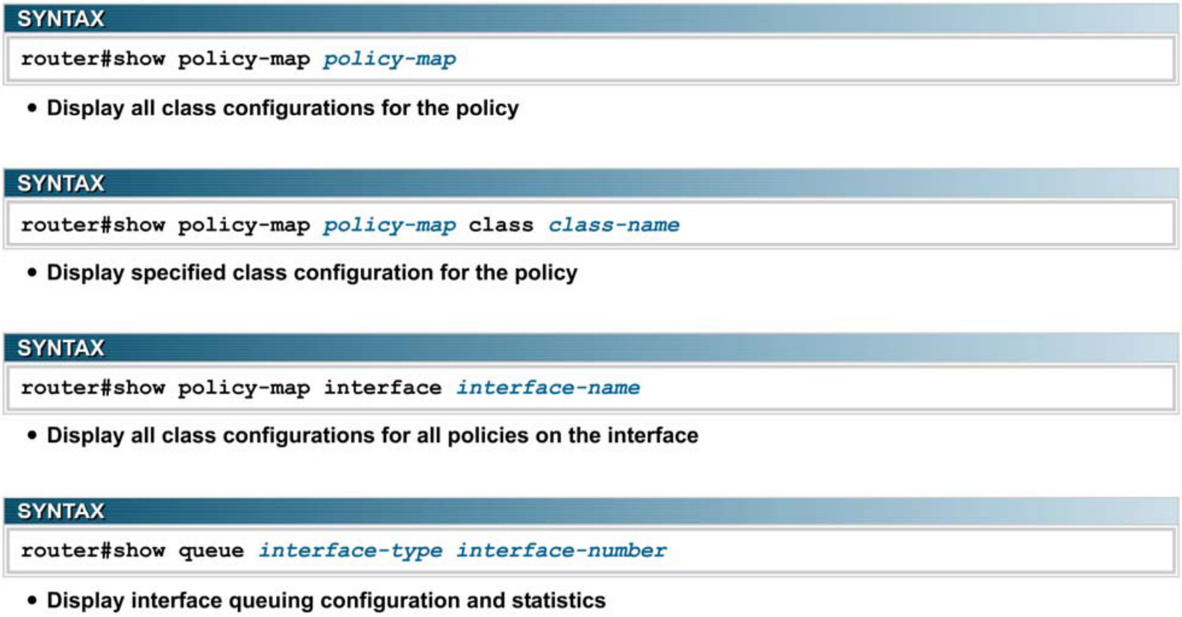
```
router(config)#class-map class1
router(config-cmap)#match input-interface vlan50
!
router(config)#policy-map policy1
router(config-pmap)#class class1
router(config-pmap-c)#bandwidth 1000
router(config-pmap-c)#random-detect
!
router(config)#interface vlan40
router(config-if)#service-policy output policy1
```

In this example, a CLASS-MAP named “class1” is created. All traffic entering the Catalyst on the VLAN 50 interfaces is placed in the “class1” class.

A POLICY-MAP named “policy1” reserves 1 Mbps of bandwidth for traffic in the “class1” class. Additionally, Weighted Random Early Detection (WRED) is enabled for the “class1” class. (WRED will be discussed later in this module.)

The “policy1” policy is applied to output traffic on interface VLAN 40. Notice that the policy is modular, and can therefore be applied to additional VLAN interfaces.

CB-WFQ Monitoring Commands



SYNTAX

```
router#show policy-map policy-map
```

- Display all class configurations for the policy

SYNTAX

```
router#show policy-map policy-map class class-name
```

- Display specified class configuration for the policy

SYNTAX

```
router#show policy-map interface interface-name
```

- Display all class configurations for all policies on the interface

SYNTAX

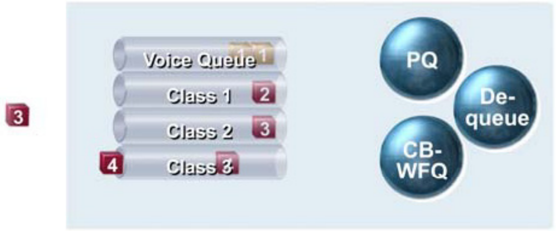
```
router#show queue interface-type interface-number
```

- Display interface queuing configuration and statistics

Cisco provides multiple **`show`** commands for verifying CB-WFQ configurations, as shown above. Note that the counters displayed after issuing the **`show policy-map interface`** command are updated only if congestion is present on the interface.

Low Latency Queuing (LLQ)

This section introduces and demonstrates Low Latency Queuing as a congestion management tool useful for data and latency-sensitive traffic requiring a minimum bandwidth.

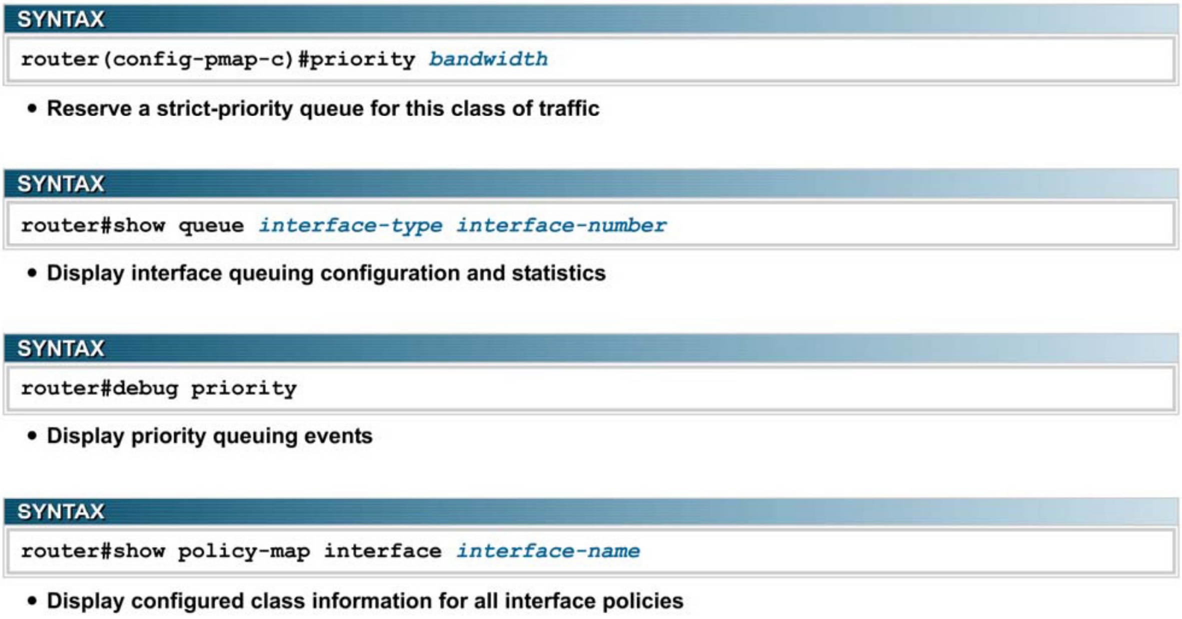


- **Consistent configuration and operation across all media types**
  - Frame Relay
  - Leased lines
  - ATM
- **Entrance criteria to a class can be defined by an ACL**
  - Not limited to UDP ports as with IP RTP Priority
  - Use of IP RTP Priority should be phased out
  - Ensure trust boundary is defined to ensure simple classification and entry to a queue

Low Latency Queuing (LLQ) is an extension to CB-WFQ. Like CB-WFQ, LLQ is configured in MQC-style, with CLASS-MAPs and a POLICY-MAP. However, LLQ can place a specified class of traffic into a priority queue. So, in addition to reserving an amount of bandwidth for the priority queue, traffic in that queue will receive priority treatment over other queues.

LLQ is a popular queuing technique for latency-sensitive applications, such as voice or video. An earlier queuing technique for prioritizing voice was IP RTP Priority. However, IP RTP Priority attempted to recognize voice traffic by placing even numbered UDP port numbers (typically in a range of 16,384 – 32,767) into the priority queue. While this is sufficient for streaming voice traffic, additional traffic types, such as TCP call-setup traffic, cannot be placed into the priority queue. LLQ overcomes this limitation by allowing all traffic within a defined class to be placed in a priority queue.

LLQ Commands



SYNTAX

```
router(config-pmap-c)#priority bandwidth
```

- Reserve a strict-priority queue for this class of traffic

SYNTAX

```
router#show queue interface-type interface-number
```

- Display interface queuing configuration and statistics

SYNTAX

```
router#debug priority
```

- Display priority queuing events

SYNTAX

```
router#show policy-map interface interface-name
```

- Display configured class information for all interface policies

LLQ configuration differs from CB-WFQ configuration by a single command. In policy-map-class configuration mode, instead of issuing the **`bandwidth`** ***`bandwidth`*** command, the **`priority`** ***`bandwidth`*** command is issued.

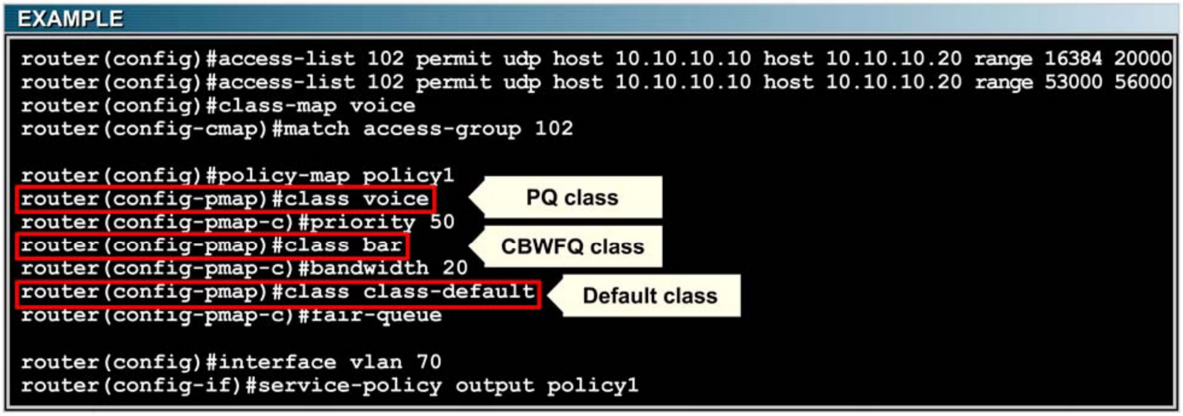
Following is a collection of commands that can be used to verify and troubleshoot the LLQ configuration:

`show queue` ***`interface-type interface-number`***: Displays the queuing configuration for an interface.

`debug priority`: Displays queuing events as they happen.

`show policy-map` ***`interface-name`***: Displays what policy is applied to an interface, in addition to the classes and class characteristics that make up the policy.

LLQ Example



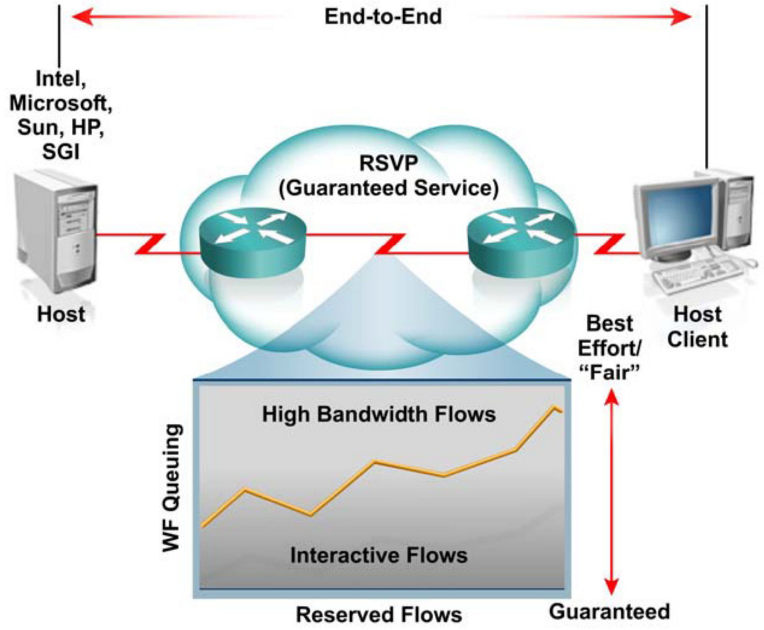
EXAMPLE

```
router(config)#access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384 20000
router(config)#access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000 56000
router(config)#class-map voice
router(config-cmap)#match access-group 102

router(config)#policy-map policy1
router(config-pmap)#class voice
router(config-pmap-c)#priority 50
router(config-pmap)#class bar
router(config-pmap-c)#bandwidth 20
router(config-pmap)#class class-default
router(config-pmap-c)#fair-queue

router(config)#interface vlan 70
router(config-if)#service-policy output policy1
```

In this example a class named “voice” matches a specified range of UDP ports, sourced from 10.10.10.10 and destined for 10.10.10.20. The POLICY-MAP “policy1” gives strict priority to the “voice” class, up to a bandwidth of 50 kbps. The class “bar” reserves 20 kbps of bandwidth, and the “class-default” class is configured to use WFQ among the flows within the “class-default” class. The policy is then applied to traffic leaving the VLAN 70 interface.

RSVP

This section introduces RSVP as a tool that can reserve a specified amount of bandwidth for the duration of an application, such as a VoIP phone call.

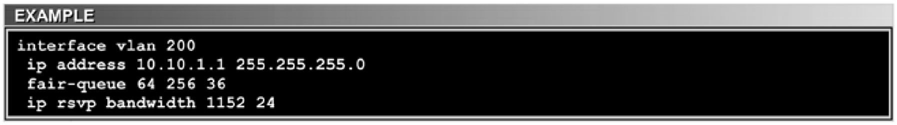


RSVP is an IETF Internet Standard (RFC 2205) protocol for allowing an application to dynamically reserve network bandwidth. RSVP enables applications to request a specific QoS for a data flow, as shown above. Cisco's implementation also allows RSVP to be initiated within the network, using configured proxy RSVP. Network managers can thereby take advantage of the benefits of RSVP in the network, even for non-RSVP-enabled applications and hosts.

Hosts and routers use RSVP to deliver QoS requests to the routers along the paths of the data stream and to maintain router and host state to provide the requested service, usually bandwidth and latency. RSVP uses a mean data rate; the largest amount of data the router will keep in queue, and minimum QoS to determine bandwidth reservation.

WFQ or WRED acts as the workhorse for RSVP, setting up the packet classification and scheduling required for the reserved flows. Using WFQ, RSVP can deliver an integrated services guaranteed service. Using WRED, it can deliver a controlled load service. WFQ continues to provide its advantageous handling of non-reserved traffic by expediting interactive traffic and fairly sharing the remaining bandwidth between high-bandwidth flows. WRED provides its commensurate advantages for non-RSVP flow traffic. RSVP can be deployed in existing networks with a software upgrade.

RSVP Example



```
interface vlan 200
 ip address 10.10.1.1 255.255.255.0
 fair-queue 64 256 36
 ip rsvp bandwidth 1152 24
```

In this example, interface VLAN 200 is being enabled for RSVP reservations. The command **`ip rsvp bandwidth 1152 24`** specifies that a total of 1.152 Mbps may be reserved by multiple RSVP applications. However, a single reservation can be a maximum of 24 kbps. Also, notice that WFQ is being enabled for the interface. When used with RSVP, WFQ gives an extremely low weight to RSVP flows.

Lesson Summary

This lesson accomplished the following:

- Demonstrated the configuration of Class-Based Weighted Fair Queuing (CB-WFQ)
- Demonstrated the configuration of Low Latency Queuing (LLQ)
- Demonstrated how to reserve bandwidth with RSVP

Next Steps

After completing this lesson, go to:

- Congestion Avoidance

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t7/pqcbwfq.htm

Congestion Avoidance

Overview

When a router's queue fills to capacity, additional packets will be discarded. This packet discard behavior can cause undesirable behavior in TCP flows. This lesson discusses the effects of packet discard and introduces tools, both at Layer 2 and at Layer 3, to prevent queues from filling to capacity.

Importance

Congestion avoidance is crucial to maximizing bandwidth in a network that carries TCP flows and experiences periodic congestion.

Objectives

Upon completing this lesson, you will be able to:

- Describe the Catalyst 6500's Layer 2 congestion avoidance approach
- Configure Layer 3 congestion avoidance on the MSFC with WRED

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Remote Access Networks (BCRAN) course or have equivalent knowledge

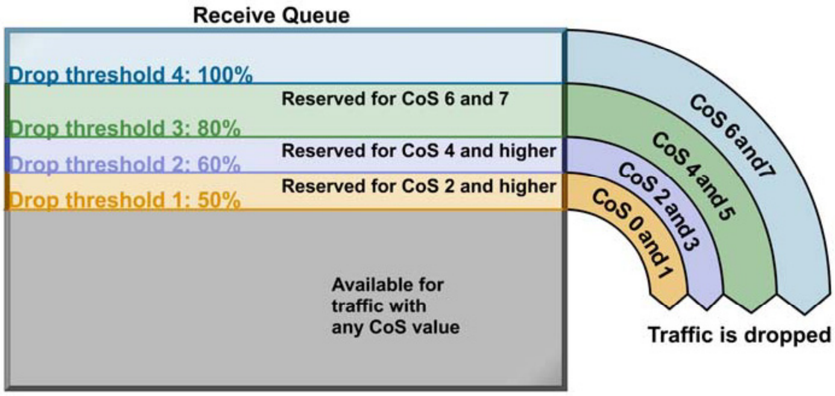
Outline

This lesson includes these sections:

- Overview
- Layer 2 Congestion Avoidance
- Layer 3 Congestion Avoidance
- Summary

Layer 2 Congestion Avoidance

This section explains the Catalyst 6500's approach to congestion avoidance at Layer 2, based on Layer 2 Class of Service markings.



Congestion avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks, as opposed to congestion management techniques that operate to control congestion after it occurs. The Catalyst 6500 can perform congestion avoidance at Layer 2, or the MSFC can perform congestion avoidance, using WRED.

Within each port-based ingress queue, the Catalyst 6500 provides four user-defined levels of packet drop thresholds for incoming traffic. This provides a proven mechanism to ensure that higher priority traffic is given preference when a port's receive queue fills beyond a user-defined threshold. These capabilities apply to both dedicated queues as well as the shared queues. On an egress port there are two user defined thresholds that can be defined ensuring high priority traffic is given precedence when a transmit queue reaches a particular threshold due to network congestion on the egress side.

If a port is configured with the **`set port qos`** ***`mod_num/port_num`*** **`trust trust-cos`** command, QoS implements CoS-value-based receive drop thresholds to avoid congestion in received traffic.

Ports with a single receive queue have this default drop threshold configuration:

- Using receive queue drop threshold 1, the switch drops incoming frames with CoS 0 or 1 when the receive queue buffer is 50 percent or more full.
- Using receive queue drop threshold 2, the switch drops incoming frames with CoS 2 or 3 when the receive queue buffer is 60 percent or more full.

- Using receive queue drop threshold 3, the switch drops incoming frames with CoS 4 or 5 when the receive queue buffer is 80 percent or more full.

- Using receive queue drop threshold 4, the switch drops incoming frames with CoS 6 or 7 when the receive queue buffer is 100 percent full.

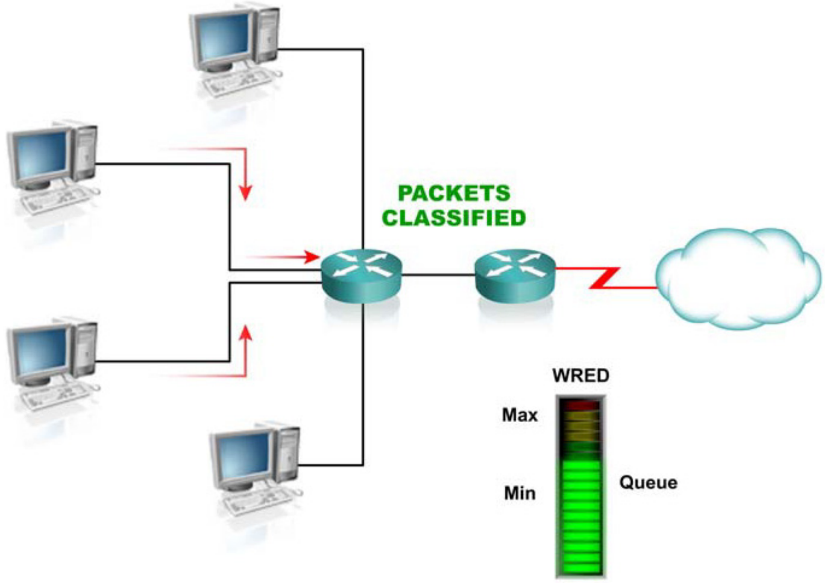
Ports with dual receive queues have this default drop threshold configuration:

- Frames with CoS 0, 1, 2, 3, 4, 6, or 7 go to the standard receive queue.
  - Using standard receive queue drop threshold 1, the switch drops incoming frames with CoS 0 or 1 when the receive queue buffer is 50 percent or more full.
  - Using standard receive queue drop threshold 2, the switch drops incoming frames with CoS 2 or 3 when the receive queue buffer is 60 percent or more full.
  - Using standard receive queue drop threshold 3, the switch drops incoming frames with CoS 4 or 5 when the receive queue buffer is 80 percent or more full.
  - Using standard receive queue drop threshold 4, the switch drops incoming frames with CoS 6 or 7 when the receive queue buffer is 100 percent full.

- Frames with CoS 5 go to the strict-priority receive queue (queue 2), where the switch drops incoming frames only when the strict-priority receive queue buffer is 100 percent full.

Layer 3 Congestion Avoidance

This section details the operation and configuration of Weighted Random Early Detection (WRED) as a tool for Layer 3 congestion avoidance.



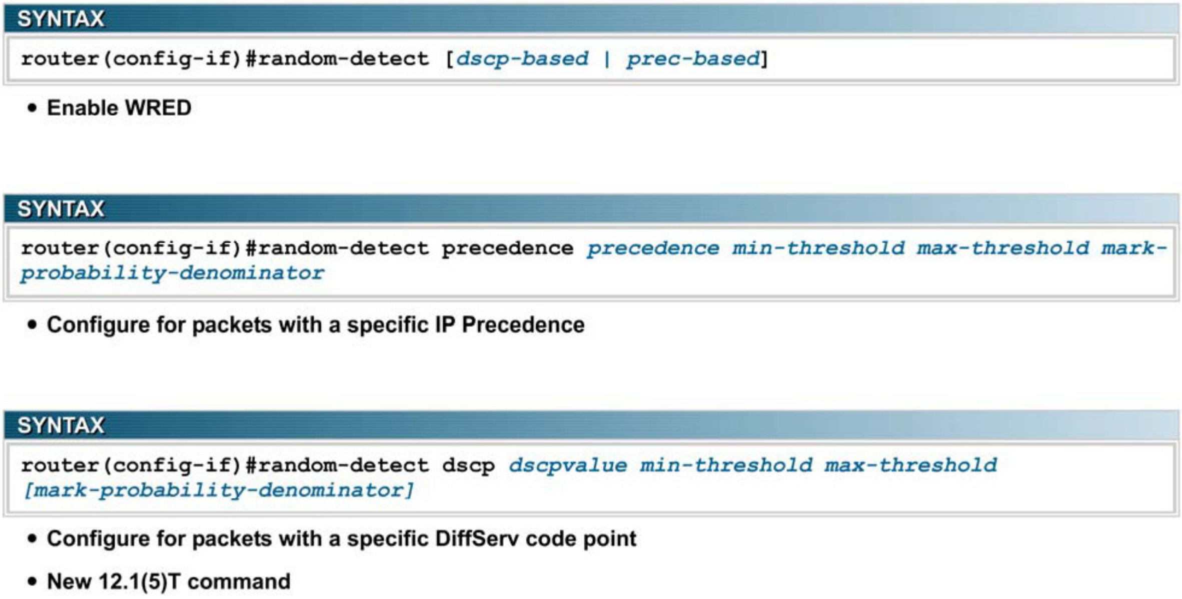
While queuing tools provide congestion management, there is still a need for congestion avoidance, especially for TCP flows. When congestion occurs on an output queue, the default behavior is to "tail drop" packets. Tail dropping packets involves the discarding of packets attempting to enter the queue, after the queue is full.

The result of a queue filling up is the simultaneous tail drop behavior of all flows attempting to enter the queue. This action causes all of the TCP flows to enter TCP Slow Start, which is the reducing of the TCP window size to 1. The window size then increases exponentially up to half of the original congestion window size. At that point, the window size increases linearly. This phenomenon of multiple TCP flows simultaneously entering TCP Slow Start is called Global Synchronization, which results in unused bandwidth, due to the small window size of all TCP flows.

Therefore, a mechanism is needed to prevent Global Synchronization. An industry-standard method, called RED (Random Early Detection), will begin to randomly discard packets as the queue nears capacity. Since the queue never fills to capacity, due to the random discarding of packets, then Global Synchronization never occurs.

One negative aspect of RED, however, is that it does not distinguish between flows. Cisco's implementation takes RED a step further, and discards packets at a rate depending on the IP Precedence or DSCP values. Support for DSCP-based WRED was introduced in IOS 12.1(5)T.

WRED Syntax



SYNTAX

```
router(config-if)#random-detect [dscp-based | prec-based]
```

- Enable WRED

SYNTAX

```
router(config-if)#random-detect precedence precedence min-threshold max-threshold mark-probability-denominator
```

- Configure for packets with a specific IP Precedence

SYNTAX

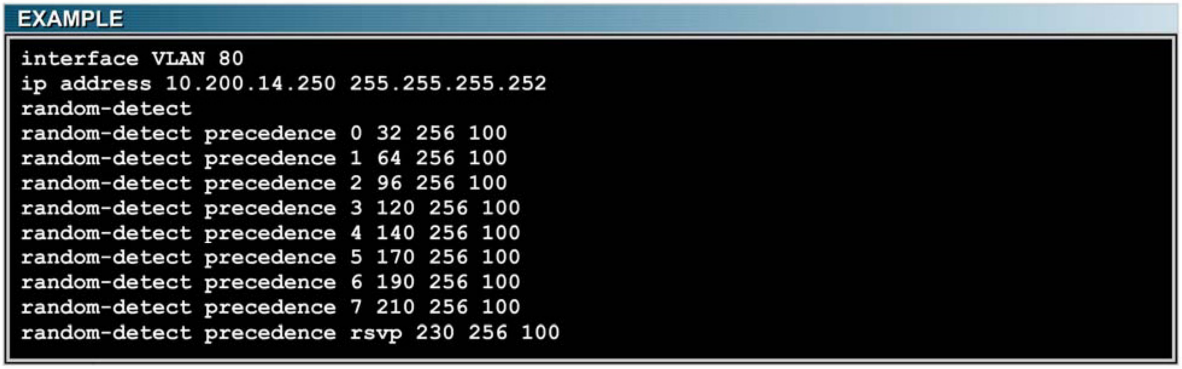
```
router(config-if)#random-detect dscp dscpvalue min-threshold max-threshold
[mark-probability-denominator]
```

- Configure for packets with a specific DiffServ code point
- New 12.1(5)T command

To enable Weighted Random Early detection on an interface, use the command **`random-detect`** `[`**`dscp-based`** `|` **`prec-based`**`]`, where **`dscp-based`** causes WRED to discard based on Differentiated Services Code Point values, and **`prec-based`** causes WRED to discard based on IP Precedence values. If neither dscp-based nor prec-based is specified, then WRED defaults to prec-based.

WRED has default values of when it will begin discarding packets. It also has a default value for what that probability of discard is for various IP Precedence and DSCP values. These parameters may be altered. To specify the characteristics for prec-based WRED, use the command **`random-detect precedence`** ***`precedence min-threshold max-threshold mark-probability-denominator`***, where ***`precedence`*** is an IP Precedence value in the range of 0 – 7, ***`min-threshold`*** specifies the average queue depth after which WRED will begin discarding packets, ***`max-threshold`*** is the queue depth after which all packets will be discarded, and ***`mark-probability-denominator`*** is the fraction of packets dropped when the queue depth approaches the max-threshold. For example, a mark probability denominator of 100 would indicate that when the queue depth equaled the max-threshold, there would be a 1 in 100 chance that a packet with the specified IP Precedence value would be discarded. The probability of packet discard increases linearly from the min-threshold, with a zero probability, up to the max-threshold with a probability defined as 1/(*mark-probability-denominator*). Similarly, WRED's discard parameters can be adjusted for DSCP-based WRED with the command **`random-detect dscp`** ***`dscpvalue min-threshold max-threshold`*** `[`***`mark-probability-denominator`***`]`, where ***`dscpvalue`*** can be a DSCP value in the range of 0 – 63.

WRED Example



EXAMPLE

```
interface VLAN 80
ip address 10.200.14.250 255.255.255.252
random-detect
random-detect precedence 0 32 256 100
random-detect precedence 1 64 256 100
random-detect precedence 2 96 256 100
random-detect precedence 3 120 256 100
random-detect precedence 4 140 256 100
random-detect precedence 5 170 256 100
random-detect precedence 6 190 256 100
random-detect precedence 7 210 256 100
random-detect precedence rsvp 230 256 100
```

In the example, IP Precedence-based WRED is enabled on the VLAN 80 interface. Additionally, the minimum threshold, maximum threshold, and mark probability denominator values have been altered such that as any packet approaches the **`max-threshold`** queue depth, there will be a 1 in 100 chance that the packet will be discarded. Notice that higher priorities have higher **`min-threshold`** values, indicating that the queue depth would have to be greater for high priority packets to be discarded.

Lesson Summary

This lesson accomplished the following:

- Described the Catalyst 6500's Layer 2 congestion avoidance approach
- Illustrated the configuration of Layer 3 congestion avoidance on the MSFC with WRED

Next Steps

After completing this lesson, go to:

- Policing

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.htm

Policing

Overview

While congestion management tools help guarantee a minimum amount of bandwidth for a class of traffic, policing can limit the amount of bandwidth consumed by a particular class of traffic. This lesson explains how to police bandwidth at Layer 2 and at Layer 3.

Importance

The ability to limit bandwidth can be critical in avoiding oversubscription and maintaining service level agreements (SLAs).

Objectives

Upon completing this lesson, you will be able to:

- Configure QoS ACLs in the Cat OS
- Configure CAR on the MSFC

Learner Skills and Knowledge

To fully benefit from this lesson, you must have these prerequisite skills and knowledge:

- Completed the Building Cisco Remote Access Networks (BCRAN) course or have equivalent knowledge

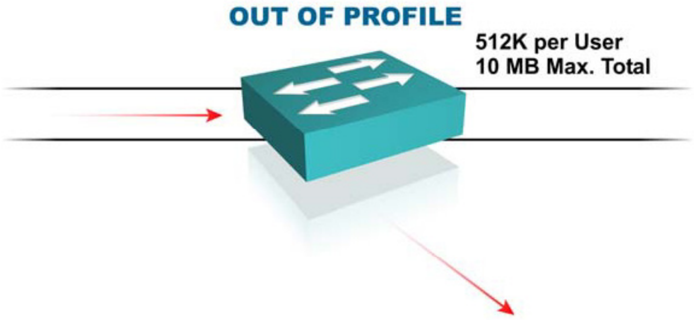
Outline

This lesson includes these sections:

- Overview
- Policing Rules
- Committed Access Rate (CAR)
- Summary

Policing Rules

This section introduces the concept of policing rules and demonstrates the configuration of policing rules using the Cat OS.



Policing limits the amount of bandwidth sent into or out of an interface. If the specified limit is exceeded, the exceeding traffic can either be discarded or reclassified with a lower priority. Policing can be accomplished via the Cat OS using QoS ACLs, or the IOS CAR (Committed Access Rate) feature may be used.

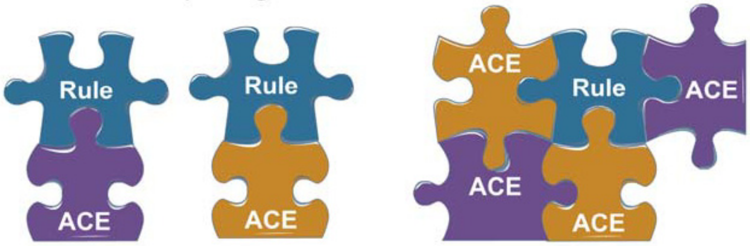
Types of Policing Rules

Microflow

- Bandwidth limit applied separately to each applicable ACE
- Maximum of 63 policing rules

Aggregate

- Bandwidth limits applied cumulatively to all applicable ACEs
- Maximum of 1023 policing rules



In each policing rule, you specify if out-of-profile packets are to be dropped or to have a new DSCP value applied to them (applying a new DSCP value is called "markdown"). Since out-of-profile packets do not retain their original priority, they are not counted as part of the bandwidth consumed by in-profile packets.

For all policing rules, QoS uses a configurable table that maps received DSCP values to marked-down DSCP values. When markdown occurs, QoS gets the marked-down DSCP value from the table. You cannot specify a marked-down DSCP value in individual policing rules.

Note By default, the markdown table is configured so that no markdown occurs: the marked-down DSCP values are equal to the original DSCP values. To enable markdown, configure the table as appropriate for your network.

You give each policing rule a unique name when you create it, and then use the name to include the policing rule in an ACE. The same policing rule can be used in multiple ACEs.

You can create two kinds of policing rules: *microflow* and *aggregate*.

QoS applies the bandwidth limit specified in a microflow policing rule separately to each flow that matches any ACEs that use that particular microflow policing rule. You can create up to 63 microflow policing rules.

QoS applies the bandwidth limits specified in an aggregate policing rule cumulatively to all flows that match any ACEs that use that particular aggregate policing rule. You can create up to 1023 aggregate policing rules.

You can include both a microflow policing rule and an aggregate policing rule in each ACE to police a flow based on both its own bandwidth utilization, and on its bandwidth utilization combined with that of other flows.

For example, you could create a microflow policing rule named "group_individual" with bandwidth limits suitable for individuals in a group and you could create an aggregate policing rule named "group_all" with bandwidth limits suitable for the group as a whole. You could include both policing rules in ACEs that match the group's traffic. The combination would affect individuals separately and the group cumulatively.

You can include a microflow policing rule only in IP ACEs. You cannot include a microflow policing rule in IPX or MAC ACEs. IPX and MAC ACEs support only aggregate policing rules.

By default, microflow policing rules only affect routed traffic. To enable microflow policing of nonrouted traffic, enter the **`set qos bridged-packet-microflow-policing`** command.

QoS does not apply microflow policing rules to Multilayer Switching (MLS) candidate frames.

To avoid inconsistent results, all ACEs that include the same aggregate policing rule must use the same ACE keyword: **`trust-dscp`**, **`trust-ipprec, trust-cos`**, or **`dscp`**. If the ACE uses the dscp keyword, all traffic that matches the ACE must come through ports configured with the same port keyword: **`trust-dscp`**, **`trust-ipprec`**, **`trust-cos`**, or **`untrusted`**. If the ACL is attached to a VLAN, all ports in the VLAN must be configured with the same port keyword.

For ACEs that include both a microflow policing rule and an aggregate policing rule, QoS responds to an out-of-profile status from either policing rule and, as specified by the policing rule, applies a new DSCP value or drops the packet. If both policing rules return an out-of-profile status, and if either policing rule specifies that the packet is to be dropped, it is dropped, otherwise QoS applies a new DSCP value.

Attaching ACLs

- **Filter by port**
  - Default
- **Filter by VLAN**
  - IPX
  - IP

You can configure each port for either port-based QoS (default) or VLAN-based QoS and attach ACLs to the selected interface. You can attach up to three named ACLs, one of each type (IP, IPX, and Ethernet) to each port and VLAN.

On ports configured for VLAN-based QoS, you can attach named ACLs to the port's VLAN; or for a trunk, you can attach named ACLs to any VLANs allowed on the trunk.

On a port configured for VLAN-based QoS, traffic received through the port is compared to any named ACLs attached to the port's VLAN. If you do not attach any named ACLs to the port's VLAN, or if the traffic does not match an ACE in a named ACL, QoS compares the traffic received through the port to the default ACLs.

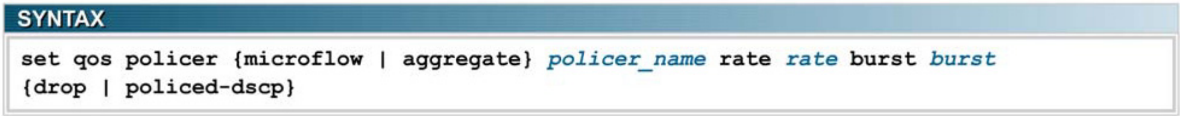
On a trunk configured for VLAN-based QoS, traffic received through the port is compared to any named ACLs attached to the traffic's VLAN. For traffic in VLANs that have no named ACLs attached, or if the traffic does not match an ACE in a named ACL, QoS compares the traffic to the default ACLs.

On ports configured for port-based QoS, you can attach named ACLs to the port.

On a port configured for port-based QoS, traffic received through the port is compared to any named ACLs attached to the port. If you do not attach any named ACLs to the port, or if the traffic does not match an ACE in a named ACL, QoS compares the traffic received through the port to the default ACLs.

On a trunk configured for port-based QoS, traffic in all VLANs received through the port is compared to any named ACLs attached to the port. If you do not attach any named ACLs to the port, or if the traffic does not match an ACE in a named ACL, QoS compares the traffic received through the port to the default ACLs.

Creating Policing Rules



SYNTAX

```
set qos policer {microflow | aggregate} policer_name rate rate burst burst
{drop | policed-dscp}
```

To create a policing rule, perform this task in privileged mode:

Table 7-8: Defining Policing Rules

Task	Command
Step 1 Create a policing rule.	`set qos policer {microflow	aggregate} policer_name rate rate burst burst {drop	policed-dscp}`
Step 2 Verify the configuration.	`show qos policer {config	runtime} {microflow	aggregate	all}`

The *policer_name* parameter can be up to 31 characters long, is case sensitive, and may include a-z, A-Z, 0-9, the dash character (-), the underscore character (_), and the period character (.). Policing rule names must start with an alphabetic character (not a digit) and must be unique across all microflow and aggregate policing rules. You cannot use keywords from any command as a policing rule name.

The valid values for the *rate* parameter are 32 Kbps (entered as 32) to 8 Gbps (entered as 8000000), or to classify all traffic as out of profile, set the *rate* parameter to zero (0).

The valid values for the *burst* parameter are 1 Kb (entered as 1) to 32 Mb (entered as 32000).

Note The recommended minimum value of burst size is 32 Kb; specifying a smaller value might result in a lower than specified rate. You can experiment with smaller values, but if there are problems, you should up your burst to this minimum recommended value.

QoS programs the hardware with values that are multiples of 32K, not with the specific value entered.

Enter either the **drop** keyword to cause all out-of-profile traffic to be dropped or the **policed-dscp** keyword to cause all out-of-profile traffic to be marked down as specified in the markdown map.

This example shows how to create a microflow policing rule with a 1-Mbps rate limit and a 10-Mb burst limit that marks down out-of-profile traffic:

```
Console> (enable) set qos policer microflow my-micro rate 1000 burst 10000
policed-dscp
Hardware programming in progress...
QoS policer for microflow my-micro created successfully.
Console> (enable)
```

Deleting Policing Rules

You can only delete policing rules if they are not attached to any interfaces.

To delete one or all policing rules, perform this task in privileged mode:

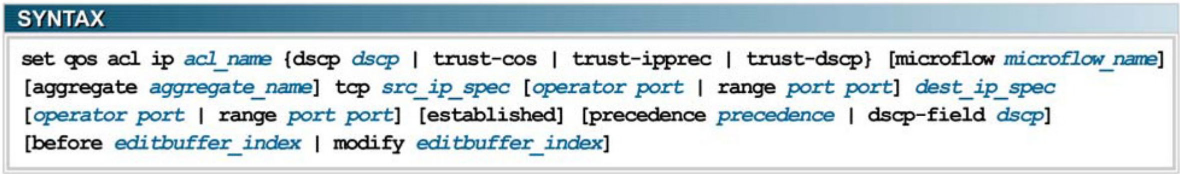
Table 7-9: Clearing Policing Rules

Task	Command
Step 1 Delete one or all policing rules.	**clear qos policer {microflow | aggregate}** {*policer_name* | **all**}
Step 2 Verify the configuration.	**show qos policer {config | runtime} {microflow | aggregate | all}**

This example shows how to delete the microflow policing rule named *my_micro*:

```
Console> (enable) clear qos policer microflow my_micro
my_micro QoS microflow policer cleared.
Console> (enable)
```

Creating QoS ACLs



SYNTAX

```
set qos acl ip acl_name {dscp dscp | trust-cos | trust-ipprec | trust-dscp} [microflow microflow_name]
[aggregate aggregate_name] tcp src_ip_spec [operator port | range port port] dest_ip_spec
[operator port | range port port] [established] [precedence precedence | dscp-field dscp]
[before editbuffer_index | modify editbuffer_index]
```

In IP ACEs, specify source and destination IP addresses and masks (represented by the *src_ip_spec* and *dest_ip_spec* parameters in the following sections) in the form *ip_address mask*. The mask is mandatory. Use zero bits, which need not be contiguous, where you want wildcards.

Use any of the following formats for the address and mask:

- Four-part dotted-decimal 32-bit values
- The keyword **any** as an abbreviation for a wildcard address and wildcard mask of 0.0.0.0 255.255.255.255
- The abbreviation **host** *ip_address* for an address and wildcard mask of *ip_address* 0.0.0.0

Port Operator Parameters

In IP ACEs, the *operator* parameter can be **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** keywords with a port or, for the **range** keyword, a pair of port parameters.

IP ACEs for TCP Traffic

To create or modify an IP ACE for TCP traffic, perform this task in privileged mode:

Table 7-10: Configuring QoS ACLs for TCP Traffic

Task	Command
Step 1 Create or modify an IP ACE for TCP traffic.	**set qos acl ip** *acl_name* {**dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] **tcp** *src_ip_spec* [*operator port* | **range** *port port*] *dest_ip_spec* [*operator port* | **range** *port port*] **[established] [precedence** *precedence* | **dscp-field** *dscp*] **[before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2 Verify the configuration.	**show qos acl info {acl_name | all} editbuffer** [*editbuffer_index*]

The **established** keyword matches traffic with the ACK or RST bits set.

This example shows how to create an IP ACE for TCP traffic:

```
Console> (enable) set qos acl ip my_IPacl trust-ipprec microflow my-micro
aggregate my-agg tcp any any

my_IPacl editbuffer modified. Use `commit' command to apply changes.

Console> (enable)
```

IP ACEs for UDP Traffic

To create or modify an IP ACE for UDP traffic, perform this task in privileged mode:

Table 7-11: Configuring QoS ACLs for UDP Traffic

Task	Command
Step 1 Create or modify an IP ACE for UDP traffic.	**set qos acl ip** *acl_name* **{dscp** *dscp* | **trust-cos** | **trust-ipprec** | **trust-dscp**} [**microflow** *microflow_name*] [**aggregate** *aggregate_name*] *udp src_ip_spec* [*operator* port | **range** *port port*] *dest_ip_spec* [*operator port* | **range** *port port*] [precedence *precedence* | **dscp-field** *dscp*] [**before** *editbuffer_index* | **modify** *editbuffer_index*]
Step 2 Verify the configuration.	**show qos acl info {acl_name | all} editbuffer** [*editbuffer_index*]

Only the **range** *operator* keyword accepts a second *port* parameter.

This example shows how to create an IP ACE for UDP traffic:

```
Console> (enable) set qos acl ip my_IPacl trust-ipprec microflow my-micro
aggregate my-agg udp any any

my_IPacl editbuffer modified. Use `commit' command to apply changes.

Console> (enable)
```

Attaching ACLs to Interfaces

You can attach one ACL of each type to each VLAN and to each port configured for port-based QoS. You cannot attach ACLs to a port configured for VLAN-based QoS. When an ACL of a particular type (IP, IPX, or Ethernet) is already attached to an interface, attaching a different ACL of the same type detaches the previous ACL.

To attach an ACL to a port or a VLAN, perform this task in privileged mode:

Table 7-12: Mapping an ACL to a Port or VLAN

Task	Command
Step 1 Attach an ACL to an interface.	**set qos acl map** *acl_name* {*mod_num/port_num* | *vlan*}
Step 2 Verify the configuration.	**show qos acl map** {**config** | **runtime**} {*acl_name* | *mod_num/port_num* | *vlan* | **all**}

This example shows how to attach an ACL named *my_acl* to port 2/1:

```
Console> (enable) set qos acl map my_acl 2/1
Hardware programming in progress...
ACL my_acl is attached to port 2/1.
Console> (enable)
```

This example shows how to attach an ACL named *my_acl* to VLAN 4:

```
Console> (enable) set qos acl map my_acl 4
Hardware programming in progress...
ACL my_acl is attached to vlan 4.
Console> (enable)
```

Detaching ACLs from Interfaces

To detach an ACL from a port or a VLAN, perform this task in privileged mode:

Table 7-13: Clearing ACL Mappings

Task	Command
Step 1 Detach an ACL from an interface.	**clear qos acl map** *acl_name* {*mod_num/port_num* | *vlan* | **all**}
Step 2 Verify the configuration.	**show qos acl map** {**config** | **runtime**} {*acl_name* | *mod_num/port_num* | *vlan* | **all**}

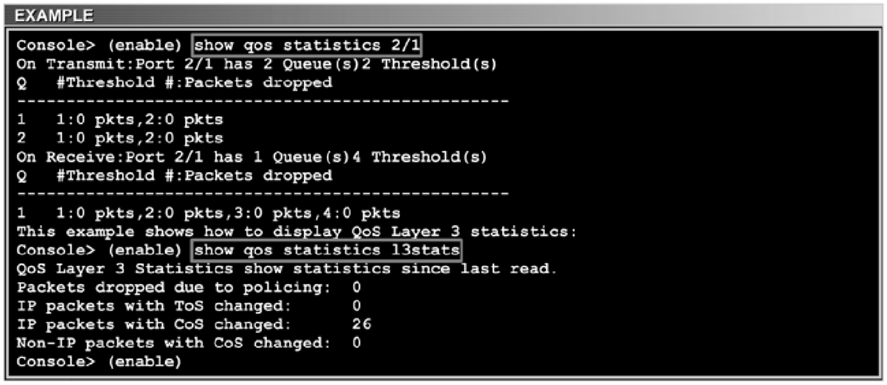
This example shows how to detach an ACL named *my_acl* from port 2/1:

```
Console> (enable) clear qos acl map my_acl 2/1
Hardware programming in progress...
ACL my_acl is detached from port 2/1.
Console> (enable)
```

This example shows how to detach an ACL named *my_acl* from VLAN 4:

```
Console> (enable) clear qos acl map my_acl 4
Hardware programming in progress...
ACL my_acl is detached from vlan 4.
Console> (enable)
```

Displaying QoS Statistics



```
Console> (enable) show qos statistics 2/1
On Transmit:Port 2/1 has 2 Queue(s)2 Threshold(s)
Q   #Threshold #:Packets dropped
-------------------------------------------------
1   1:0 pkts,2:0 pkts
2   1:0 pkts,2:0 pkts
On Receive:Port 2/1 has 1 Queue(s)4 Threshold(s)
Q   #Threshold #:Packets dropped
-------------------------------------------------
1   1:0 pkts,2:0 pkts,3:0 pkts,4:0 pkts
This example shows how to display QoS Layer 3 statistics:
Console> (enable) show qos statistics l3stats
QoS Layer 3 Statistics show statistics since last read.
Packets dropped due to policing:  0
IP packets with ToS changed:      0
IP packets with CoS changed:      26
Non-IP packets with CoS changed:  0
Console> (enable)
```

To display QoS statistics, perform this task:

Table 7-14: Displaying QoS Statistics

Task	Command
Display QoS statistics.	**show qos statistics** {*mod_num*[/*port_num*] | **L3stats**}

This example shows how to display QoS statistics for port 2/1:

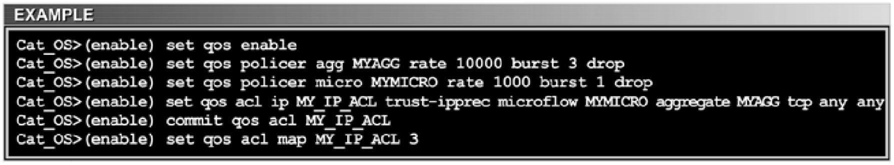
```
Console> (enable) show qos statistics 2/1
On Transmit:Port 2/1 has 2 Queue(s) 2 Threshold(s)
Q #  Threshold #:Packets dropped
---  -----------------------------------------------
1    1:0 pkts, 2:0 pkts
2    1:0 pkts, 2:0 pkts
On Receive:Port 2/1 has 1 Queue(s) 4 Threshold(s)
Q #  Threshold #:Packets dropped
---  -----------------------------------------------
1    1:0 pkts, 2:0 pkts, 3:0 pkts, 4:0 pkts
```

This example shows how to display QoS Layer 3 statistics:

```
Console> (enable) show qos statistics l3stats
QoS Layer 3 Statistics show statistics since last read.
```

```
Packets dropped due to policing: 0
IP packets with ToS changed:     0
IP packets with CoS changed:     26
Non-IP packets with CoS changed: 0
Console> (enable)
```

QoS ACL Example



EXAMPLE

```
Cat_OS>(enable) set qos enable
Cat_OS>(enable) set qos policer agg MYAGG rate 10000 burst 3 drop
Cat_OS>(enable) set qos policer micro MYMICRO rate 1000 burst 1 drop
Cat_OS>(enable) set qos acl ip MY_IP_ACL trust-ipprec microflow MYMICRO aggregate MYAGG tcp any any
Cat_OS>(enable) commit qos acl MY_IP_ACL
Cat_OS>(enable) set qos acl map MY_IP_ACL 3
```

In the above example, an aggregate policing rule called “MYAGG” is created with a rate of 10 Mbps, and a burst of 3 kbps. If an aggregate flow violates this limit, then the traffic will be dropped.

A microflow policing rule named “MYMICRO” is configured for a rate of 1 Mbps with a burst of 1 Kb. Again, exceeding traffic will be dropped.

A QoS ACL named “MY_IP_ACL” is created that references IP Precedence values and sets bandwidth limits using the “MYAGG” and “MYMICRO” policing rules.

The “MY_IP_ACL” QoS ACL is committed to NVRAM, and it is then applied the VLAN 3.

Following are the individual commands:

Enable QoS

```
set qos enable
```

Create Aggregate policing rule

```
set qos policer agg MYAGG rate 10000 burst 3 drop
```

Create Microflow policing rule

```
set qos policer micro MYMICRO rate 1000 burst 1 drop
```

Create QoS ACL

```
set qos acl ip MY_IP_ACL trust-ipprec microflow MYMICRO aggregate MYAGG tcp any
any
```

Commit the ACL

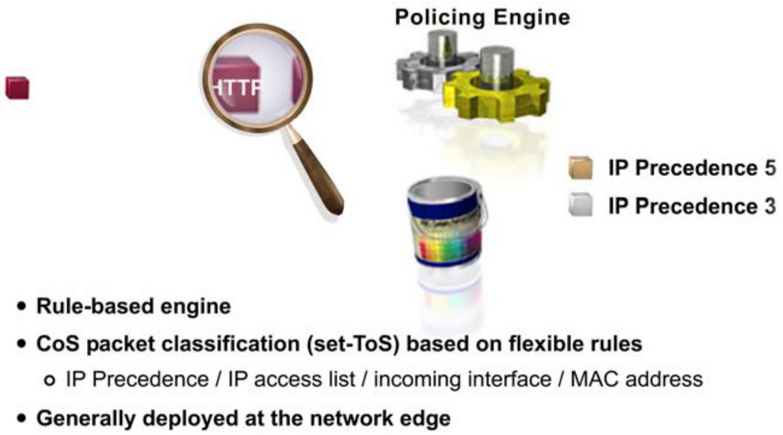
```
commit qos acl MY_IP_ACL
```

Map ACL to VLAN 3

```
set qos acl map MY_IP_ACL 3
```

Committed Access Rate (CAR)

This section describes how the MSFC can be used to police traffic using the IOS' Committed Access Rate feature.



- **Rule-based engine**
- **CoS packet classification (set-ToS) based on flexible rules**
  - IP Precedence / IP access list / incoming interface / MAC address
- **Generally deployed at the network edge**

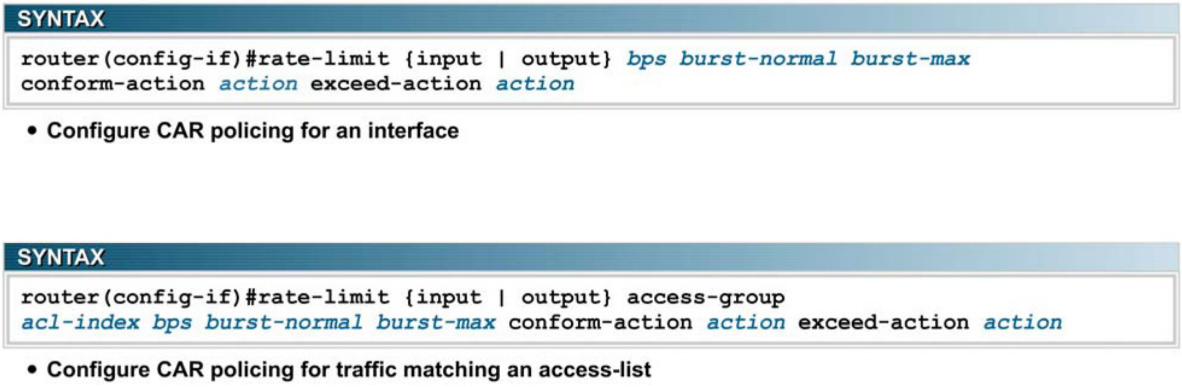
Committed Access Rate (CAR) is a policing tool that can be used on the MSFC to limit the amount of bandwidth given to a particular traffic classification. Additionally, CAR can mark traffic with IP Precedence values.

Typically, traffic is classified using an access-list, and if the bandwidth of the traffic matching the access-list is less than the specified CIR then a "conform" action is performed. If the bandwidth of the traffic matching the access-list is greater than the CIR an "exceed" action is performed. CAR also supports the policing of an interface, not just traffic that matches an access-list.

Following are CAR's supported conform and exceed actions:

- **Transmit** – Send the packet.
- **Drop** – Discard the packet.
- **Continue** – Go to the next CAR rate-limit statement in the list.
- **Set Precedence and Transmit** – Rewrite the IP Precedence value in the packet's ToS byte to the specified value, and send the packet.
- **Set Precedence and Continue** – Rewrite the IP Precedence value in the packet's ToS byte to the specified value, and go to the next CAR rate-limit statement in the list.

CAR Syntax



SYNTAX

```
router(config-if)#rate-limit {input | output} bps burst-normal burst-max
conform-action action exceed-action action
```

- Configure CAR policing for an interface

SYNTAX

```
router(config-if)#rate-limit {input | output} access-group
acl-index bps burst-normal burst-max conform-action action exceed-action action
```

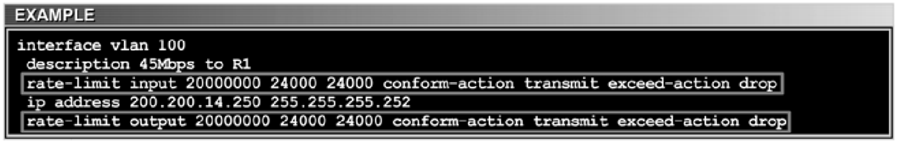
- Configure CAR policing for traffic matching an access-list

To configure rate limiting on an entire interface use the command **rate-limit** {**input** | **output**} ***bps burst-normal burst-max*** **conform-action** ***action*** **exceed-action** ***action***, where ***bps*** is the CIR to which CAR is policing in bits per second, ***burst-normal*** is the committed burst per time interval in bytes, ***burst-max*** is the excess burst per time interval in bytes, and ***action*** can be one of the following:

- **continue** – Evaluate the next **rate-limit** command.
- **drop** – Drop the packet.
- **set-prec-continue** ***new-prec*** – Set the IP Precedence, and continue to the next **rate-limit** command.
- **set-prec-transmit** ***new-prec*** – Set the IP Precedence, and send the packet.
- **transmit** – Send the packet.

To configure rate limiting for traffic specified by an access-list, use the command **`rate-limit`** {**`input`** | **`output`**} **`access-group`** ***`acl-index bps burst-normal burst-max`*** **`conform-action`** ***`action`*** **`exceed-action`** ***`action`***, where ***`acl-index`*** is the access-list number that identifies the traffic to be policed.

CAR Example



EXAMPLE

```
interface vlan 100
 description 45Mbps to R1
 rate-limit input 20000000 24000 24000 conform-action transmit exceed-action drop
 ip address 200.200.14.250 255.255.255.252
 rate-limit output 20000000 24000 24000 conform-action transmit exceed-action drop
```

In the example, the VLAN 100 interface is being policed to 20 Mbps, with a burst capability of 24 kilobytes per time interval. If traffic on the interface conforms to the 20 Mbps rate limit, then the traffic is passed, due to the **`transmit`** conform action. If traffic exceeds the 20 Mbps rate limit, then the traffic is dropped, due to the **`drop`** exceed action.

Also, notice that there are two **`rate-limit`** commands in this example. One is applied inbound on the interface, while one is applied outbound on the interface. While traffic shaping is used on outbound traffic only, policing can be used inbound, outbound, or both simultaneously.

Lesson Summary

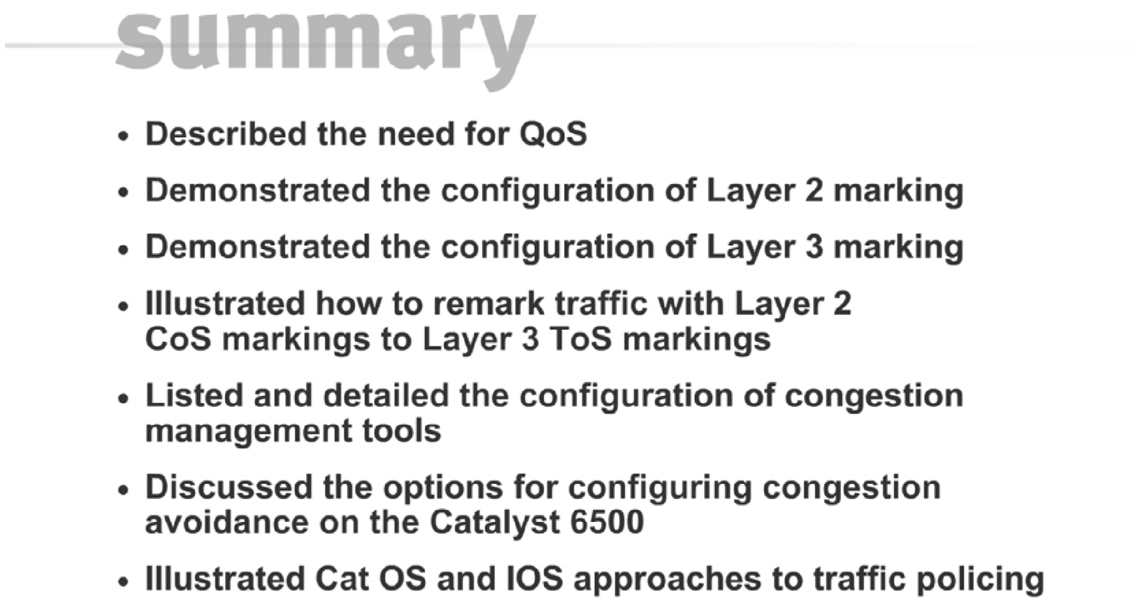
This lesson accomplished the following:

- Demonstrated the configuration of QoS ACLs in the Cat OS
- Demonstrated the configuration of CAR on the MSFC

References

For additional information, refer to these resources:

- http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t2/ftpoli.htm



summary

- Described the need for QoS
- Demonstrated the configuration of Layer 2 marking
- Demonstrated the configuration of Layer 3 marking
- Illustrated how to remark traffic with Layer 2 CoS markings to Layer 3 ToS markings
- Listed and detailed the configuration of congestion management tools
- Discussed the options for configuring congestion avoidance on the Catalyst 6500
- Illustrated Cat OS and IOS approaches to traffic policing

Review Questions

Q1) Which of the following is a QoS Layer 2 marking?

A) IP Precedence

B) CoS

C) DSCP

D) ToS

Answer: B

Q2) How many levels of priority are provided by DSCP?

A) 5

B) 6

C) 8

D) 64

E) 128

Answer: D

Q3) What congestion management technique gives priority treatment to traffic in a defined class?

A) WFQ

B) IP RTP Priority

C) LLQ

D) CB-WFQ

Answer: C

Q4) Which of the following best defines WRED?

A) A congestion avoidance tool that prevents global synchronization

B) A congestion management tool that drops packets based on ToS values

C) A policing tool the prevents a queue from filling

D) A bandwidth reservation tool that prevents tail drop

Answer: A

Q5) Which of the following tools can LIMIT the amount of traffic coming into or out of an interface?

A) CAR

B) LLQ

C) PQ-WFQ

D) CB-WFQ

Answer: A

Appendix A - Working With the Flash File System

This appendix describes how to use the Flash file system on the Catalyst 6000 family switches.

Note For complete syntax and usage information for the commands used in this appendix, refer to the *Catalyst 6000 Family Command Reference* publication.

This appendix consists of these sections:

- Understanding How the Flash File System Works
- Working with the Flash File System

Understanding How the Flash File System Works

The Flash file system on a Catalyst 6500 family supervisor engine provides a number of useful commands to help you manage software image and configuration files.

The Flash file system on the supervisor engine consists of two Flash devices on which you can store files:

- bootflash: onboard Flash memory
- slot0: Flash PC card in the PCMCIA slot

Working with the Flash File System

These sections describe how to work with the Flash file system:

- Setting the Default Flash Device
- Setting the Text File Configuration Mode
- Listing the Files on a Flash Device
- Copying Files
- Deleting Files
- Restoring Deleted Files
- Verifying a File Checksum
- Formatting a Flash Device

Setting the Default Flash Device

When you set the default Flash device for the switch, the default device is assumed when you enter a Flash file system command without specifying the Flash device.

To set the default Flash device, perform this task:

Table A-1: Setting the Default Flash Device

	Task	Command
Step 1	Set the default Flash device for the switch.	**cd [[***m***/][bootflash: | slot0:]]**
Step 2	Verify the default Flash device for the switch.	**pwd** [*mod*]

This example shows how to change the default Flash device to slot0: and verify the default device:

```
Console> (enable) cd slot0:

Console> (enable) pwd

slot0
Console> (enable)
```

Setting the Text File Configuration Mode

When you use text file configuration mode, the switch stores its configuration as a text file in nonvolatile storage, either in NVRAM or Flash memory. This text file consists of commands entered by you to configure various features. For example, if you disable a port, the command to disable that port will be in the text configuration file.

Because the text file only contains commands you have used to configure your switch, it typically uses less NVRAM or Flash memory space than binary configuration mode. Because the text file in most cases requires less space, NVRAM is a good place to store the file. If the text file exceeds NVRAM space, it can also be saved to Flash memory.

When operating in text file configuration mode, most user settings are not immediately saved to NVRAM; configuration changes are only written to DRAM. You will need to enter the **`write memory`** command to store the configuration in nonvolatile storage.

Note VLAN commands are not saved as part of the configuration file when the switch is operating in text mode with the VTP mode set to server.

To set the text file configuration mode, perform this task:

Table A-2: Setting the Text File Configuration Mode

	Task	Command
Step 1	Set the file configuration mode for the system to text.	**set config mode** {**binary** | **text**} [**nvram** | *device*:*file-id*]
Step 2	Verify the file configuration mode for the system.	**show config mode**
Step 3	Save the text file configuration.	**write memory**
Step 4	Display the current runtime configuration.	**show running-config all**
Step 5	Display the startup configuration that will be used after the next reset.	**show config**

This example shows how to configure the system to save its configuration as a text file in NVRAM, verify the configuration mode, and display the current runtime configuration:

```
Console> (enable) set config mode text nvram

Console> (enable) show config mode

Console> (enable) show running-config all

Console> (enable) show config

Console> (enable)
```

Listing the Files on a Flash Device

To list the files on a Flash device, perform one of these tasks:

Table A-3: Listing Files on a Flash Device

Task	Command
■ Display a list of files on a Flash device.	**dir** [[*m/]device*:][*filename*]
■ Display a list of deleted files on a Flash device.	**dir** [[*m/*]*device*:][*filename*] **deleted**
■ Display a list of all files on a Flash device, including deleted files.	**dir** [[*m/*]*device*:][*filename*] **all**
■ Display a detailed list of files on a Flash device.	**dir** [[*m/*]*device*:][*filename*] **long**

This example shows how to list the files on the default Flash device:

```
Console> (enable) dir


-#- -length- -----date/time------ name
  4  3134688 Mar 15 1999 08:27:01 cat6000-sup.5-2-1-CSX.bin
  5  3231989 Jan 24 1999 12:04:40 cat6000-sup.5-1-1-CSX.bin
  6      135 Feb 17 1999 11:30:05 dns_config.cfg

1213952 bytes available (6388224 bytes used)
Console> (enable)
```

This example shows how to list the files on a Flash device other than the default device:

```
Console> (enable) dir slot0:

-#- -length- -----date/time------ name
  1  3209261 Jun 16 1998 13:18:19 cat6000-sup.5-2-1-CSX.bin
  2      135 Jul 17 1998 11:32:53 dns-config.cfg
  3  3231989 Jul 17 1998 16:54:23 cat5000-sup3.4-1-2.bin
  4     8589 Jul 17 1998 17:02:52 6000_config.cfg

9933504 bytes available (6450496 bytes used)
Console> (enable)
```

This example shows how to list the deleted files on the default Flash device:

```
Console> (enable) dir deleted

-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
  1 .D ffffffff 81a027ca   41bdc   22     7004 Apr 01 1998 15:27:45 5002.config.
4.1.98.cfg
  2 .D ffffffff ccce97a3   43644   23     6630 Apr 01 1998 15:36:47 5002.default
.config.cfg
  3 .D ffffffff 81a027ca   45220   15     7004 Apr 19 1998 10:05:59 5002_config.
cfg

1213952 bytes available (6388224 bytes used)
Console> (enable)
```

Copying Files

To copy a file, perform one of these tasks in privileged mode:

Table A-4: Copying Files

Task	Command
■ Copy a Flash file to a TFTP server, rcp server, Flash memory, another Flash device, or to the running configuration.	**copy** *file-id* **{tftp** | **rcp** | **flash** | *file-id* | **config}**
■ Copy a file from a TFTP server, rcp server to Flash memory, to a Flash device, or to the running configuration.	**copy {tftp** | **rcp} {flash** | *file-id* | **config}**
■ Copy a file from Flash memory to a TFTP server, rcp server, to a Flash device, or to the running configuration.	**copy flash {tftp** | **rcp** | *file-id* | **config}**
■ Copy the running configuration to Flash memory, another Flash device, to a TFTP server, or rcp server.	**copy config {flash** | *file-id* | **tftp** | **rcp}**

This example shows how to copy a file from the default Flash device to another Flash device:

```
Console> (enable) copy cat6000-sup.5-2-1-CSX.bin slot0:



13174216 bytes available on device slot0, proceed (y/n) [n]? y

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCcccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
ccccccccccccCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
File has been copied successfully.
Console> (enable)
```

This example shows how to copy a file from a TFTP server to the running configuration:

```
Console> (enable) copy tftp config

IP address or name of remote host []? 172.20.52.3

Name of file to copy from []? dns_config.cfg


Configure using tftp:dns_config.cfg (y/n) [n]? y

/
Finished network download.  (135 bytes)
>>
>> set ip dns server 172.16.10.70 primary
172.16.10.70 added to DNS server table as primary server.
>> set ip dns server 172.16.10.140
172.16.10.140 added to DNS server table as backup server.
>> set ip dns enable
DNS is enabled
>> set ip dns domain corp.com
Default DNS domain name set to corp.com
Console> (enable)
```

This example shows how to download a configuration file from a TFTP server for storage on a Flash device:

```
Console> (enable) copy tftp flash

IP address or name of remote host []? 172.20.52.3

Name of file to copy from []? dns-config.cfg

Flash device [slot0]?
Name of file to copy to [dns-config.cfg]?

9932056 bytes available on device slot0, proceed (y/n) [n]? y

/
File has been copied successfully.
Console> (enable)
```

This example shows how to copy the running configuration to Flash memory:

```
Console> (enable) copy config flash

Flash device [bootflash]? slot0:

Name of file to copy to []? 6000_config.cfg


Upload configuration to slot0:6000_config.cfg
9942096 bytes available on device slot0, proceed (y/n) [n]? y

.....
..........
.......

..........
...........
..

Configuration has been copied successfully.
Console> (enable)
```

This example shows how to upload a configuration file on a Flash device to a TFTP server:

```
Console> (enable) copy slot0:6000_config.cfg tftp

IP address or name of remote host []? 172.20.52.3

Name of file to copy to [6000_config.cfg]?

/

File has been copied successfully.

Console> (enable)
```

This example shows how to upload an image from a remote host into Flash using rcp:

```
Console> (enable) copy rcp flash


IP address or name of remote host []? 172.20.52.3


Name of file to copy from []? 6000_config.cfg


Flash device [bootflash]?
Name of file to copy to [6000_config.cfg]?

4369664 bytes available on device bootflash, proceed (y/n) [n]? y

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCC
File has been copied successfully.
Console> (enable)
```

Deleting Files

Caution If you enter the **`squeeze`** command on a Flash device, you cannot restore files deleted prior to the **`squeeze`** command

To delete files on a Flash device, perform this task in privileged mode:

Table A-5: Deleting Files

	Task	Command
Step 1	Delete a file on a Flash device.	**`delete`** `[[`*`m/`*`]`*`device`*`:]`*`filename`*
Step 2	If desired, permanently remove all deleted files on the Flash device (this operation can take a number of minutes to complete).	**`squeeze`** `[`*`m/`*`]`*`device`***`:`**
Step 3	Verify the files are deleted.	**`dir`** `[[`*`m/`*`]`*`device`*`:][`*`filename`*`]`

This example shows how to delete a file from a Flash device:

```
Console> (enable) delete dns_config.cfg


Console> (enable)
```

This example shows how to permanently remove all deleted files from a Flash device:

```
Console> (enable) squeeze slot0:
```

```
All deleted files will be removed, proceed (y/n) [n]? y

Squeeze operation may take a while, proceed (y/n) [n]? y

Erasing squeeze log
Console> (enable)
```

Restoring Deleted Files

You must specify the index number of a deleted file to identify the file to undelete. The index number for each file appears in the first column of the **dir** command output. A file cannot be undeleted if a valid file with the same name already exists. Instead, you must delete the existing file and then undelete the desired file. A file can be deleted and undeleted up to 15 times.

To restore deleted files on a Flash device, perform this task in privileged mode:

Table A-6: Restoring Deleted Files

	Task	Command
Step 1	Identify the index number of the deleted files on the Flash device.	**dir** [[*m/*]*device*:][*filename*] **deleted**
Step 2	Undelete a file on a Flash device.	**undelete** *index* [[*m/*]*device*:]
Step 3	Verify that the file is restored.	**dir** [[*m/*]*device*:][*filename*]

This example shows how to restore a deleted file:

```
Console> (enable) dir deleted

-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
6 .D ffffffff 42da7f71  657a00   14      135 Jul 17 1998 11:30:05 dns_config.cfg

1213952 bytes available (6388224 bytes used)
Console> (enable) undelete 6

Console> (enable) dir

-#- -length- -----date/time------ name
  4  3134688 Apr 27 1998 08:27:01 cat6000-sup.5-2-1.bin
  5  3231989 Jun 24 1998 12:04:40 cat6000-sup.5-2-1.bin
  6      135 Jul 17 1998 11:30:05 dns_config.cfg

1213952 bytes available (6388224 bytes used)
Console> (enable)
```

Verifying a File Checksum

To verify the checksum of a file on a Flash device, perform this task in privileged mode:

Table A-7: Verifying a File Checksum

Task	Command
Verify the checksum of a file on a Flash device.	**verify** [[*m*/]*device*:] *filename*

This example shows how to verify the checksum of a file:

```
Console> (enable) verify cat6000-sup.5-2-1-CSX.bin

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCC
File bootflash:cat6000-sup.5-2-1-CSX.bin verified OK
Console> (enable)
```

Formatting a Flash Device

Before you use a new Flash device, you must format it. You can reserve up to 16 spare sectors for use when other sectors fail (by default, none are reserved). If you do not reserve spare sectors and later some sectors fail, you will have to reformat the entire Flash memory, erasing all existing data.

Note Flash PC cards formatted on Supervisor Engine 1 or on a route-switch processor (RSP)-based Cisco 7500 series router are interchangeable if the router is running software at least at the same level as the supervisor engine. You cannot use Flash PC cards formatted on a route processor (RP)-based Cisco 7000 series router without reformatting.

When you format a Flash device, you can specify the *monlib* file (the ROM monitor library), which the ROM monitor uses to access files in the Flash file system. The monlib file is also compiled into the software image.

In the **format** command syntax, use the *device2* argument to specify the device that contains the monlib file to use. If you omit the entire *device2* argument, the switch formats the device using the monlib file that is bundled with the software. If you omit just the device name (*device2*) from the [[*device2:*][*monlib-filename*]] argument, the switch formats the device using the named monlib file from the default Flash device. If you omit the *monlib-filename* from the [[*device2:*][*monlib-filename*]] argument, the switch formats the device using the monlib file from *device2*. If you specify the entire [[*device2:*][*monlib-filename*]] argument, the switch formats the device using the specified monlib file from the specified device. If the switch cannot find a monlib file, it terminates the formatting process.

Note If the Flash device has a volume ID, you must provide the volume ID to format the device. The volume ID is displayed using the **show flash** *m/device*: **filesys** command.

To format a Flash device, perform this task in privileged mode:

Table A-8: Formatting a Flash Device

Task	Command
Format a Flash device.	**format [spare** *spare-number*] [*m*/]*device1*: [[*device2*:] [*monlib-filename*]]

This example shows how to format the Flash device in slot0:

```
Console> (enable) format slot0:

All sectors will be erased, proceed (y/n) [n]?y

Enter volume id (up to 31 characters):
Formatting sector 1
Format device slot0 completed.
Console> (enable)
```

Note Supervisor Engine 2 and Supervisor Engine 1 do not support the same Flash PC card format. To use a Flash PC card with Supervisor Engine 2, format the card with Supervisor Engine 2. To use a Flash PC card with Supervisor Engine 1, format the card with Supervisor Engine 1.