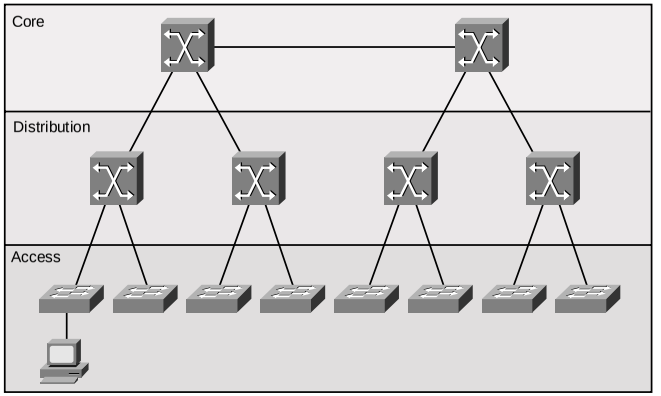
BCMSN Quick Reference Sheets

The Evolving Network Model

The Hierarchical Design Model

Cisco Systems has used the three-level *Hierarchical Design Model* for years. This older model provided a high-level idea of how a reliable network could be conceived, but was largely conceptual because it didn't provide specific guidance.



This simple drawing illustrates how the three-layer model might have been built out. A distribution Layer 3 switch would be used for each building on campus, tying together the access-switches on the floors. The core switches would link the various buildings together.

The layers break up a network in the following way:

- **Access layer**—End stations attach to VLANs.
  - —Clients attach to switch ports
  - —VLAN assigned/broadcast domains established
  - —Built using low-cost ports
  - —Simple quality of service (QoS) policies applied
- **Distribution layer**—Intermediate devices route and apply policies.
  - —VLANs terminated; routing is done between them
  - —Policies applied, such as
    - route selection
    - access lists
    - QoS
- **Core layer**—Backbone provides high-speed path between distribution elements.
  - —Distribution devices interconnected
  - —High speed (there's plenty of traffic)
  - —No policies (tough enough to keep up)

Later versions of this model showed redundant distribution and core devices and connections to make the model more fault tolerant. A set of distribution devices and their accompanying access layer switches were called a switch block.

Problems with the Hierarchical Design Model

This early model was a good starting point, but it failed to address key issues, such as

- Where do wireless devices fit in?
- How should Internet access and security be provisioned?
- How should remote access, such as dialup or virtual private network (VPN), be accounted for?
- Where should workgroup and enterprise services be located?

Enterprise Composite Network Model

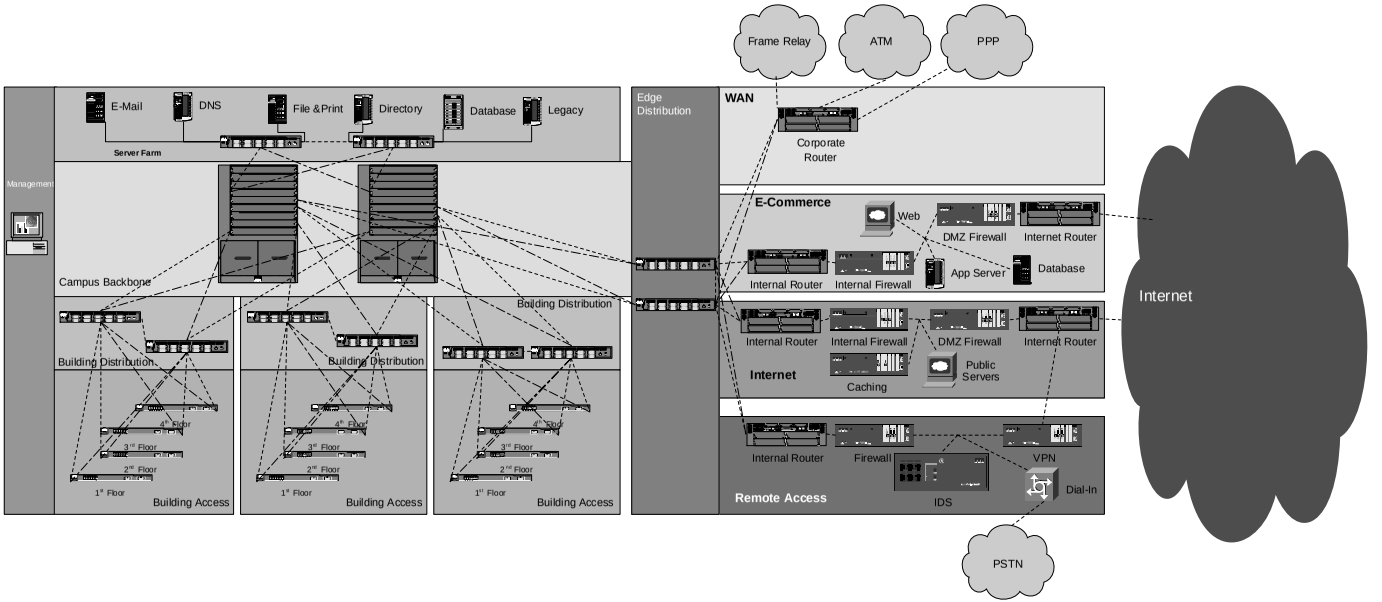
The Cisco newer model, the enterprise composite model, is significantly more complex and attempts to address the major shortcoming of the hierarchical model by expanding the older version and making specific recommendations about how and where certain network functions should be implemented. This model is based on the principles described in the Cisco Architecture for Voice, Video, and Integrated Data (AVVID).

The enterprise composite model is broken up into three large sections:

- Enterprise campus
- Enterprise edge
- Service provider edge—The different public networks that are attached

The first section, the enterprise campus, looks like the old hierarchical model with some added details. It features six sections:

1. Campus backbone—Like the old "core"
2. Building distribution
3. Building access
4. Management
5. Edge distribution—A distribution layer out to the WAN
6. Server farm—For enterprise services



The enterprise edge details the connections from the campus to the wide area and includes the following:

1. E-commerce
2. Internet connectivity
3. Remote access—Dial and VPN
4. Wide-area network (WAN)—Internal links

The service provider edge is just a list of the public networks that facilitate wide-area connectivity:

1. Internet service provider (ISP)
2. Public Switched Telephone Network (PSTN)—Dialup
3. Frame Relay, Asynchronous Transfer Mode (ATM), and Point-to-Point Protocol (PPP)—Private connectivity

Multilayer Switching

Comparing Devices

Layer 2 switches

- MAC address learning
- Hardware-based bridge
- Forwarding/filtering based on MAC address
- Spanning tree to avoid loops
- Wire speed, low latency
- Scalable

Routers

- Understand network structure
- Forward along best path based on Layer 3 address
- Can apply policies to traffic
- Security
- Quality of service
- Routing
- Lower speed, higher latency

Layer 3 Switches

- Hardware-based routing
- Provide flow accounting
- Understand network structure
- Forward along best path
- Can apply policies to traffic
- Security
- Quality of service
- Routing
- Wire speed, low latency

Comparing Ethernet Versions

All versions of Ethernet have features in common:

- Same frame definition and field values
- Same MAC address structure

Ethernet

- 10 Mbps using Manchester encoding, half or full duplex
- Links extend 100 m, typically on CAT-5 cable
- Not typically deployed today

Fast Ethernet

- 100 Mbps using 4B5B encoding, half or full duplex
- Links extend 100 m on CAT-5 or CAT-6 cable
- Used for client attachment today

Gigabit Ethernet

- 1000 Mbps (1 Gbps) using 8B10B encoding, full duplex
- 1000Base-T supports 100 m links using CAT-5 or CAT-6 cable
- 1000Base-SX supports 550 m links using multimode fiber
- 1000Base-LX supports 10 km links using single-mode fiber
- Used to aggregate traffic to distribution or core switches today

10Gigabit Ethernet

- 10,000 Mbps (10Gbps), full duplex only
- Supports multimode (less than 300 m) and single-mode fiber (up to 40 km)
- Not common today; sometimes used to aggregate traffic in backbone

Long Range Ethernet

- 5–15 Mbps
- Links use very high data rate digital subscriber line (VDSL) modulation to extend 500 feet on CAT-1/2/3
- Used to provide broadband in multi-unit dwellings (apartments, office buildings, hotels)

Metro Ethernet

- Uses "dark fiber" or service provider
- Ethernet principles extended into metropolitan-area network (MAN)

Switching Roles in the Enterprise Composite Model

- Building Access—Typically Layer 2 switches
- Building Distribution—Typically Layer 3 switches
- Campus Backbone—Layer 2 switches if no Layer 3 capabilities required
- Server Farm—Usually Layer 3 switches at access and distribution

Catalyst Switch Basics

CatOS Versus IOS

CatOS

- Layer 2 switching
- Can use MSFC with IOS for Layer 3 (multilayer switching)functionality
- Found on Catalyst 4000 and 6500 (optional)

IOS

- Layer 2 and 3 switching
- Ports can be "routed" or "switched"
- Found on Catalyst 2950 (Layer 2 only), 3550, 4000, and 6500 (optional)

Saving Catalyst Files

- Trivial File Transfer Protocol (TFTP)
  - —To copy IOS to TFTP: **copy flash tftp**
  - —To copy IOS from TFTP: **copy tftp flash**
  - —Verify Flash contents: **show flash**
  - —To save current configuration to NVRAM: **copy run start**
  - —To save current configuration to TFTP: **copy run tftp**

IOS Troubleshooting

Show

- Provides snapshots of device performance
- Low overhead
- Information organized

Debug

- Provides real-time display of device performance
- High overhead
- Uses **show processes** to see processor utilization
- Information not organized
- Uses **service timestamps debug datetime msec** to see event times
- Focuses debugging to minimize impact

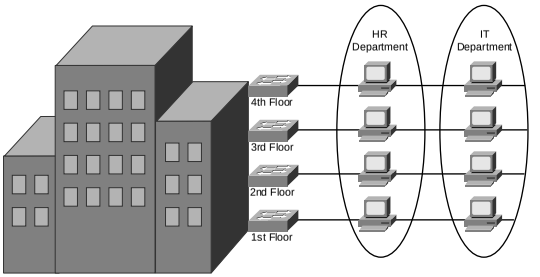
VLAN Implementation

What Is a VLAN?

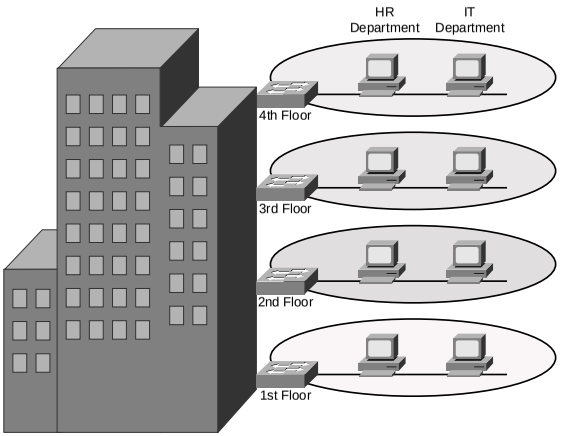
A VLAN is a logical LAN or a logical subnet. It defines a broadcast domain. A physical subnet is a group of devices sharing the same physical wire. A logical subnet is a group of switch ports assigned to the same VLAN, regardless of their physical location in a switched network.

Two types of VLANs are

- **End-to-end VLAN**—Hosts in the VLAN reside on several different switches and are scattered throughout the network. Used when hosts are assigned to VLANs based on functions or workgroups, rather than physical location. VLANs should not extend past the Building Distribution submodule.



- **Geographic (local) VLAN**—Hosts are assigned to VLANs based on their location, such as a floor in a building. A router accomplishes sharing of resources between VLANs. This type is typically found in the Building Access submodule.



VLAN membership can be assigned either statically by port, or dynamically by MAC address using a VLAN Membership Policy Server (VMPS).

Creating a VLAN in Global Config Mode

```
(config)#vlan 12
(config-vlan)#name MYVLAN
```

Creating a VLAN in Database Mode

```
#vlan database
(vlan)#vlan 12 name MYVLAN
```

Delete a VLAN by using the same command with **no** in front of it. You do not need to include the name when deleting.

Assigning Ports to VLANs

When statically assigning ports to VLANs, first make it an access port and then assign the port to a VLAN. At the interface configuration prompt, type

```
switchport mode access
switchport access vlan 12
```

To use dynamic VLAN assignment, the commands are similar. At interface configuration mode, type

```
switchport mode access
switchport access vlan dynamic
```

If using dynamic, you must also enter the IP address of the VMPS server at global configuration mode:

```
vmps server ip address
```

Verifying VLAN Configuration

To see a list of all the VLANs and the ports assigned to them, use the command **show vlan**. To narrow down the information displayed, you can use these keywords after the command: **brief, id,** *vlan-number*, or **name** *vlan-name*.

```
ASW# show vlan brief
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3,
                                                Fa0/10,Fa0/11,Fa0/12
20   VLAN0020                         active    Fa0/5,Fa0/6,Fa0/7
21   VLAN0021                         active    Fa0/8,Fa0/9
1002 fddi-default                     active
1003 trcrf-default                    active
1004 fddinet-default                  active
1005 trbrf-default                    active
```

Other verification commands include the following:

- **show running-config interface** *interface no.*—Use to verify the VLAN membership of the port:

```
ASW# show run interface fa0/5
Building configuration...
Current configuration 64 bytes
interface FastEthernet 0/5
 switchport access vlan 20
 switchport mode access
```

- **show mac address-table interface** *interface no.* **vlan** *vlan no.*—Use to view MAC addresses learned through that port for the specified VLAN:

```
ASW# show mac address-table interface fa0/1
          Mac Address Table
-------------------------------------------
Vlan    Mac Address     Type            Ports
----    -----------     ----            -----
1       0030.b656.7c3d  DYNAMIC         Fa0/1
Total Mac Addresses for this criterion: 1
```

- **show interfaces** *interface no.* **switchport**—Use to see detailed information about the port configuration, such as entries in the Administrative Mode and Access Mode VLAN fields:

```
ASW# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: false
Unknown unicast blocked: false
Unknown multicast blocked: false
Broadcast Suppression Level: 100
Multicast Suppression Level: 100
Unicast Suppression Level: 100
```

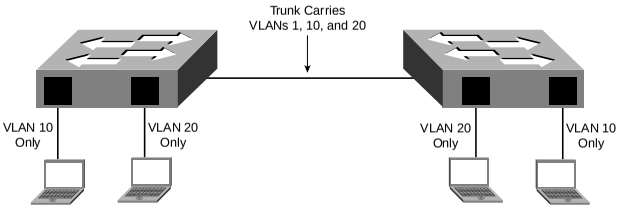
Troubleshooting VLAN Issues

The three steps in troubleshooting VLAN problems are

1. **Check the physical connectivity**—Make sure the cable is good and the network adapter and switch port are both good. Check the port's link LED.
2. **Check the switch configuration**—If you see frame check sequence (FCS) errors or late collisions, suspect a duplex mismatch. Also check configured speed on both ends of the link. Increasing collisions might mean an overloaded link, such as with a broadcast storm.
3. **Check the VLAN configuration**—If two hosts can't communicate, make sure they are both in the same VLAN. If a host can't connect to a switch, make sure the host and the switch are in the same VLAN.

VLAN Trunking

A *trunk* is a link that carries traffic for more than one VLAN. Trunks multiplex traffic from multiple VLANs. Trunks connect switches and allow ports on multiple switches to be assigned to the same VLAN.



Two methods of identifying VLANs over trunk links are

- **ISL (Inter-Switch Link)**—Cisco proprietary; encapsulates the original frame in a header that contains VLAN information. Is protocol-independent; can identify Cisco Discovery Protocol (CDP) and bridge protocol data unit (BPDU) frames.
- **802.1Q**—Standards-based; tags the frames (inserts a field into the original frame immediately after the source MAC address field); supports Ethernet and Token Ring networks.

When a frame comes into a switch port, the frame is tagged internally within the switch with the VLAN number of the port. When it reaches the outgoing port, that internal tag is removed. If the exit port is a trunk port, then its VLAN is identified either in the ISL encapsulation or the 802.1Q tag. The switch on the other end of the trunk removes the ISL or 802.1Q information, checks the VLAN of the frame, and adds the internal tag. If the exit port is a user port, then the original frame is sent out unchanged, making the use of VLANs transparent to the user.

If a non-trunking port receives an ISL-encapsulated frame, the frame is dropped. Also, if the ISL header and footer cause the MTU size to be exceeded, it might be counted as an error.

If a non-trunking port receives an 802.1Q frame, the source and destination MAC addresses are read, the tag field is ignored, and the frame is switched normally at Layer 2.

Configuring a Trunk Link

Ports can become trunk ports either by static configuration or dynamic negotiation using Dynamic Trunking Protocol (DTP). A switch port can be in one of five DTP modes:

- **Access**—The port is a user port and cannot be a trunk.
- **Trunk**—The port is a trunk and negotiates trunking with the port on the other end of the link.

- **Nonnegotiate**—When this keyword is added, the port is a trunk and does not do DTP negotiation with the other side of the link.
- **Dynamic Desirable**—Actively negotiates trunking with the other side of the link. Becomes a trunk if the port on the other switch is set to **trunk**, **dynamic desirable**, or **dynamic auto** mode.
- **Dynamic Auto**—Passively waits to be contacted by the other switch. Becomes a trunk if the other end is set to **trunk** or **dynamic desirable** mode.

Configure a port for trunking at the interface configuration mode:

```
switchport mode {dynamic {auto | desirable} | trunk}
```

If dynamic mode is used, DTP negotiates trunking state and encapsulation. If trunk mode is used, you must specify encapsulation:

```
switchport trunk  encapsulation {isl | dot1q | negotiate}
```

Native VLAN with 802.1Q

If you are using 802.1Q, you must specify a native VLAN for the trunk link with this command:

```
switchport trunk native vlan vlan no.
```

Frames from the native VLAN are sent over the trunk link untagged. It is the VLAN the port would be in if it were not a trunk and must match on both sides of the trunk link. VLAN 1 is the default native VLAN for all ports.

VLAN Mapping

ISL trunking recognizes only VLANs numbered 1–1001, but 802.1Q can use VLANs 0–4094. If you are using both ISL and 802.1Q in your network and have VLANs numbered above 1001, you have to map the 802.1Q VLANs to ISL numbers. Some rules about mapping VLANs are as follows:

- You can configure only eight mappings.
- Mappings are local to the switch—The same mappings must be configured on all switches in the network.
- You can map only to Ethernet ISL VLANs.
- The 802.1Q VLANs with the same number as mapped ISL VLANs are blocked. (For example, you map 802.1Q VLAN 1500 to ISL VLAN 150, and then 802.1Q VLAN 150 is blocked on that switch.)
- Don't map the 802.1Q native VLAN.

VLANs Allowed on the Trunk

By default, a trunk carries traffic for all VLANs. You can change that behavior for a particular trunk link by giving the following command at the interface config mode:

```
switchport trunk allowed vlan vlans
```

Make sure that both sides of a trunk link allow the same VLANs.

Verifying a Trunk Link

You can use two commands to verify your trunk configuration:

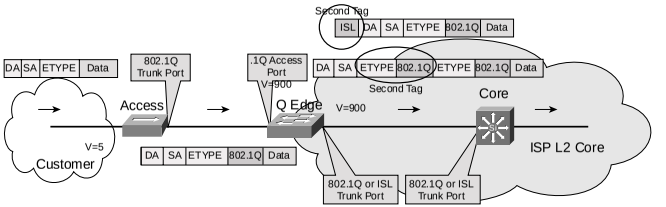
```
show running-config
show interfaces [interface no.] switchport | trunk
```

Using the **trunk** keyword with the **show interfaces** command gives information about the trunk link:

```
ASW# show interfaces fastethernet 0/1 trunk
Port       Mode         Encapsulation  Status        Native vlan
Fa0/1    desirable      n-802.1q         trunking      1
Port      Vlans allowed on trunk
Fa0/1     1-150
<further output omitted>
```

802.1Q Tunnels

Tunneling is a way to send 802.1Q-tagged frames across a foreign network (such as a service provider's network) and still preserve the original 802.1Q tag. The service provider (SP) configures its end of the trunk link as a tunnel port and assigns a VLAN to carry your traffic within its network. The SP switch then adds a second 802.1Q tag to each frame that came in the tunnel port. Other switches in the SP network see only this second tag, and don't read the original tag. When the frame exits the SP network, the extra tag is removed, leaving the original 802.1Q tag to be read by the receiving switch in your network.



Layer 2 Protocol Tunneling

If a service provider separates sections of your network, you can use Layer 2 protocol tunneling to tunnel CDP, Spanning Tree Protocol (STP), and VLAN Trunking Protocol (VTP) frames across the SP's cloud. This is called Generic Bridge PDU Tunneling (GBPT). Frames from the previously mentioned control protocols are encapsulated as they enter the SP's network on a tunnel port, and de-encapsulated when they exit that network.

Troubleshooting Trunking

- Both sides of the link in the correct trunking mode?
- Same trunk encapsulation on both sides?
- If 802.1Q, same native VLAN on both sides?

VLAN Trunking Protocol

VLAN Trunking Protocol (VTP) runs over trunk links and synchronizes the VLAN databases of all switches in the VTP domain. A VTP domain is an administrative group. All switches within that group must have the same VTP domain name configured, or they will not synchronize databases.

VTP works by using configuration revision numbers and VTP advertisements:

- All switches send out VTP advertisements every five minutes or when a change to the VLAN database happens (a VLAN is created, deleted, or renamed.)
- VTP advertisements contain a configuration revision number. This number is increased by 1 for every VLAN change.
- When a switch receives a VTP advertisement, it compares the configuration revision number against the one in its VLAN database.
- If the new number is higher, the switch overwrites its database with the new VLAN information and forwards the information to its neighbor switches.
- If the number is the same, the switch ignores the advertisement.
- If the new number is lower, the switch replies with the more up-to-date information contained in its own database.

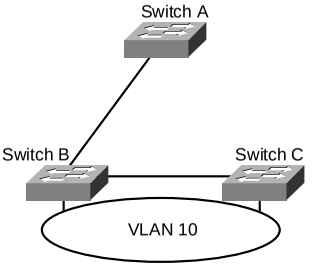
VTP Switch Roles

A VTP switch can be

- **A server**—The default. Servers can create, delete, and rename VLANs. They originate both periodic and triggered VTP advertisements and synchronize their databases with other switches in the domain.
- **A client**—Clients cannot make VLAN changes. They originate periodic VTP advertisements and synchronize their databases with other switches in the domain.
- **Transparent**—Can create, delete, and rename VLANs, but its VLANs are local only. Does not originate advertisements; does not synchronize its database with any other switches. It forwards VTP advertisements out its trunk links, however.

VTP Pruning

Recall that, by default, switches flood broadcasts, multicasts, and unknown unicasts across trunk links. Suppose a host in VLAN 10 on Switch B sends a broadcast. Hosts in VLAN 10 on Switch C need to see that broadcast, but Switch A has no ports in VLAN 10, so it just drops the broadcast traffic.



Enabling VTP Pruning causes the switch to keep track of VLAN port assignments in its downstream switches. The switch then sends only flooded traffic on trunks toward switches that have ports assigned to the VLAN originating the traffic. It prunes flooded traffic from all other trunks. VTP Pruning increases the available bandwidth by preventing unnecessary traffic on trunk links.

Two versions of VTP exist—Version 1 and Version 2. To use Version 2, all switches in the domain must be capable of using it. Configure one server for Version 2, and the information is propagated through VTP. Version 2 has the following added features:

- It supports Token Ring VLANs.
- Transparent switches pass along messages from both versions of VTP.
- Consistency checks are performed only when changes are configured through the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

Configuring VTP

VTP configuration is done at the global config mode. To configure the switch's VTP mode, type

```
vtp {server | client |transparent}
```

To configure the VTP domain name, type

```
vtp domain name
```

To configure a VTP password (all switches in the domain must use the same password), type

```
vtp password password
```

To configure the switch to use VTP Version 2, type

```
vtp v2-mode
```

To enable pruning, type

```
vtp pruning
```

To specify which VLANs are to be pruned, type

```
switchport trunk pruning vlan {add | except | none | remove} vlan-list
  [,vlan[,vlan[,,,]]
```

Verifying and Monitoring VTP

To get basic information about the VTP configuration, use **show vtp status**.
The following example shows the default settings:

```
ASW# show vtp status
VTP Version                     : 1
Configuration Revision          : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 5
VTP Operating Mode              : Server
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      :
```

Troubleshooting VTP

Here are some common items to check when troubleshooting problems with VTP:

- Make sure you are trunking between the switches. VTP is sent only over trunk links.
- Make sure the domain name matches on both switches (name is case sensitive).
- If the switch is not updating its database, make sure it is not in transparent mode.
- If using passwords, make sure they all match. To remove a password, use **no vtp password**.

Adding a New Switch to a VTP Domain

Adding a new switch in client mode does not prevent it from propagating its incorrect VLAN information. A server synchronizes to a client if the client has the higher configuration revision number. You must reset the revision number back to 0 on the new switch. The easiest way to do this is to change the domain name. Then, change it back to the correct one and attach the switch to the network.

Understanding the Spanning Tree Protocol

Switches either forward or filter Layer 2 frames. The way they make the forwarding/filtering decision can lead to loops in a network with redundant links. Spanning tree is a protocol that detects potential loops and breaks them.

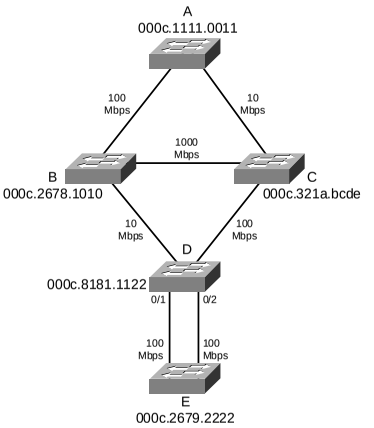
A Layer 2 switch is functionally the same thing as a transparent bridge. Transparent bridges

- Learn MAC addresses by looking at the source address of incoming frames. They build a table mapping MAC address to port number.
- Forward broadcasts and multicasts out all ports except the on they came in on. (This is called *flooding*.)
- Forward unknown unicasts out all ports except the one they came in on. An unknown unicast is a message bound for a unicast MAC address that is not in the switch's table of addresses and ports.
- Do not make any changes to the frames as they forward them.

Spanning Tree Protocol (STP) works by selecting a root bridge, then selecting one loop-free path from the root bridge to every other switch. (STP uses the term *bridge* because it was written before there were switches.) Consider the following switched network.

Spanning tree must select:

- One root bridge
- One root port per non-root bridge
- One designated port per network segment



Spanning Tree Election Criteria

1. Lowest Root Bridge ID (BID)
2. Lowest path cost to the root
3. Lowest sender Bridge ID
4. Lowest sender Port ID (PID)

Bridge ID = Bridge priority : Bridge MAC address

Bridge priority = A 2-byte value, 0–65,535 (0–FFFF hex).

Default priority is 32,768 (8000 hex)

Port ID = Port priority : port number

Port Priority = A 6-bit value, 0–63, default is 32

Path cost—Cumulative value of the cost of each link between the bridge and the root. An old way of calculating cost and a new way of calculating cost exists:

Link Speed	Old Cost	New Cost
10 Mbps	100	100
100 Mbps	10	19
1 Gbps	1	4
10 Gbps	1	2

The STP Election

Root Bridge Election

Looking at the example, first select the root bridge. Assume each switch is using the default priority.

- Switch A BID = 80-00-00-0c-11-11-00-11
- Switch B BID = 80-00-00-0c-26-78-10-10
- Switch C BID = 80-00-00-0c-32-1a-bc-de
- Switch D BID = 80-00-00-0c-81-81-11-22
- Switch E BID = 80-00-00-0c-26-79-22-22

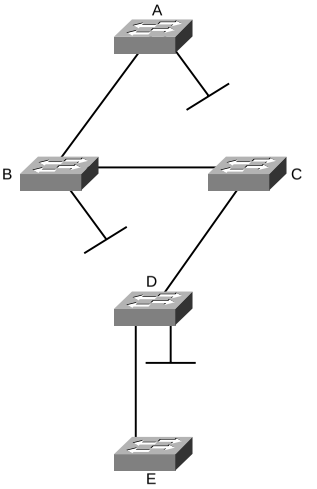
Switch A has the lowest BID, so it is elected the root. Each non-root switch must now elect a root port.

Root Port Election

- **Switch B**—Uses the connected link to A, path cost of 19 (link speed of 100 Mbps).
- **Switch C**—The connected link has a path cost of 100 (link speed of 10 Mbps), the link through B has a path cost of 38 (two 100 Mbps links), and so that port is chosen.
- **Switch D**—The link through B has a path cost of 119, the path cost through C to A is 119, the path through C then B is 57, so that port is chosen.
- **Switch E**—The lowest path cost is the same for both ports (76—through D to C to B to A). Next, check sender BID. Sender for both ports is D, so that doesn't break the tie. Next, check sender Port ID. Assuming default port priority, the PID for 0/1 is lower than the PID for 0/2, so the port on the left is the root port.

Designated Port Election

- The root bridge is the designated bridge for every segment connected to it (A–B and A–C in our example).
- **Segment B–D**—B has the lowest path cost to root (19 versus 119), so it is designated for this segment.
- **Segment C–D**—C has the lowest path cost to the root (100 versus 119), so it is designated for this segment.
- **Segment B–C**—B has the lowest path cost to the root (19 versus 100), so it is designated for this segment.
- **Both segments D–E**—D has the lowest path cost to the root (57 versus 76), so it is designated for both segments.



Now the looped topology has been turned into a tree, with A at the root. Notice that no more redundant links exist.

Bridge Protocol Data Units

Switches exchange Bridge Protocol Data Units (BPDUs). Two types of BPDUs exist: configuration and topology change.

Configuration BPDUs are sent every two seconds from the root towards downstream switches. They are used during an election, maintain connectivity between switches, and send timer information from the root.

Topology Change (TCN) BPDUs are sent towards the root when

- There is a link failure.
- A port starts forwarding, and there is already a designated port.
- The switch receives a TCN from a neighbor.

When a switch receives a TCN BPDU, it acknowledges that with a configuration BPDU that has the Topology Change Acknowledgment bit set.

When the root bridge receives a TCN, it starts sending configuration BPDUs with the Topology Change bit set for a period of time equal to Max Age plus Forward Delay. Switches that receive this change their MAC table Aging Time to the Forward Delay time, causing MAC addresses to age out faster. The topology change also causes a new election of the root bridge, root ports, and designated ports.

BPDU Fields

Some of the fields in the BPDU include the following:

- Root Bridge ID
- Sender's Root Path Cost
- Sender's Bridge ID
- Sender's Port ID
- Message Age
- Hello time—2 sec by default
- Forward Delay—15 sec by default
- Max Age—20 sec by default

Spanning Tree Port States

When a port is first activated, it transitions through the following stages:

Port State	Timer	Actions
Blocking	Max Age (20 sec)	Discards frames, does not learn MAC addresses, does receive BPDUs
Listening	Forward Delay (15 sec)	Discards frames, does not learn MAC addresses, receives BPDUs to determine its role in the network
Learning	Forward Delay (15 sec)	Discards frames, does learn MAC addresses, does receive and transmit BPDUs
Forwarding		Accepts frames, learns MAC addresses, receives and transmits BPDUs

Designing for Spanning Tree

To optimize data flow in the network, design and configure switches for the following STP roles:

- Primary and secondary root bridges (set priority values)
- Designated and root ports (set port priorities/path cost)
- Enable STP enhancements such as Root Guard

Spanning Tree and PVST

With PVST (per-VLAN spanning tree), a different instance of STP exists for each VLAN. To derive the VLAN BID, the switch picks a different MAC address from its base pool for each VLAN. Each VLAN has its own root bridge, root port, etc. You can configure these so that data flow is optimized, and traffic load is balanced among the switches.

Spanning tree is enabled by default on every VLAN.

Configuring Spanning Tree

To change the STP priority value, type

```
Switch(config)# spanning-tree vlan vlan no. priority value
```

To configure a switch as root without manually changing priority values, type

```
Switch(config)# spanning-tree vlan vlan no. root {primary | secondary}
```

To change the STP port cost for an access port, type

```
Switch(config-if)# spanning-tree cost value
```

To change the STP port cost for a VLAN on a trunk port, type

```
Switch(config-if)# spanning-tree vlan vlan no. cost value
```

To display the STP information for a particular VLAN, type

```
Switch# show spanning-tree vlan vlan no.
```

To display the STP information for an interface, type

```
Switch # show spanning-tree interface interface no. [detail]
```

To verify STP timers, type

```
#show spanning-tree bridge brief
```

Spanning Tree Enhancements

Cisco has some proprietary enhancements to spanning tree that help speed up network convergence. They include the following:

- Port Fast
- Uplink Fast
- Backbone Fast

Port Fast

Port Fast is for access (user) ports only. It causes the port to bypass the STP listening and learning states, and transition directly to forwarding. If a BPDU is received, Port Fast is abandoned, the port placed in blocking, and the switch runs through the entire Spanning Tree procedure.

```
(config-if)# spanning-tree portfast
```

Uplink Fast

Uplink Fast is for speeding convergence when a direct link to an upstream switch fails. The switch identifies backup ports for the root port (these are called an *uplink group*). If the root port fails, one of the ports in the uplink group is unblocked and transitions immediately to forwarding—bypassing the listening and learning stages. It should be used in wiring closet switches with at least one blocked port:

```
(config)# spanning-tree uplinkfast
```

Backbone Fast

Backbone Fast is used for speeding convergence when a link fails that is not directly connected to the switch. It helps the switch detect indirect failures. If a switch running Backbone Fast receives an inferior BPDU from its designated bridge, it knows a link on the path to the root has failed. (An inferior BPDU is one that lists the same switch for root bridge and designated bridge.)

The switch then tries to find an alternate path to the root by sending a Root Link Query (RLQ) protocol data unit (PDU) out all alternate ports. The root then responds with a RLQ response, and the port receiving this response can transition to forwarding. Alternate ports are determined in this way:

- If the inferior BPDU was received on a blocked port, the root port and any other blocked ports are considered alternates.
- If the inferior BPDU was received on the root port, all blocked ports are considered alternates.
- If the inferior BPDU was received on the root port and there are no blocked ports, the switch assumes it has lost connectivity with the root and advertises itself as root.

Configure this command on all switches in the network:

```
(config)#spanning-tree backbonefast
```

Rapid Spanning Tree Protocol

Rapid Spanning Tree Protocol (RSTP)—802.1w—is a standards-based, non-proprietary way of speeding STP convergence. Switch ports exchange an explicit handshake when they transition to forwarding. RSTP describes different port states than regular STP, as shown in the following table.

STP Port State	Equivalent RSTP Port State
Disabled	Discarding
Blocking	Discarding
Listening	Discarding
Learning	Learning
Forwarding	Forwarding

RSTP Port Roles

RSTP also defines different spanning-tree roles for ports:

- **Root port**—The best path to the root (same as STP)
- **Designated port**—Same role as with STP
- **Alternate port**—A backup to the root port
- **Backup port**—A backup to the designated port
- **Disabled port**—One not used in the spanning tree
- **Edge port**—One connected only to an end user

BPDU Differences in RSTP

In regular STP, BPDUs are originated by the root and relayed by each switch. In RSTP, each switch originates BPDUs, whether or not it receives a BPDU on its root port. All 8 bits of the BPDU type field are used by RSTP. The TC and TC Ack bits are still used; the other 6 bits specify the port's role and its RSTP state and are used in the port handshake. The RSTP BPDU is set to Type 2, Version 2. PVST is done by Rapid per-VLAN spanning tree plus (PVST+) on Catalyst switches.

RSTP Fast Convergence

- RSTP uses a mechanism similar to Backbone Fast—When an inferior BPDU is received, the switch accepts it. If the switch has another path the root, it uses that and informs its downstream switch of the alternate path.
- Edge ports work the same as Port Fast ports—They automatically transition directly to forwarding.
- Link type—If you connect two switches through a point-to-point link and the local port becomes a designated port, it exchanges a handshake with the other port to quickly transition to forwarding. Full-duplex links are assumed to be point-to-point; half-duplex links are assumed to be shared.
- Also, backup and alternate ports can transition to forwarding when no BPDUs are received from a neighbor switch (similar to Uplink Fast).

If an RSTP switch detects a topology change, it sets a TC timer to twice the hello time and sets the TC bit on all BPDUs sent out its designated and root ports until the timer expires. It also clears the MAC addresses learned on these ports.

If an RSTP switch receives a TC BPDU, it clears the MAC addresses on that port and sets the TC bit on all BPDUs sent out its designated and root ports until the TC timer expires.

Multiple Spanning Tree (MST)

With MST, you can group VLANs and run just one instance of spanning tree for a group of VLANs. This cuts down on the number of BPDUs in your network. Switches in the same MST Region share the same configuration and VLAN mappings. Configure MST with these commands:

```
(config)# spanning-tree mode mst
(config)# spanning-tree mst configuration
(config-mst)# name region_name
(config-mst)# revision number
(config-mst)# instance number vlan vlan range
(config-mst)# end
```

To be compatible with 802.1Q trunking, which has one Common Spanning Tree (CST) for all VLANs, MST runs one instance of an Internal Spanning Tree (IST). The IST appears as one bridge to a CST area and is MST instance number 0. The original MST spanning trees (called M-Trees) are active only within the region—they combine at the edge of the CST area to form one.

EtherChannel

EtherChannel is a way of combining several physical links between switches into one logical connection. Normally, spanning tree would block redundant links. EtherChannel gets around that and allows load balancing across those links. Load balancing is done based on such things as source or destination MAC address or IP address. At global config mode, type

```
port-channel load-balance type
```

A logical interface—the Port Channel interface—is created. Configuration can be applied both to the logical and physical interfaces.

Here are some guidelines for EtherChannel:

- Interfaces in the channel do not have to be physically next to each other or on the same module.
- All ports must be the same speed and duplex.
- All ports in the EtherChannel bundle should be enabled.
- None of the bundle ports can be a Switch Port Analyzer (SPAN) port.
- Assign an IP address to the logical Port Channel interface, not the physical ones.
- Put all bundle ports in the same VLAN, or make them all trunks. If they are trunks, they must all carry the same VLANs and use the same trunking mode.
- Configuration you apply to the Port Channel interface affects the entire EtherChannel. Configuration you apply to a physical interface affects only that interface.

Configuring an EtherChannel

Basically, for a Layer 3 EtherChannel, configure the logical interface and then put the physical interfaces into the channel group:

```
interface port-channel number
no switchport
ip address address mask
```

Then, at each port that is part of the EtherChannel:

```
interface { number | range interface - interface}
channel-group number mode {auto | desirable | on}
```

Putting the IP address on the Port Channel interface creates a Layer 3 EtherChannel. Simply putting interfaces into a channel group creates a Layer 2 EtherChannel, and the logical interface is automatically created.

The Cisco proprietary Port Aggregation Protocol (PAgP) dynamically negotiates the formation of a channel. Three PAgP modes exist:

- **On**—The port channels without using PAgP negotiation. The port on the other side must also be set to On.
- **Auto**—Responds to PAgP messages but does not initiate them. Port channels if the port on the other end is set to Desirable. This is the default mode.
- **Desirable**—Port actively negotiates channeling status with the interface on the other end of the link. Port channels if other side is Auto or Desirable.

Also, a non-proprietary protocol called Link Aggregation Control Protocol (LACP), IEEE 802.3ad, does the same thing. LACP has two modes:

- **Active**—Port actively negotiates channeling with the port on the other end of the link. Channel forms if other side is passive or active.
- **Passive**—Responds to LACP messages but does not initiate them. Channel forms if other end is set to active.

If you want to use LACP, specify it under the interface and put the interface in either active or passive mode.

```
channel-protocol lacp
```

Verifying an EtherChannel

Here are some typical commands for verifying an EtherChannel:

- **show running-config interface** *number*
- **show interfaces** *number* **etherchannel**
- **show etherchannel** *number* **port-channel**
- **show etherchannel summary**

Additional Spanning Tree Features

Some additional features available to help you tune spanning tree include

- BPDU Guard
- BPDU Filtering
- Root Guard
- Unidirectional Link Detection (UDLD)
- Loop Guard

BPDU Guard

BPDU Guard prevents loops if another switch is attached to a Port Fast port. When BPDU Guard is enabled on an interface, it is put into an error-disabled state (basically, shut down) if a BPDU is received on the interface. It can be enabled at either global config mode—in which case it affects all Port Fast interfaces—or at interface mode. Port Fast does not have to be enabled for it to be configured at a specific interface.

```
(config)# spanning-tree portfast bpduguard default
(config-if)# spanning-tree bpduguard enable
```

BPDU Filtering

BPDU filtering is another way of preventing loops in the network. It also can be enabled either globally or at the interface and functions differently at each. In global config, if a Port Fast interface receives any BPDUs, it is taken out of Port Fast status. At interface config mode, it prevents the port from sending or receiving BPDUs. The commands are

```
(config)# spanning-tree portfast bpdufilter default
(config-if)# spanning-tree bpdufilter enable
```

Root Guard

Root Guard is meant to prevent the wrong switch from becoming the spanning-tree root. It is enabled on ports other than the root port, on switches other than the root. If a Root Guard port receives a BPDU that would cause it to become a root port, the port is put into "root-inconsistent" state and does not pass traffic through it. If the port stops receiving these BPDUs, it automatically re-enables itself.

```
(config-if)# spanning-tree guard root
```

Unidirectional Link Detection (UDLD)

A switch notices when a physical connection is broken, by the absence of Layer 1 electrical keepalives (Ethernet calls this a link beat). But sometimes, a cable is intact enough to maintain keepalives, but not to pass data in both directions. This is a unidirectional link. UDLD detects a unidirectional link by sending periodic hellos out the interface. It also uses probes, which must be acknowledged by the device on the other end of the link. UDLD operates at Layer 2. The port is shut down if a unidirectional link is found.

- To enable UDLD on all fiber-optic interfaces, use this command:

```
(config)# udld enable
```

Although this command is given at global config mode, it applies only to fiber ports.

- To enable UDLD on non-fiber ports, give the same command at interface config mode.
- To disable UDLD on a specific fiber port, use this command:
  `(config-if)# udld disable`
- To disable UDLD on a specific non-fiber port, use this command:
  `(config-if)#no udld enable`
- To re-enable all interfaces shut by UDLD:
  `#udld reset`
- To verify UDLD status:
  `#show udld interface`

Loop Guard

Loop Guard prevents loops that might develop if a port that should be blocking inadvertently transitions to the forwarding state. This can happen if the port stops receiving BPDUs (perhaps because of a unidirectional link or a software/configuration problem in its neighbor switch). When one of the ports in a physically redundant topology stops receiving BPDUs, the STP conceives the topology as loop-free. Eventually, the blocking port becomes designated, and moves to forwarding state, thus creating a loop. With Loop Guard enabled, an additional check is made.

If no BPDUs are received on a blocked port for a specific length of time, Loop Guard puts that port into "loop inconsistent" blocking state, rather than transitioning to forwarding state. Loop Guard should be enabled on all switch ports that have a chance of becoming root or designated ports. It is most effective when enabled in the entire switched network, in conjunction with UDLD.

To enable Loop Guard for all point-to-point links on the switch, use the following command:

```
(config)# spanning-tree loopguard default
```

To enable Loop Guard on a specific interface, type

```
(config-if)# spanning-tree guard loop
```

Loop Guard automatically re-enables the port if it starts receiving BPDUs once again.

Troubleshooting STP

Some common things to look for when troubleshooting Spanning Tree Protocol include

- **Duplex mismatch**—When one side of the link is half duplex and the other is full duplex. Causes late collisions and FCS errors.
- **Unidirectional link failure**—When the link is up but data only flows in one direction. Can cause loops.
- **Frame corruption**—Physical errors on the line cause BPDUs to be lost, and the port incorrectly begins forwarding. Caused by duplex mismatch, bad cable, or too long of cable.
- **Resource errors**—STP is implemented in software, so a switch with an overloaded CPU or memory can neglect some STP duties.
- **Port Fast configuration errors**—Connecting a switch to two ports that have Port Fast enabled. Can cause a loop.
- **STP tuning errors**—Max Age or Forward Delay set too short can cause a loop. Network diameter set too low causes BPDUs to be discarded and affect STP convergence.

Identifying a Bridging Loop

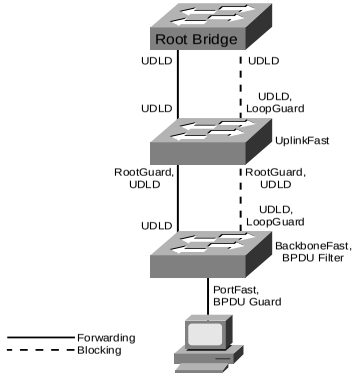
Suspect a loop if you see the following:

- You capture the traffic on the overloaded link and see the same frames multiple times. This signifies a loop.
- All users in one bridging domain have connectivity problems at the same time.
- An abnormally high activity exists when checking port utilization.

To remedy a loop quickly, shut redundant ports and then enable them one at a time. Some switches allow debugging of STP (not 3550/2950) to help in diagnosing problems.

What to Use Where

Confused by all the acronyms and STP features? The following diagram shows the STP features you might use in your network, and where you might use them.



Multilayer Switching

Understanding the Switching Process

Here are the steps involved in Layer 2 forwarding:

Input

1. Receive frame
2. Verify frame integrity
3. Apply inbound VLAN access control list (ACL)
4. Lookup destination MAC

Output

1. Apply outbound VLAN ACL
2. Apply outbound QoS ACL
3. Select output port
4. Queue on port
5. Rewrite
6. Forward

Here are the steps involved in Layer 3 forwarding:

Input

1. Receive frame
2. Verify frame integrity
3. Apply inbound VLAN ACL
4. Lookup destination MAC

Routing

1. Input ACL
2. Switch if entry cached
3. Identify exit interface and next-hop address using routing table
4. Output ACL

Output

1. Apply outbound VLAN ACL
2. Apply outbound QoS ACL
3. Select output port
4. Queue on port
5. Rewrite source and destination MAC, IP checksum, and frame check sequence (FCS); decrement Time to Live (TTL)
6. Forward

Understanding the Switching Table

Content Addressable Memory (CAM)

- Used for Catalyst 4000 Layer 2 forwarding tables
- Used for Catalyst 6500 Layer 2 and NetFlow forwarding tables
- Binary values (0 or 1)
- Match must be exact

Ternary Content Addressable Memory (TCAM)

- Used for Catalyst 3550, 4000, and 6500 Layer 3 switching
- Ternary (3) values (0, 1, or wildcard)
- Entries are in VMR form:
  —Value—Pattern to be matched
  —Mask—Masking bits associated with pattern
  —Result—Consequences of a match (permit/deny, or more complex information

Understanding Switch Forwarding Architectures

Centralized Forwarding

- Decision made by single table
- Used by 4000 and 6500

Distributed Forwarding

- Decision made at port or module
- Used by 3550 and 6500 with distributed forwarding card

NetFlow Switching

- Decision made cooperatively by route processor and Multilayer Switching (MLS)
- First packet switched in software; result cached
- Subsequent packets switched in hardware

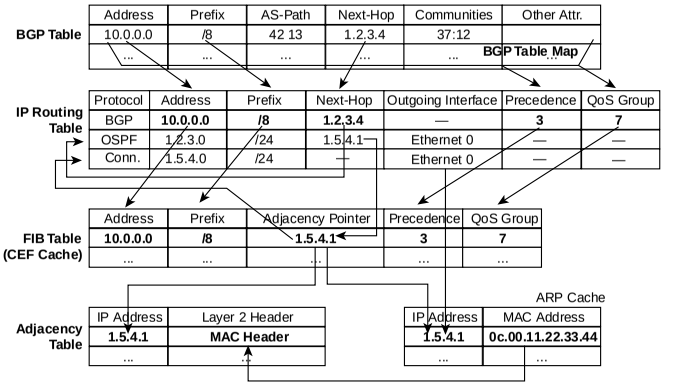
Cisco Express Forwarding (CEF)

- Topology based switching (via Forwarding Information Base [FIB])
- Can be centralized or distributed

Cisco Express Forwarding (CEF)

CEF does the following:

- Separates control plane hardware from data plane hardware
- Controls plane runs in software and builds Forwarding Information Base (FIB) and adjacency table
- The data plane uses hardware to forward most IP unicast traffic
- Handles traffic that must be forwarded in software (much slower) including:
  - —Packets originating from device
  - —Packets with IP header options
  - —Tunneled traffic
  - —802.3 (IPX) frames
- Supports load sharing
- FIB is an optimized routing table, stored in TCAM
- Builds adjacencies from Address Resolution Protocol (ARP) data



ARP throttling:

- First packet to destination forwarded to route processor
- Subsequent traffic dropped until MAC resolved
- Prevents overwhelming RP with redundant ARP requests
- Helps during denial-of-service attacks; removed when MAC resolved or in 2 seconds

Configuring and Troubleshooting CEF

By default, CEF is on and supports per destination load sharing.

To disable CEF:

- 4000: **no ip cef.**
- 3550: On each interface, use **no ip route-cache cef.**
- 6550 with Policy Feature Card, Distributed FC, and Multilayer Switch FC: Cannot be disabled.

View CEF information:

```
show interface fastethernet 2/2 | begin L3
```

View switching statistics:

```
show interface fastethernet 2/2 | include switched
```

View FIB: **show ip cef**
View detailed CEF FIB entry:

```
show ip cef fastethernet 2/2 10.0.0.1 detail
```

Troubleshoot CEF drops:

```
debug ip cef drops
```

Troubleshoot packets not forwarded by CEF:

```
debug ip cef receive
```

Troubleshoot CEF events:

```
debug ip cef events
```

Inter-VLAN Routing

Inter-VLAN Routing Using Multilayer Switches

Port roles:

- VLAN port—Acts as Layer 2 switching port with a VLAN
  - —Static VLAN—Use **switchport** command to identify VLAN
  - —Dynamic VLAN—Use VLAN Membership Policy Server (VMPS)
- Trunk port—Passes multiple VLANs and differentiates by tagging
  - —Use **switchport** command to set parameters
  - —ISL or 802.1Q
- Switch virtual interface (SVI)—Virtual routed port in a VLAN
  - —Use to route or fallback bridge between VLANs
  - —Default SVI for VLAN 1 automatically created
  - —Associate with VLAN using **interface vlan#**
- Routed port—Acts as Layer 3 routed port
  - —Place in Layer 3 mode with **no switchport**
  - —Not associated with VLAN
  - —Turn on routing using **ip routing**
  - —Assign address and enable routing protocols as needed

Inter-VLAN Routing

Multilayer switches do the following:

- Enable IP routing using **ip routing**
- Create SVI using **interface vlan#**
- Assign IP address to each interface

Router-on-a-stick—Attach router to switch using trunk line (ISL or 802.1Q):

- Easy to implement
- Uses existing equipment
- Much more latency than MLS solution
- Configure by creating subinterface—**interface fastethernet 1/0.7**
- Associate VLAN to interface with command **encapsulation isl 7** or **encapsulation dot1q 7**:
  - —ISL—No address on main interface
  - —802.1Q—Address on main interface for native (untagged) VLAN

Multilayer Switch Reliability

Equipment Goals

Network hardware should be reliable and fault-tolerant. The network should be optimized and every opportunity to implement redundancy scrutinized.

Opportunities to implement hardware redundancy:

- Supervisor card
  - —Second Supervisor provides backup without cost of new switch.
  - —Supervisor configuration and switchover maintained by Route Processor Redundancy (RPR) or Route Processor Redundancy Plus (RPR+).
  - —RPR—Redundant Supervisor boots, draws configuration, when main Supervisor dies. About 3 minutes failover.
  - —RPR+—Redundant Supervisor already booted and maintained in synchronized state. Both Supervisors must use same IOS. Failover in less than minute.
  - —FIB table blank at switchover.
  - —Enable with command **redundancy**.
  - —Identify redundancy protocol using **mode rpr-plus**.
  - —View settings with **show redundancy states**.
  - —Uplink ports for backup Supervisor are active.
  - —Split redundant links between Supervisors so that failure doesn't remove both links.

- Power supply
  —Some switch models allow for redundant power supplies.
  —Place in backup mode with command **power redundancy-mode redundant**.
  —View power supply settings using **show power**.
- Fans
- Hot swap modules

Topological redundancy:

- Provides redundant switching paths so there isn't a single point of failure.
- Implements network monitoring to recognize failures and repair them.
- With redundant paths, device can be offline for upgrades without disrupting network service.
- Does not co-locate devices, so that problems in the space do not affect more than a single piece of equipment.

Use all methods to split traffic between redundant paths, increasing aggregate network bandwidth

Default Gateway Redundancy

Gateway Discovery

Specifying a default gateway leads to a single point of failure.

Many methods exist for hosts to dynamically discover gateways, but all have problems.

- Proxy ARP
  —Host ARPs for all destinations, even remote.
  —Router responds with its MAC.
  —Problem: Slow failover because ARP entries take minutes to timeout.
- ICMP Router Discovery Protocol (IRDP)
  —Routers use IRDP to advertise default routes.
  —IRDP advertisements have a lifetime—If the lifetime expires without hearing a readvertisement, another gateway is chosen.
  —Problem: Slow failover because advertisements have a default lifetime of 30 minutes.
- Routing protocol
  —PC runs routing protocol to discover best routes.
  —Usually RIP.

Router Redundancy

Instead of making the host responsible for choosing a new gateway, router redundancy protocols allow two or more routers to support a shared MAC address. If the primary router is lost, the backup router assumes control of traffic forwarded to that MAC.

Hot Standby Router Protocol (HSRP)

- Cisco proprietary.
- Two or more devices support a virtual router with made up MAC and unique IP address.
- *Active* router forwards traffic.
- *Standby* is backup. Monitors periodic hellos to detect Active failure.
- Active router is chosen because it has higher HSRP priority (default 100).
- A new router with a higher priority does not cause an election unless it is configured to PREEMPT.
- Shared MAC is 0000.0c07.AC*xx*, where *xx* is the HSRP group.
- Multiple groups (virtual routers) allowed.
- On failure, standby device starts using IP and MAC of the virtual router.
- Interface tracking allows priority to change if a connection is lost.
- HSRP devices move between these states:
  —Initial—HSRP not running.
  —Learn—The router does not know the virtual IP address and is waiting to hear from the active router.
  —Listen—Router knows IP and MAC of virtual router, but not the identity of other HSRP group members.
  —Speak—Router sends period HSRP hellos and elects active router.
  —Standby—Router monitors hellos from active router and assumes responsibility if active router fails.
  —Active—Router forwards packets on behalf of the virtual router.

Configuring HSRP:

- Configure router as member of HSRP group 39 for virtual router with IP 10.0.0.1:

```
Router(config-if)# standby 39 ip 10.0.0.1
```

- Configure priority (default 100, prefers highest):

```
Router(config-if)# standby 39 priority 150
```

- Allow router to take over if active router has lower priority:
  `Router(config-if)# standby 39 preempt`
- Change hello timer to 2 seconds and hold timer to 7 seconds. Can be set between 1–255 seconds (default is hello 3 seconds and hold 10 seconds):
  `Router(config-if)# standby 39 timers 2 7`
- Track interface—If serial0 is down, decrement HSRP priority by 100:
  `Router(config-if)# standby 39 track s0 100`

NOTE Other routers must be configured for PREEMPT to take control.

- View HSRP status:
  `show standby interface fasteth 0/0 or show standby brief`
- Monitor HSRP activity:
  `debug standby`

Virtual Router Redundancy Protocol (VRRP)

- Similar to HSRP, but open standard (RFC 2338).
- Two or more devices support either real addresses or virtual router addresses.
- *Master* router forwards traffic. If a real address is being supported, owner of real address *must* be master.
- *Backup* takes over if master fails. Monitors periodic hellos to detect active failure.
- Master chosen because 1) it owns the real address or 2) it has higher priority (default 100).
- Multiple redundancies (real or virtual) allowed.

Gateway Load Balancing Protocol (GLBP)

- Similar to HSRP or VRRP, but simultaneous use of gateways allowed, maximizing bandwidth.
- Automatically detects and routes around gateway failure.
- Three modes:
  - —Weighted load balancing—Traffic is balanced proportional to configured weight.
  - —Host-dependent load balancing—A given host always uses the same router.
  - —Round-robin load balancing—Each router MAC used to respond to ARP requests in turn.
- Active Virtual Gateway (AVG or *master gateway*) is the only router to respond to ARPs. It uses this capacity to balance load.
- GLBP can track interface; if interface goes down ARPs redirect traffic to other routers.

Single Router Mode (SRM):

- Used by switches with redundant MSFC2 cards
- Only one MSFC forwards traffic
- If first MSFC fails, backup starts. Current FIB used until new router starts.
- Both must run same IOS and have same configuration

Configuring Single Router Mode (SRM)

- Enable redundancy:
  `L3Switch(config)# redundancy`
- Enable high availability:
  `L3Switch(config-r)# high-availability`
- Enable SRM:
  `L3Switch(config-r-ha)# single-router-mode`
- Verify:
  `L3Switch# show redundancy`

Server Load Balancing (SLB)

SLB distributes client requests between several servers. Clients send traffic to a single virtual address, and SLB intelligently distributes requests to the group.

- Lighter load on each server results in better performance
- Server failures are recognized, and server is removed from group until restored.
- Individual server might be removed for maintenance.

Configuring SLB

- Define a server farm name:
  `L3Switch(config)# ip slb serverfarm ponderosa`
- Identify real servers by IP address:
  `L3Switch(config-slb-sfarm)# real 10.1.2.3`
- Activate SLB for each real server:
  `L3Switch(config-slb-real)# inservice`

- View the list of real servers in server farm:
  `L3Switch# show ip slb real`
- View status of server farm:
  `L3Switch# show ip slb serverfarm`
- Define virtual server farm name:
  `L3Switch(config)# ip slb vserver benjamin`
- Identify virtual server IP address:
  `L3Switch(config-slb-vserver# virtual 202.101.100.9 255.255.255.0`
- Link virtual server with server farm:
  `L3Switch(config-slb-vserver)# serverfarm`
- Activate virtual server:
  `L3Switch(config-slb-vserver)# inservice`

IP Multicast and IP Telephony in a Switched Network

A multicast is a single data stream sent from one source to a group of recipients. In contrast, a unicast is traffic from one source to one destination. A broadcast is traffic from one source to all destinations. Some features of multicast traffic are as follows:

- Sending host does not know the identity of the receiving hosts; they are all identified by one group IP address.
- Group membership is dynamic. Hosts join a group, notify their upstream router, and the router begins forwarding data to them.
- Hosts can belong to more than one group.
- Hosts in a group can be located in many different places.

Multicast IP Addresses

Multicasts use the IP address range of 224.0.0.0 to 239.255.255.255. The first four bits of the first octet are always binary 1110. The remaining 28 bits identify the multicast group. Some addresses are reserved:

- 224.0.0.1 is the all-hosts group.
- 224.0.0.2 is the all-routers group.
- The rest of the 224.0.0.0/16 range is reserved for network protocols.
- 224.0.1.0 to 238.255.255.255 are for use over the Internet and are called *globally-scoped addresses*.
- Source specific multicast uses 232.0.0.0 to 232.255.255.255 addresses.
- 233.0.0.0 to 233.255.255.255 are used to assign a static multicast address for use by an organization. The second and third octets of the address are the organization's autonomous system number. This is called GLOP—a combination of *global* and *scope*.
- The 239.0.0.0 to 239.255.255.255 range is for local use within an organization. They are called *limited scope* or *administratively scoped* addresses.

Multicast Distribution Trees

Multicasts use two different ways to distribute data between a server and hosts:

- A **source-based tree** is the simplest kind. Its root is the server, and it forms branches out through the network to all the members of the multicast group. A source tree is identified by (*S*,*G*) where *S* is the IP address of the server and *G* is the group multicast address. It creates optimal paths between the server and the hosts, but takes more router resources. Every router along the path must maintain path information for every server.
- A **shared tree** selects a common root, called a rendezvous point (RP). The server sends traffic to the RP, which forwards it toward hosts belonging to the group. The tree is identified by (*,*G*) where * means any source and *G* is the group multicast address. Shared trees use less router resources, but might result in suboptimal paths.

Reverse Path Forwarding

Multicast routers identify upstream ports (pointing toward the server or RP) and downstream ports (pointing toward other receivers) for each multicast group. The upstream port is found using Reverse Path Forwarding (RPF). RPF involves looking at the routing table to see which interface the router would use to send unicast traffic to

that server or RP. That interface is the upstream port, or RPF port, for that multicast group. The RPF check is done every 5 seconds. It is used in this way:

- If a multicast packet arrives on the RPF port, the router forwards the packet out the interfaces listed in the outgoing interface list of a multicast routing table.
- If the packet does not arrive on the RPF port, the packet is discarded to prevent loops.

Protocol Independent Multicast (PIM)

PIM is a protocol used between routers to keep track of where to forward traffic for each multicast group. It can use information gathered from any routing protocol. PIM can run in dense mode or sparse mode.

PIM Dense Mode

PIM dense mode uses source-based trees. When running in dense mode, PIM assumes that every router needs to receive multicasts. Any router that doesn't want to receive it must send a prune message upstream toward the server. PIM dense mode is most appropriate when:

- Multicast servers and receivers are near each other.
- There are just a few servers and many receivers.
- You have a high volume of multicast traffic.
- The multicast stream is fairly constant.

PIM Sparse Mode

PIM sparse mode uses shared distribution trees. It does not assume that any routers want to receive the multicast, but instead waits to hear an explicit message from them, joining the group. Then, it adds branches to the tree to reach the hosts behind those routers. PIM sparse mode uses rendezvous points to connect hosts and servers. After the connection is made, PIM switches over to a source tree. Sparse mode is used when:

- Pockets of users are widely dispersed around the network.
- Multicast traffic is intermittent.

PIM Sparse-Dense Mode

An interface can be configured in sparse-dense mode. Then, if the router knows of a RP for its group, it uses sparse mode. Otherwise, it uses dense mode. Additionally, it makes the interface capable of receiving multicasts from both sparse and dense mode groups.

Configuring Multicast Routing and PIM

- Give this command to enable multicast routing:
  `(config)# ip multicast routing`

- PIM mode must be configured at each interface with the following command. Configuring PIM on an interface also enables IGMP on that interface:
  `(config-if)# ip pim {sparse-mode | dense-mode | sparse-dense-mode}`

- When using sparse mode, a RP must be specified. A router knows that it is an RP when it sees its own address in the command:
  `(config)# ip pim rp-address ip-address`

Auto-RP

Auto-RP automates the discovery of RPs in a sparse or sparse-dense PIM network. RPs advertise themselves to a router designated as an RP mapping agent. The mapping agent then decides on one RP per group and sends that information to the other routers.

- To configure a router as an RP, type
  `(config)# ip pim send-rp-announce type number scope ttl group-list access-list-number`

- To configure a router as a mapping agent, type
  `(config)# ip pim send-rp-discovery scope ttl`

PIM Version 2

Cisco routers with recent versions of the IOS use PIM v2 by default. Some differences between PIM v1 and PIM v2 include the following:

- PIM v1 is Cisco proprietary, whereas PIM v2 is standards-based.
- Both versions can dynamically map RPs to multicast groups. PIM v1 uses an Auto-RP mapping agent; PIM v2 uses a bootstrap router (BSR).
- PIM v1 uses a Time-T2222o-Live value to bound its announcements, PIM v2 uses a
  configured domain border.
- In PIM v2, sparse and dense mode are group properties, not interface properties.

To configure PIM v2, configure at least one router as a BSR and selected routers as RPs. To configure a BSR:

`(config)# ip pim bsr-candidate interface hash-mask-length [priority]`

To configure a router as a candidate RP:

`(config)# ip pim rp-candidate type number ttl group-list access-list-number`

Internet Group Management Protocol

When a host wants to join a multicast group, it sends an Internet Group Management Protocol (IGMP) message to the router. The router periodically checks for group members on each segment. There are three versions of IGMP.

IGMP Version 1

Multicast routers query each segment periodically to see if there are still hosts in multicast groups with a query sent to the all-hosts address of 224.0.0.1. One host on the segment responds. Hosts silently leave a group; the router doesn't know they are gone until it queries and nobody responds.

IGMP Version 2

Version 2 adds explicit leave messages that hosts send when they leave a group. Queries are sent to specific multicast group addresses, not the all-hosts address.

IGMP Version 3

Hosts are able to tell the router not only which multicast groups they belong to, but also which sources they accept multicasts from. It adds two modes for requesting membership in a multicast group:

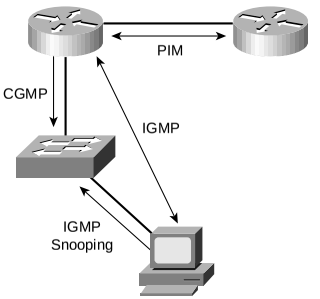
- **Include mode**—The receiver lists the groups to which it belongs, and the servers it uses.
- **Exclude mode**—The receiver lists the group to which it belongs, and the servers it doesn't use.

Cisco Group Management Protocol

Switches flood multicasts by default. Cisco Group Management Protocol (CGMP) lets a router tell a switch which hosts belong to which multicast group, so the switch can add that information to its port-to-MAC address mapping. Then, when a multicast comes in, the switch forwards it out only to ports that have hosts belonging to that group. CGMP is Cisco proprietary.

IGMP Snooping

IGMP snooping is another way for the switch to find out which ports have multicast hosts. When it is enabled, the switch opens all multicast packets, looking for IGMP join or leave messages. When it finds one, it records that information and uses it for forwarding multicasts. Because every multicast packet has to be opened, this can cause a performance hit on the switch.



Verifying Multicast Routing

Some commands to verify multicast routing include the following:

- **show ip mroute**—This shows the contents of the multicast routing table. For each group, it lists the mode, the RPF neighbor, the group identifier, and outgoing interfaces.
- **show ip mroute summary**—Lists each multicast group without as much detail.
- **show ip mroute active**—Shows the active sources, and the sending rate of each.
- **show ip mroute count**—Shows traffic statistics for each multicast group.
- **show ip pim interface**—Lists each interface doing multicasting, its PIM mode, and number of neighbors
- **show ip pim rp**—Lists RPs the router knows about.
- **show ip pim rp-hash**—Shows the RP selected for each multicast group.
- **show ip pim bsr**—Lists the current BSR.

Cisco IP Telephony

Packet loss is one of the biggest enemies of voice transmissions, and is often caused by jitter and congestion. Jitter (variable delay) causes buffer over- and underruns. Congestion at the interface can be caused by traffic from a fast port being switched to exit out a slower port, which causes the transmit buffer to be overrun.

Cisco switches are well suited to support both voice and video transmission (AVVID) because of the following features:

- They support multiple VLANs on each access port, by using Voice VLANs. This enables the IP Phones to belong to a separate VLAN from the computers.
- They can classify, mark, and police traffic, as well as provide differentiated queuing to different classes of traffic.
- They can be configured to trust the QoS markings provided by the IP phones or other devices.

Preparing the Network

When adding voice or video to an existing network, you should examine several things in advance:

1. **What features are needed?**—Power for IP phones, voice VLANs on the switches, network redundancy for high availability, security for voice calls, QoS settings.
2. **The physical plant**—Cabling at least CAT-5.
3. **Electrical power for the IP phones**—Use either inline power from Catalyst switch or power patch panel. Need uninterruptible power supply (UPS) with auto-restart, monitoring, and 4-hour response contract. Also generator backup. Maintain correct operating temperatures.
4. **Bandwidth**—Commit no more than 75 percent of bandwidth. Consider all types of traffic—voice, video and data. Have more than enough bandwidth if possible. Include both voice and call-control traffic in your planning.

Network and Bandwidth Considerations

The network requirements for Voice over IP (VoIP) include

- Maximum delay of 150–200 ms (one-way)
- No more than 1 percent packet loss
- Maximum average jitter of 30 ms
- Bandwidth of 21—106 kbps per call, plus about 150 bps per phone for control traffic

The network requirements for streaming video include

- Maximum delay of 4–5 sec (one-way)
- No more than 2 percent packet loss
- No jitter requirements
- Bandwidth needed depends on the video stream

The network requirements for video conferencing include

- Maximum delay of 150—200 ms (one-way)
- No more than 1 percent packet loss
- Maximum average jitter of 30 ms
- Bandwidth required is 20 percent more than the size of the videoconferencing stream

A formula to use when calculating bandwidth needed for voice calls is as follows:
(Packet payload + all header in bits) * Packet rate per second

Auxiliary (or Voice) VLANs

Cisco switches can be configured to dynamically place IP telephones into a VLAN separate from the data VLANs. They can do this even when the phone and PC are physically connected to the same switch port. A term you might see is VVID—this is the voice VLAN ID, which is the same thing as the number of the auxiliary VLAN.

Voice VLANs allow phones to be dynamically placed in a separate IP subnet from hosts, to have QoS (using 802.1Q/p headers) and security policies applied, and makes troubleshooting easier.

Voice in the Building Access Submodule

Include the following in the Building Access Submodule when implementing VoIP:

- Auxiliary VLANs
- 802.1p/Q encapsulation between the phone and the switch, which allows QoS marking
- Use switches that support multiple output queues
- Use switches that support inline power to IP phones
- Configure the following on switch ports connected to IP Phones, and in the network in general:
  - —STP Port Fast
  - —Root Guard
  - —Unidirectional Link Detection (UDLD)
  - —Uplink Fast

Support for Voice in the Building Distribution Submodule

To support VoIP, use the following in the Building Distribution Submodule:

- Make sure VoIP ports do not participate in routing—either use passive interface or configure the network statements under the routing protocols properly.
- Use HSRP for Layer 3 redundancy.
- Tune the routing protocol to allow for fast detection of a lost path and quick convergence when the network changes.

Implementing QoS in a Switched Network

Quality of service (QoS) configurations give special treatment to certain traffic at the expense of others. Using QoS in the network addresses these problems:

- Packet loss due to data being dropped at a congested interface
- Delay of sensitive data such as voice and video
- Jitter (variable delay)

People sometimes think that there is no need for QoS strategies in a LAN. However, switch ports can experience congestion because of port speed mismatches, many people trying to access the switch backbone, and many people trying to send traffic to the same switch port (such as a server port).QoS is disabled by default on switches. It is enabled at the interface configuration mode with the following command:

```
(config-if)# mls qos
```

QoS Techniques

Three QoS strategies are commonly implemented on interfaces where traffic enters the switch:

- **Classification**—Distinguishing one type of traffic from another. After traffic is classified, other actions can be performed on it. Examples: access lists, class maps, NBAR.
- **Marking**—Placing class of service (CoS), IP Precedence, or DiffServ Code Point (DSCP) values on the classified traffic.
- **Policing**—Determining whether or not a specific type of traffic is within preset bandwidth levels. If so, it is usually allowed and might be marked. If not, the traffic is typically marked or dropped. Example: CAR and class-based policing.

Some other QoS techniques are typically used on outbound interfaces:

- **Traffic shaping and conditioning**—Attempts to send traffic out in a steady stream, at a specified rate. Buffers traffic that goes above that rate and sends it when there is less traffic on the line.
- **Queuing**—Once traffic is classified and marked, one way it can be given special treatment is to be put into different queues on the interface, to be sent out at different rates and times. Examples: priority queuing, weighted fair queuing, custom queuing. The default queuing method for a switch port is FIFO.
- **Dropping**—Normally, interface queues accept packets until they're full and then drop everything after that. You can implement prioritized dropping, so that less important packets are dropped before more important ones. Example: Weighted Random Early Detection (WRED).

Integrated Services (IntServ)

Integrated services is a QoS model that guarantees a specific amount of bandwidth to the identified traffic, throughout the entire network. A check is made of the path from sender to receiver, and each router along the way has to reserve bandwidth for that flow. This is done using RSVP—Resource Reservation Protocol. If the network cannot provide the required bandwidth, the session is not allowed. RSVP is typically used for voice applications.

Differentiated Services (DiffServ)

Differentiated services provide levels of service based on the value of certain bits in the IP or ISL header, or the 802.1Q tag. Each hop along the way must be configured to treat the marked traffic the way you want—this is called per-hop behavior (PHB).

- In the Layer 3 IP header, you use the 8-bit ToS field. You can set either IP Precedence, using the top 3 bits, or Differentiated Services Code Points (DSCP) using the top 6 bits of the field. The bottom 2 bits are not used. The default DSCP value is 0, which corresponds to best-effort delivery.
- At Layer 2, with ISL, you can set 3 of the 4 bits in the ISL priority field to set the class of service (CoS). With 802.1Q, you set the 3 802.1p bits to the COS. The values of these 3 bits correspond to the IP Precedence values.

IP Precedence/Class of Service

Using three bits for IP Precedence gives you 8 possible values. The following table shows the values and their meaning. Precedence 5 is usually used for voice traffic; 6 and 7 are reserved for such things as routing protocols. Normal data is typically given Precedence 0. These same values apply for CoS bits also.

Precedence/CoS	Name
7	Network
6	Internet
5	Critical

Precedence/CoS	Name
4	Flash-override
3	Flash
2	Immediate
1	Priority
0	Routine

Translating Between DSCP and CoS

When traffic comes into the switch already marked with a COS or IP Precedence value and the switch trusts that, it assigns a DSCP value for its own internal use. If the frame has an existing DSCP value and the switch trusts that, it assigns the same value for the internal DSCP. Similarly, the switch can also translate a DSCP value into a CoS setting when sending data out a trunk port. The default CoS to DSCP mappings are shown in this table:

CoS	DSCP	CoS	DSCP
0	0	4	32
1	8	5	40
2	16	6	48
3	24	7	56

The default mappings of DCSP to COS are shown in this table:

DSCP	CoS
0–7	0
8–15	1
16–23	2
24–31	3
32–39	4
40–47	5
48–55	6
56–63	7

DiffServ Assured Forwarding

The 6 DSCP bits can be broken down into two sections: the first 3 bits define the DiffServ Assured Forwarding (AF) class, and the next 2 bits define the drop probability within that class. The sixth bit is 0 and unused. AF classes 1–4 are defined, and within each class, 1 is low drop probability, 2 is medium, and 3 is high (meaning that traffic is more likely to get dropped if there is congestion). Each hop still needs to be configured for how to treat each AF class.

	Low Drop	Medium Drop	High Drop
Class 1	AF11	AF12	AF13
Class 2	AF21	AF22	AF23
Class 3	AF31	AF32	AF33
Class 4	AF41	AF42	AF43

DiffServ Expedited Forwarding

Another predefined DiffServ classification is Expedited Forwarding (EF). This is equivalent to DSCP 46 and is for use by your highest priority traffic, such as voice. You configure each hop in the network for the type of service you want EF traffic to receive.

Classifying Traffic and Marking for QoS

Mark traffic for QoS as close to the source as possible. If the source is an IP telephone, it can mark its own traffic. If not, the building access module switch can do the marking. If those are not under your control, you might need to mark at the distribution layer. Classifying and marking slows traffic flow, so don't do it at the core. All devices along the path should then be configured to trust the marking and provide a level of

service based on it. The place where trusted marking is done is called the *trust boundary*. To configure a switch to trust the markings at an interface:

```
(config-if)#mls qos trust {dscp | cos}
```

When IP traffic comes in already marked, the switch has some options about how to handle it. It can

- Trust the DSCP value in the incoming packet, if present
- Trust the IP Precedence value in the incoming packet, if present
- Trust the CoS value in the incoming frame, if present
- Classify the traffic based on an IP access control list, or a MAC address ACL

Handling Non-IP Traffic

Non-IP traffic does not have fields in the header for Type of Service. The switch can handle this in the following ways:

- Use the default port CoS value if the frame does not have a value assigned
- Trust the already-assigned CoS value in the incoming frame, if present
- Classify the traffic based on a MAC address access control list

Classifying and Marking Using MQC

Modular QoS command-line interface (MQC) is a method of classifying traffic, marking the traffic, and setting policies for that traffic that can be used on most devices with most kinds of policies. Here are the general steps:

1. Create the necessary access control lists, if classifying traffic by ACL, or configure NBAR if your switch supports that (e.g., 6500).
2. Create the class maps that specify matching such things as ACLs, protocol, DSCPs, or IP Precedence values.
3. Create a policy map that calls each class map and defines the policy for each.
4. Apply the policy map to the appropriate switch ports.

When access control lists (ACLs) are used to classify traffic, the way a switch reacts to specific access control entries (ACEs) is different in a QoS context than with security-based ACLs. In a QoS access list,

- If the traffic matches a *permit* statement, the designated QoS action is taken
- If the traffic matches a *deny* statement, the rest of the ACEs in that ACL are skipped, and the switch goes to the next ACL.
- If there are multiple ACLs in a policy applied to an interface, the switch stops reading them as soon as a permit statement match is found for the traffic.
- If the traffic does not match any ACL entry, the switch just gives best-effort delivery to the traffic.

Configuring MQC

First, configure the access lists if using them.

Second, configure a class map for each classification of traffic:

```
(config)# class-map [match-any | match-all] name
(config-cmap)# match match options, such as ACL
```

Third, configure a policy map that calls the class maps and sets policies or types of treatment for each class:

```
(config)#policy-map name
(config-pmap)#class class-map name
(config-pmap-c)#policy options, such as  set DSCP
```

Finally, apply the MQC policy to the desired interface(s), either inbound or outbound. Note: Policy maps that classify traffic using ACLs, that set DSCP or IP Precedence, or that tell the interface to trust existing markings can be applied only inbound.

```
(config-if)#service-policy {output | input} name
```

Queuing Methods

FIFO (First-In, First-Out)

The default on switch ports. If QoS is not enabled, there is one software queue. If QoS is enabled, there are four software queues per port, but they are all weighted and serviced equally, with best-effort delivery. Traffic is placed in them based on CoS value:

Queue 1—CoS values 0 and 1
Queue 2—CoS values 2 and 3
Queue 3—CoS values 4 and 5
Queue 4—CoS values 6 and 7

Priority Queuing (PQ)

Queues are assigned different priority values, and the high priority queue gets serviced before anything else. Priority queuing is done on the 3550 using the expedite queue, which is a strict priority queue. It is serviced ahead of the other queues until it is empty. This queue is configured on the 3550 at interface configuration mode with the command: **priority-queue out**.

Custom Queuing (CQ)

Reserves a part of the interface bandwidth for the different queues. Can classify and place specific traffic into the queues.

Weighted Fair Queuing (WFQ)

Gives weights to different types of traffic, and allows lower weighted traffic more bandwidth. Traffic can be weighted by flow (conversation) or using class maps.

Low Latency Queuing (LLQ)

Has one priority queue and usually voice traffic is put into this. Uses class-based WFQ for the rest of the interface traffic. Configure this under the class statement in the policy map:

```
(config-pmap-c)# priority bandwidth
```

IP RTP Priority

Is similar to LLQ in that is has a priority queue and uses WFQ for other traffic. However, here the priority queue is completely for voice traffic. RTP is Real Time Protocol, the protocol used by Voice over IP. It is configured at the interface:

```
(config-if)# ip rtp priority start-port port-range BW
```

Weighted Round Robin (WRR)

This is the process that takes packets from the queues, decides which queue goes when, and how many packets can be sent from each queue at a time. During times of interface congestion, WRR weights queues, and more packets are sent from higher weighted queues, thus giving them more bandwidth.

What Happens When the Software Queues Get Full?

By default, when a software queue is full (congested) the switch just drops all other traffic bound for that queue. This is called *tail drop*. It can cause some problems:

- TCP global synchronization.
- TCP buffer starvation.
- Delay and jitter.
- High priority traffic is dropped while low priority traffic is sent.

Congestion avoidance is accomplished by using Weighted Random Early Detection (WRED). WRED starts dropping lower priority traffic (based on DSCP or IP Precedence values) as the queue starts to fill, and drops high priority traffic only when the queue is almost full. The drop thresholds and the drop ratios are configurable. WRED works best with TCP traffic, because TCP dynamically adjusts its sending rate when packets are dropped. Do not use WRED for voice traffic. If the queue fills completely, tail drop is used.

On the 3550, the gigabit Ethernet ports can use either tail drop or WRED (Weighted Random Early Detection). The 10/100 ports can use only tail drop. WRED is enabled either in a policy map or at the interface—the command is the same:

```
random-detect dscp-based
```

Traffic Policing

By using the QoS policing function, bandwidth use can be controlled on physical interfaces in the switch. Traffic cannot be policed per VLAN or on an SVI. Policing specifies an amount of bandwidth allowed for a particular type of traffic, and generally drops traffic over that amount. It can also be configured to allow the excess traffic, but mark it with a different DSCP value.

The 3550 switch can police bandwidth use either for each individual class of traffic (individual policing), or it can limit bandwidth use for all traffic (aggregate policing).

Traffic Shaping

Traffic shaping also controls the amount of traffic used by a specified type of traffic, but shaping buffers the excess traffic instead of dropping it. Because data is usually bursty, the buffered traffic can be sent out between bursts. It thus smoothes out the flow of traffic.

Creating Bandwidth by Compression

Compressing the traffic on a line creates more useable bandwidth; because each frame is smaller, there are fewer bits to transmit. You can compress the whole payload, or just compress the protocol headers with TCP or RTP header compression. Cisco supports three Layer 2 payload compression algorithms:

- Stacker
- Predictor
- Microsoft Point-to-Point Compression (MPPC)

Link Fragmentation and Interleave (LFI)

A typical network has a range of packet sizes. Small packets can be delayed waiting for a large packet to be sent out the interface. LFI breaks large packets into smaller segments and intersperses the smaller packets between the pieces of the big ones. This reduces delay and jitter.

In summary, options that are available to you when configuring a switch's outbound (egress) queues include

- Changing the CoS-to-queue map
- Assigning drop thresholds to each queue
- Mapping DSCPs to the drop thresholds
- Enabling either WRED or tail drop
- Changing the size of buffer space allotted to each queue
- Changing the relative weight of each queue

QoS at the Building Access Layer

Enable QoS at the building access layer, then classify and mark the traffic, and perhaps do policing. If the traffic is already classified by a trusted end station, configure the switch to trust the markings. Configure voice VLANs if using IP phones.

QoS at the Building Distribution Layer

Enable QoS at the building distribution layer and then configure the switch to trust the priority marking it receives from the access layer switches. If the markings are from an untrusted source, configure the switch to override them. You might then want to modify the switch's default per-hop behavior based on these values. If using WRED, you might change the DSCP-to-threshold mappings. You might also change the DSCP-to-CoS mappings, to put traffic in different egress queues. Lastly, you typically change the relative weights of the queues on the egress interface.

QoS at the Campus Backbone

No classification or marking should be done at the core layer, as this slows down traffic. A congestion avoidance mechanism such as WRED might be used, along with interface queuing techniques such as class-based weighted fair queuing or low latency queuing to guarantee bandwidth to critical applications.

QoS for Voice over IP

In a network with voice traffic, configure either the end stations or the switch to mark the voice traffic with IP Precedence 5 or DSCP 46. Configure the egress interface for priority queuing, then configure the DSCP-to-CoS mappings to put the voice traffic in the Expedite queue (on the 3550) or the highest priority queue (on the 2950).

Verifying QoS

Use the following commands to verify your QoS configurations and actions:

- **show class-map** [*name*]—Displays the configured class maps, or just the one named.
- **show policy-map** [*name*]—Displays the configured policy maps, or just the one named.
- **show policy-map** [**interface** [*interface-spec* [*input* | *output*] [**class** *class-name*]]]—Displays the policy maps and statistics by interface and/or class.
- **show queueing** [**interface** *interface-no.*]—Shows the queuing strategy and statistics for any queues configured on the interface.
- **show policy interface** *interface-no.*—Displays the policies for all classes applied to the interface, along with statistics.
- **debug ip rsvp**—If using RSVP for voice, shows information about packets received and sent.
- **debug priority**—Shows information on the priority queue.

Optimizing Performance of Campus Networks

Techniques to Optimize Performance

- Monitor network continuously
- Understand nominal behavior (baseline)
  - —Utilization
  - —Response times
  - —Errors
- Anticipate capacity issues—New hardware or circuits can take weeks to be installed

Protocol Analysis tools in Cisco switches

Switched Port Analyzer (SPAN)

- Copies network traffic from a switch port or VLAN to a listening port. Can monitor incoming, outgoing, or both.
- Captures the traffic with a protocol analyzer (such as Sniffer or Ethereal) attached to listening port.

- Multiple SPAN sessions are supported.
- The following example configures SPAN to copy traffic from port fastethernet 2/2 (incoming and outgoing) to fastethernet 2/48 as session number 7.

```
L3Switch(config)# monitor session 7 source interface fastethernet 2/2 both
L3Switch(config)# monitor session 7 destination interface fastethernet 2/48
```

VLAN-Based SPAN (VSPAN)

- Same idea as SPAN but copies all traffic incoming or outgoing on ports in a VLAN to monitor port.
- Traffic internally routed to VLAN not monitored (does not come in or go out a VLAN port).
- Monitor port might be in same or different VLAN.

Remote SPAN (RSPAN)

- Same idea as SPAN but copies traffic to a remote monitor port.
- Supports source ports and source VLANs.
- VTP pruning can block monitored traffic.
- Monitored traffic carried over a single-purpose VLAN.

```
L3Switch(config)# vlan 999
L3Switch(config-vlan)# remote-span
```

- The following example configures RSPAN to copy traffic from port fastethernet 2/2 (incoming and outgoing) to VLAN 999 as session number 8:

```
L3Switch(config)# monitor session 8 source interface fastethernet 2/2 both
L3Switch(config)# monitor session 7 destination remote vlan 999
```

- View SPAN settings:

```
L3Switch# show monitor session 7
```

Network Analysis Module (NAM)

- Module for Catalyst 6000/6500
- Accumulates flow information using Remote Monitoring (RMON) and by monitoring VLANs
- Uses TrafficDirector or any RMON application to analyze data

To configure NAM, assign IP settings and start web server:

```
L3Switch# session slot processor 1
root@localhost#ip address 10.0.0.2 255.255.255.0
root@localhost#ip broadcast 10.0.0.255
root@localhost#ip host MyNAM
root@localhost#ip gateway 10.0.0.1
root@localhost#ip domain stewart.hickory.nc.us
root@localhost#ip nameserver 10.0.0.254
root@localhost#snmp location At home
root@localhost#snmp contact Brent Stewart
root@localhost#snmp community public ro
root@localhost#snmp community private rw
```

- Identify the set of information you want collected. Choose from these collections:
  - —addressmap
  - —art (application response time)
  - —etherstat
  - —priostats
  - —vlanstats

```
root@localhost#autostart addressmap enable
```

- Configure the NAM collection port 1. NAM must monitor session 1.

```
L3Switch(config)# monitor session 1 destination interface gigabit 8/0
```

Viewing NAM:

```
show module
show interface gigabit 8/1
```

Security in the Campus Network

Securing Cisco Devices

Here are some basic security suggestions for network devices:

- Use passwords that are not susceptible to dictionary attack. Add numbers or substitute numbers and symbols for letters, for example, substituting *0* for *o* and using *cisc0*.
  - —Console
  - —AUX
  - —Enable
  - —SNMP
  - —VTP
- Limit Telnet access using access lists.

- Physically secure access to the device.
- Use banners that warn against unauthorized access.
- Remove unused services:

```
no service finger
no service config
no service tcp-small-services
no service udp-small services
no cdp enable
no ip http
```

- Set up and monitor syslog.
- Disable automatic trunking on all non-trunk ports

Authentication, authorization, and accounting (AAA):

- Verifies ID.
- Limits privileges.
- Logs usage for billing or monitoring.
- Configures authentication:

```
L3Switch(config)# aaa new-model
```

—Identify authentication methods (RADIUS first and then the local username/password database in this example):

```
L3Switch(config)# aaa authentication login default radius local
```

—Apply to a line:

```
L3Switch(config)# line vty 0 4
L3Switch(config-line)# login authentication default
```

- Configure authorization:

—Identify authorization methods (RADIUS in this example):

```
L3Switch(config)#aaa authorization network default radius
```

—Apply to interface:

```
L3Switch(config)# interface s0/1
L3Switch(config-line)# ppp authorization default
```

- Configure accounting:

—Identify accounting method:

```
L3Switch(config)# aaa accounting network default start-stop radius
```

—Apply to interface:

```
L3Switch(config)# interface s0/1
L3Switch(config-line)# ppp accounting default
```

Limiting MAC Access

- Port security limits the number of MAC addresses learned on a port:

```
L3Switch(config-if)# switchport port-security max 1 violation shutdown
```

- 802.1X limits network access by authenticating at data link before allowing access:

```
L3Switch(config)# aaa new-model
L3Switch(config)# aaa authentication dot1x default group radius
L3Switch(config)# dot1x system-auth-control
L3Switch(config)# int fasteth2/1
L3Switch(config-if)# dot1x port-control auto
```

- View security settings:

```
L3Switch# show port-security
```

Access Lists

Cisco switches support

- Traditional Router ACL (RACL)
- QoS ACL
- VLAN ACL (VACL)

VLAN ACL (VACL)

- Applied against all VLAN traffic.
- Similar to route-maps:

—Statements contain match and set conditions

—Statements numbered for ordering

- Actions: Permit, Deny, Redirect
- The following is a sample VACL to drop traffic that matches ACL 101:

```
L3Switch(config)# vlan access-map Kaitlyn 5
L3Switch(config-access-map)# match ip address 101
L3Switch(config-access-map))# action drop
L3Switch# vlan filter Kaitlyn vlan_list 10
```

- View VACL settings:

```
show vlan access-map Kaitlyn
show vlan filter access-map Kaitlyn
```

Private VLAN (PVLAN)

PVLANs allow service providers to isolate customers into separate multiaccess domains. Using a VLAN for each customer isn't scalable. PVLANs isolate a set of ports from other ports in a VLAN.

Port and VLAN types

- **Community**—Communicate with a community, plus promiscuous
- **Isolated**—Communicate just with promiscuous
- **Promiscuous**—Communicate with all

To configure VLAN, enter the following at the prompt:

```
L3Switch(config)# vlan 777
L3Switch(config-vlan)# private-vlan isolated
```

Metro Ethernet

Ethernet as a metropolitan area solution provides attractive features.

For consumers:

- Low cost
- High bandwidth (>1 G)

For service providers:

- Provisioned over dark fiber or existing services
- Profitable
- Supports new services

Transparent LAN Service (TLS)

- Customer switches see MAN as single VLAN
- Supports point-to-point and multipoint
- All locations must peer. Some routing protocols have trouble peering more than 40 devices.
- Easy to implement.
- Broadcast and multicasts aren't controlled, QoS is difficult, and it's not scalable.

Directed VLAN Service (DVS)

- Customer switches see MAN as multiple VLANs, each going to a specific neighbor.
- Supports point-to-point and multipoint.
- VLAN identifies destination, scalable, SPs prefer.
- Requires many VLANs.

Metro Ethernet Over SONET

Metro Ethernet over Synchronous Optical Network (SONET) uses existing bandwidth and redundancy of SONET to facilitate simulated Ethernet service.

- SONET has ring structure.
- Metro Ethernet over SONET emulates hub.
- SONET generally available, quick failover.
- Customer buys bandwidth in chunks of 51.84 M. (OC-x)

Metro Ethernet Over DWDM

Metro Ethernet over dense wavelength division multiplexing (DWDM) uses dark fiber or wavelength.

- Metro Ethernet over SONET emulates hub.
- Gigabit plus bandwidth, and easy to configure.
- Built on dark fiber or wavelength (not generally available).

Metro Ethernet Over CWDM

Metro Ethernet over Course Wave-Division Multiplexing (CWDM) uses dark fiber or wavelength.

- Last mile technology
- Doesn't use bandwidth as efficiently
- Last mile technology
- Cheap (comparatively)

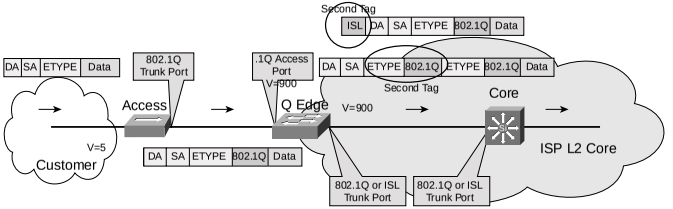
Metro Ethernet Tunneling Options

Traffic crossing the service provider can be encapsulated to preserve private VLAN tags across the backbone.

No Tunneling

- Customer traffic isolated in one or more VLANs in shared definition set.
- Easy to implement.
- Doesn't scale—Service provider runs out of VLANs!
- Used to build low-cost MAN services.

802.1Q-in-Q



- Two dot1q tags associated with frame—One for enterprise and one for service provider.
- Also called *tag stacking*.
- SP reads their tag, removes before passing back.
- Isolates enterprises from each other.
- Enterprise sees Q-in-Q as trunk service between sites.
- Spanning tree used to prevent loops.
- STP can cause issues with backbone.
- Redundant links supported with EtherChannel.
- Easy to set up and support.

MPLS Secret Decoder Ring

Before we discuss Metro Ethernet over MPLS, here's a reminder of important MPLS acronyms is appropriate:

- **Label Switch Router (LSR)**—Device that forwards traffic inside an MPLS domain.
- **Label Distribution Protocol (LDP)**—Protocol that synchronizes label definitions between LSR.
- **Label Switch Controller (LSC)**—MPLS router that works with ATM switch to forward MPLS traffic.
- **Label Edge Router (LER)**—Device that sits between Ethernet and MPLS. Maps Ethernet traffic to MPLS labels.

EoMPLS

- VLAN mapped to MPLS tunnel.
- Point-to-point only.
- Requires either full mesh, or traffic to exit MPLS to a switch and be passed back to MPLS (a *hairpin turn*).
- Very scalable.
- Supports Transport Layer Security (TLS) functionality—makes disparate networks appear as one LAN.
- Uses a tunnel label and a virtual circuit label applied by LER.
- Ingress LER uses Forwarding Equivalence Class (FEC) to map traffic to Label Switch Path.
- LSRs along LSP just use tunnel label to direct traffic.
- Virtual circuit label used by LER to demux.
- CoS mapped to 3 bit EXP field in label.

EoMPLS Point-to-Multipoint

- Provides features of EoMPLS plus multipoint configurations.
- Service acts like an Ethernet switch.
- Efficiently handles traffic (solves hairpin turn).