Exam
Manual

LearnSmart

CISCO (642-813) SWITCH

Cisco Certified Network Associate

LearnSmart's CCNP SWITCH exam manual prepares candidates for Cisco's 642-813 exam, which counts towards CCNP and CCDP certifications. This manual presents complex topics in a clear, direct style so that ambitious professionals can successfully complete the exam and advance their career. Topics covered in this guide include:

- Switch Operation
- SVI Ports
- Supporting Advanced Services
- High Availability
- And more!

Give yourself the competitive edge necessary to further your career as a network professional and purchase this exam manual today! manual today!

Cisco CCNP Switch (642-813) LearnSmart Exam Manual

Product ID: 12377
Production Date: November 10, 2011

Warning and Disclaimer

Every effort has been made to make this document as complete and as accurate as possible, but no warranty or fitness is implied. The publisher and authors assume no responsibility for errors or omissions. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this document.

Volume, Corporate, and Educational Sales

Favorable discounts are offered on all products when ordered in quantity. For more information, please contact us directly:

1-800-418-6789
solutions@learnsmartsystems.com

International Contact Information

International: +1 (813) 769-0920

United Kingdom: (0) 20 8816 8036

Abstract

This Exam Manual is meant to prepare you for the new Cisco CCNP (Cisco Certified Network Professional) SWITCH exam (642-813) that counts towards CCNP and CCDP (Cisco Certified Design Professional) certifications. It is assumed that the candidate and reader have a CCNA level knowledge of switching, although most chapters will start with a small overview of the topics seen in the CCNA studies.

In this new curriculum (version 6) Cisco has limited a number of topics, such as wireless LANs, Quality of Service (QoS) and certain security topics. You still need a basic understanding of each of these topics as well as how to configure a route and switch network to incorporate video, voice, wireless and security devices. At the same time, Cisco has added some design guidelines, especially the planning part of the design process. It also covers more in-depth routing and switching in general. In short, the CCNP is now more specialized towards routing and switching.

It is our recommendation that dedicated CCNP candidates periodically check Cisco's website to find out about the current exam blueprint, as Cisco reserves the right to change it without notice.

What to Know

The curriculum for the new CCNP Track (Version 6) has drastically changed the way Cisco measures a Cisco Certified Network Professional. You'll notice, if you review the objectives we've listed below, that issues dealing explicitly with Quality of Service (QoS) and Wireless LANs are no longer a part of the switching portion of the CCNP curriculum (in the previous version, switching was covered by the Building a Cisco Multilayer Switched Network, 642-812 exam). At the same time, Cisco has added a number of planning and design tasks to the curriculum, mirroring the actual job role of a CCNP. The official objectives for the exam are:

1. Implement VLAN Based Solution
2. Implement a Security Extension of a Layer 2 Solution
3. Implement Switch Based Layer 3 Services
4. Prepare Infrastructure to Support Advanced Services
5. Implement High Availability

In general, successful candidates should intimately know and understand the switching process, especially as it pertains to planning and designing scalable environments where Virtual LANs are used extensively, as the first domain, Implement a VLAN-based Solution contains a high percentage of tested material.

Tips

As has always been the case with Cisco tests, and especially with the CCNP-level exams, a large amount of hands-on experience with Cisco switches is vital to passing the exam. Be prepared to offer setup and configuration routines for a number of different situations, understand the ins and outs of multilayer switching and be able to provide plans and outlines for a scalable, switched network.

Planning Tasks

The word "*plan*" appears several times in the SWITCH exam (642-813) blueprint. It is a new topic included in the recently introduced curriculum that makes the CCNP certified professional a more routing- and switching-specialized professional. Cisco expects the new CCNPs and SWITCH exam takers to be able to perform the following tasks:

- Analyze network design documentation and be able to extract the information necessary for a detailed implementation plan that includes configuration of network devices.
- Analyze design documents and discover missing parts that are required before an implementation can be completed.
- Perform peer review of another engineer's implementation plan, to discover weaknesses and omissions in the planned configurations and update the implementation plan.
- Build a verification plan that lists the required **show** commands and essential information that confirms or verifies whether each planned feature has been implemented correctly.
- Write a verification plan that can be understood and used by a less experienced worker, allowing that worker to implement changes and to verify the changes worked, off-shift, when you are not on-site.
- Perform a peer review on another engineer's verification plan, to discover which key design features are not verified by that plan, and to discover inaccuracies in the plan.

The planning tasks of the exam are those tasks in the blueprint that don't require the use of the CLI. Those topics are the ones starting with the words determine, create and document. Your approach to these topics is to make sure you really understand the concepts behind the actual CLI commands, master the verification commands and most importantly, spend time thinking about the concepts, configuration, and verification commands as if you were writing a network design document, project implementation or verification plan. The specific tools designed to aid in the network design process are explained in guides and training for the Cisco Certified Design Associate (CCDA) and Cisco Certified Design Professional certifications (CCDP) and are not necessary to understand for the SWITCH exam.

In essence, you should be ready for the planning topics of the SWITCH exam when you can do the following:

- Read design goals extracted from a design document, develop a configuration that meets those goals, and discover missing information that needs to be gathered before you can complete the configuration.
- Read an extract from the design or implementation plans to determine what is wrong or missing.
- Read a configuration and design goal stated as being correct and create verification steps to confirm whether the feature works.
- Analyze an extract from a verification plan, along with the stated configuration and design goals, and determine any problems or missing elements in the verification plan.

With those concepts in mind and the information provided in this guide, you should be able to perform the planning duties required to pass the exam and also be ready to perform those duties in real life scenarios.

Table of Contents

Domain 1: Switch Operation

As you remember from your CCNA studies, a switch is a network device that operates in layer 2 of the OSI model (switches have evolved dramatically in the past years, and they now provide incredible network services; this definition is the most basic and is now considered legacy. We will explain the operations of the Multilayer Switch (MLS) briefly later in this chapter and thoroughly in domain 3 of this guide). It breaks collision domains, which are simply physical network segments where data frames can "collide" when they are transmitted at the same time on a shared broadcast medium such as Ethernet. A collision happens when two or more hosts transmit data at the same time over this medium. The Carrier Sense Multiple Access Collision Detect is the mechanism used within Ethernet where hosts determine who transmits data in Ethernet networks. A host is an Ethernet connected device such as a PC. One host listens on the wire and if it doesn't "hear" a transmission it starts sending frames. At that point if another host or hosts transmit, a collision occurs and a backoff timer is automatically set for a random period of time in every host involved in the collision. When the timer is done the host listens and if the medium is available it transmits again.

Every switch port is a separate collision domain, meaning that two hosts connected to different switch ports don't have to share the bandwidth of the media as they had when connected to hubs.

We learn in the CCNA studies that the main functions of a switch are address learning, forward/filter decisions, and loop avoidance.

Switches also provide the following:

- Hosts can operate in full duplex mode, meaning they can talk and listen at the same time. If the host is not able to operate in full duplex mode, the switch can communicate in half duplex mode where the switch and host can only send or receive data at any given time.
- Each access port offers a dedicated bandwidth to the host connected to it (or group of hosts if a hub or another switch is connected).
- Uplink ports to other switches can be trunked to send data from multiple hosts and multiple VLANs. This topic will be discussed in detail later in this guide.
- Errors in frames are not propagated because every frame received in a port is inspected for errors. If the switch finds errors the frame is discarded.
- Other types of layer 2 filtering based advanced features are possible (QoS, CoS, etc).

Address Learning and Forward/Filter Decisions

Address Learning

Switches are considered to be smart devices compared to Hubs because they keep a table of devices' locations on their ports. A switch receives a frame and checks its Forward Table, also called **MAC Address Table** and **Content Addressable Memory (CAM)**, to see if it has an entry for the frame. If it doesn't, it adds the MAC address, switchport where the frame was received and virtual LAN of the port to the table. It will flood the frame out of all ports on the switch, except for the port the frame was received on. The CAM table helps to make the switch much more efficient when forwarding frames.

Forward/filter Decisions

Forwarding decisions of a layer 2 switch are exclusively based in the destination MAC Address of the incoming frame. The switch looks for the MAC address in the CAM table and forwards the frame out the port associated with it. If it doesn't find an entry for the destination MAC address, the switch floods the frame out all ports associated with the VLAN of the frame that was received, excluding the port where the frame was received. This is called *unknown unicast flooding*. Similarly, broadcast and multicast frames are also flooded.

Loop Avoidance

This is the mechanism by which the switches prevent a frame from taking more than one path to a destination. If a loop formed, the flooded frame would end up being replicated and retransmitted over and over in the looped path, creating a real mess in our networks. The Spanning Tree Protocol was developed to prevent switching loops from happening. We will devote a lot of time to very powerful options configurations of STP and the variations of STP. Pay close attention to this topic as STP is an extremely important and incredibly powerful protocol. The means that provide loop avoidance are explained in great detail later in this section.

Switching Tables

More advanced switches utilize several tables for the switching process and not just the CAM table. The tables are designed for Layer 2 or multi-layer switching and are maintained in very fast memory in order to be able to check several fields in the tables at the same time.

Content-Addressable Memory

As mentioned previously, this is also known as the MAC Address or Forward Table. It is used to register and associate a MAC address with a specific port on which the device or devices are last known to reside. When a frame is received in a port, the source MAC address, the VLAN ID and the time stamp is associated with the port that received the frame. When a host moves to another port, the CAM table will be updated with the new entry, with the corresponding time stamp. After an established period of time, the older entry will be deleted. If a frame is received and the source address is already in the CAM table, only the time stamp will be updated. To deal with the size of the CAM tables and also to optimize resources, if a switch doesn't hear from a host after a period of time, its CAM entry will be deleted. By default, this period is 300 seconds, but this can be changed with the **MAC address-table aging-time** *seconds*, configuration command, as follows:

```
Switch(config)# MAC address-table aging-time seconds
```

You can also configure static CAM table entries with the following command:

```
Switch(config)# MAC address-table static MAC-address vlan vlan-id
interface type mod/num
```

When a switch detects a MAC address that is already registered as belonging to another port, the switch purges the old record. This is a correct procedure, because MAC addresses are unique and shouldn't be available on more than one port. If the switch detects a MAC address has been learned in alternating ports, an error message is generated and the address is flagged as flapping between interfaces. There are several causes for this and we will see the mechanism to prevent it in the Spanning Tree Protocol section.

Ternary Content-Addressable Memory (TCAM) – think of TCAM as the mechanism, a table, implemented in hardware that allows a multilayer switch (MLS) to match all ACLs for security and QoS features. Most switches have several TCAMs in order to be able to match all inbound and outbound security and QoS ACLs in a single lookup, with the resulting L2 or L3 forwarding decision.

The Cisco Catalyst Switch has two components of TCAM operation:

- **Feature Manager (FM)** – This compiles the Access Control Entities (Access Control List entities are ACLs statements) into the TCAM table. At this point the TCAM can be consulted (at wire speed) and can forward packets at wire speed.
- **Switching Database Manager (SDM) –** The SDM is used to partition and tune the TCAM partitions. Some switches don't allow this function.

Besides the table-lookup operation, the TCAM is also designed to allow a more granular, abstract operation. This feature is provided by a ternary combination that is defined by the binary values and a mask value, resulting in keys or entries with three input values: 0,1,X (doesn't matter) bit values.

Entries in the TCAM consist of Value, Mask and Result (VMR) combinations. The consultation process when a packet or frame is received goes like this:

- Certain fields within the frame/packet (MAC or IP address, TCP or UDP port numbers) header are matched to the TCAM value and mask and this yields a result that is used for the forwarding decision.

In essence the TCAM is organized by masks, and after a mask is matched, there are 8 values used for security and QoS considerations. These values and mask pairs can be evaluated simultaneously with the use of specialized hardware, to produce a result, and the final forwarding decision. The amount of masks that can be compiled into the TCAM varies in different equipment, but the values are always eight per mask.

The TCAM is a hardware chip and therefore has a limited amount of memory for entries, therefore there are instances where it overflows. This generates a log error message to alert the network administrator. This can result in some packets being forwarded utilizing the CPU, which means the "wire speed" provided by the specialized hardware, Application Specific Integrated Circuits, will not be achieved for those packets/frames. In other words, it will slow down the forwarding of packets.

The TCAM is organized by masks, and each unique mask has eight value patterns associated with it. The value patterns are 134 bits long and they consist of source and destination addresses and other information relevant to the layer 2 or 3 protocol used and the ACL type that is being compiled to the TCAM.

Switching Table Commands

There are several commands that allow us to inspect the contents of the different switching tables.

One reason to check the CAM would be to find out about the location of an end device using its MAC address. To do this, you can use the following EXEC command to find out the port on which a certain MAC address was learned:

```
show MAC address-table dynamic [address MAC-address | interface type
mod/num | vlan vlan-id]
```

```
Switch# show mac address-table dynamic address 1111.1111.1111
Mac Address Table
-------------------------------------------
Vlan    Mac Address         Type        Ports
-------------------------------------------
5       1111.1111.1111      DYNAMIC     Fa1/0/1
Total Mac Addresses for this criterion: 1
Switch#
```

Figure 1: Output from the show MAC address-table Command

From this output you can see the host with MAC address 1111.1111.1111 is connected to port Fast Ethernet 1/0/1, and is in VLAN 5. The "dynamic" specifies that the switch learned the MAC address dynamically as opposed to a MAC address that has been manually configured into the switch. This command is useful when you need to detect if a host has L2 connectivity to an uplink switch. That is a switch that is closer to the network core. You can use it to know where the host was learned if it was learned.

The other possibility is you might need to know what MAC addresses have been learned on certain ports. For that issue the following show EXEC command:

show MAC address-table dynamic interface *type number*

The actual output in the switch is as follows:

```
Switch# show mac address-table dynamic interface fastethernet1/0/2

Mac Address Table
-------------------------------------------------------------
Vlan    Mac Address     Type        Ports
-------------------------------------------------------------
7       2222.2222.2222  DYNAMIC     Fa1/0/2
Total Mac Addresses for this criterion: 1
Switch#
```

Figure 2: Showing Learned MAC Addresses by Interface

From this output you can see that host with MAC address 2222.2222.2222 is connected to port Fast Ethernet 1/0/2 which belongs to VLAN 7. If you see more than one MAC address that means the port is connected to another switch or a hub with more hosts connected to them. As with the previous command this can be used to see the host or list of hosts that have been learned in a port. A common use is to track host associations of wireless clients using APs connected to certain switch ports. Suppose you use **show MAC address-table dynamic interface** fastethernet1/0/2 command and there is an AP connected to that port. You will see the MAC address of the AP and all wireless clients associated to the AP.

As your network grows it might be necessary to know how many hosts are connected to a certain switch. You can find this information with the **show MAC address-table count** command. The output is shown here:

```
Switch# show mac address-table count

Mac Entries for Vlan 1:
---------------------------------
Dynamic Address Count   : 0
Static Address Count         : 0
Total Mac Addresses          : 0

Mac Entries for Vlan 2:
---------------------------------
Dynamic Address Count   : 89
Static Address Count         : 0
Total Mac Addresses          : 89

Mac Entries for Vlan 580:
---------------------------------
Dynamic Address Count   : 600
Static Address Count         : 0
Total Mac Addresses          : 600
Total Mac Address Space Available: 4810

Switch#
```

Figure 3: The show MAC address-table count Command

CAM table entries can also be deleted manually with the following EXEC command:

```
Switch# clear MAC address-table dynamic [address MAC-address |
interface type mod/num | vlan vlan-id]
```

You clear out the CAM entry to allow new MAC addresses to be learned immediately. Waiting for the entry to age out could be unacceptable at times because of the need to connect a new host and provide connectivity immediately. This is especially true when changing switch ports of a heavily used server.

TCAM Operation

In essence, there is no need to configure TCAM tables, as they are populated with ACEs (Access Control Entries) as you create ACLs (Access Control Lists). The only important consideration is that as your network grows and you implement QoS and security features the TCAM tables might overflow. When this happens, a log message is generated and the overflow is flagged.

If the TCAM overflows, the ACL will simply be processed by the CPU. Again, this means that the packet won't be forwarded at wire speed with the use of Application Specific Integrated Circuits (ASICs).

The Switch Forwarding Process

When a switch receives a frame in one of its ports, it places the frame in one of the port's ingress queues. At this point, the switch has three decisions to make: Where to forward the frame, whether to forward the frame or not, and how to forward the frame. All decisions are made simultaneously using different portions of the switch hardware. Basically, the switch needs to find the egress port, and examine the forwarding policies concerning Quality of Service (QoS) (the priority the frame has to be sent in comparison to others) and security. Here is a description of the three separate processes and the mechanism in charge of each decision:

- **L2 forwarding table** – The frame's destination MAC address is looked up in the CAM table; if it is found, the frame will be sent to the appropriate egress port with its VLAN id. If it is not found, the frame will be flooded out all ports on the VLAN it was received, except the port it was received.
- **Security ACLs** – These are compiled into the TCAM and are used by the switch hardware to identify and make decisions based on criteria including IP address, MAC address, protocol types and layer 4 port numbers (applications).
- **QoS ACLs** – These ACLs contain markings or QoS parameters that define and police the traffic flow. The idea here is to give specific traffic (such as voice and video) priority over other data flows that are more resistant to network delays. These ACLs also contain information used to mark outbound frames. MLS switches have dedicated hardware for this operation. This provides the ability to process frames simultaneously, in parallel, and at wire speed.

The egress queues are serviced based on the importance or priority assigned by the network administrator and/or designer; these criteria, in turn, are based on the time criticality of the communication type. Quality of Service is part of the CCVP and CCIP certifications and is an important part of the CCIE Routing and Switching certification.

Multilayer Switch Operation

Layer 3 switches are powerful. They forward frames based on layer 3 and layer 4 information contained in packets. This is called Multilayer switching (MLS). One of the key differences between L3 switches and (most) routers is that L3 switches route packages based on hardware, just like they do in the L2 switching. L3 switches can perform packet switching up to ten times faster than traditional L3 routers.

The use and performance advantages of the MLS come at a cost: they are expensive. Other than that they are in general a great improvement and a major upgrade over traditional routers and layer 2 switches within the LAN. They perform all the functions of a router, a regular layer 2 switch plus another function at layer 4, and can interact in the network with such devices.

Cisco Catalyst Switches perform packet switching , or L3 switching, using a router processor or L3 engine. This processor is in charge of downloading routing information to the hardware itself.

Multilayer Switching Methods

There are two types of multilayer switching: Route caching and topology based.

- **Route caching** – The first generation of multilayer switching (MLS), also called NetFlow switching (now considered legacy). Route caching capable devices utilize a route processor and a switching engine. It requires several flows for every port in use, making it a processor intensive method and it doesn't exactly provide hardware speed routing. This method is often referred to as the "route once, switch many" method.
- **Topology-based** – The second generation of MLS and a definite improvement over the first. Cisco's implementation is called Cisco Express Forwarding (CEF), and requires special hardware chips, which is why it is not available in all L3 switches. CEF is very scalable and requires less main CPU resources than the Route caching mechanism. This optimization is achieved by the use of application specific integrated circuits (ASIC), to forward packets and make several decisions at the same time, providing packet forwarding at wire speed. CEF has two major components, the Forwarding Information Base (FIB) and the Adjacency Table. The FIB is practically another form of a routing table, containing the traditional routing information (destination networks, network masks, next hop address, etc). The efficiency in this mechanism is achieved because it is maintained in hardware, giving the speed of L2 switching to L3 and L4 switching. The routing information contained in the FIB is updated automatically when the network changes (hence the name topology-based).

The path a packet follows when switched in a L3 switch is determined by the multilayer mechanisms, and they are all performed simultaneously:

- **L2 forwarding table –** As usual, the destination MAC address is used to determine the path. If an IP is encapsulated in the frame, the switch sends the frame to a L3 port of the switch, so it can be processed at L3.
- **L3 Forwarding table –** The FIB is consulted using the destination IP address, just like a regular routing table. After the longest match is found, the next hop address is obtained and the packet is sent out the appropriate interface. The FIB table also has the MAC addresses of each next hop, and the switch egress port, which prevents the switch from looking up the information every time a packet needs to be forwarded (remember the FIB is updated based on topology changes).
- **Security ACLs** – These are compiled into the TCAM and are used by the switch hardware to identify and make decisions based on criteria including IP address, MAC address, protocol types and layer 4 port numbers (applications).
- **QoS ACLs –** These are also compiled into the TCAM and the associated values allow the switch to perform traffic classification, policing and marking at wire speed in a single table lookup.

The packet is then placed in the appropriate egress queue on the appropriate egress port.

After this point, the procedure is basically the same as a regular switch and router, with the efficiency provided by the hardware-based architecture. The next hop IP address obtained from the FIB table has an association with a layer 2 address, which is the address that will be used to forward the frame to the next hop. This changes the time-to-live (TTL) values in the L3 packet, therefore the checksum must be recalculated. The TTL is a counter within the packet that counts down. After every layer 3 hop, the TTL decreases until it reaches 0. If the packet has a TTL of 0 and reaches another layer 3 device before getting to its destination, the packet is discarded. This prevents packets from traversing a network forever. Since the frame is also changed in this process, the layer 2 checksum must be recalculated too. Once again, this calculation is performed in hardware.

There are several exceptions to the MLS process that take place in the L3 switch. For CEF to process the packet with the advantages of hardware processing, the packet must be MLS-ready, with no further decisions required to be made about it. Packets that require a more detailed handling are sent to the switch CPU for process switching, or the traditional switching performed in software. The following packets can't be processed by CEF:

- ARP request and replies
- IP packets requiring a response from a router (TTL has expired, MTU is exceeded, etc)
- IP broadcasts and multicasts that will be relayed as unicast (first reply packet in EIGRP adjacency formation, DHCP requests, IP helper-address functions).
- Routing protocol updates
- CDP packets
- Packets that require encryption
- IPX routing advertisements
- Packets triggering NAT
- The MLS process only works for IP and IPX. Other non-IP and non IPX protocol packets (AppleTalk, DECnet, and so on) are not processed using MLS methods.

Switch Configuration

Ethernet

Ethernet is a LAN technology defined by the Institute of Electrical and Electronics Engineers (IEEE) 802.3 standard. It operates with CSMA/CD, as described earlier, which requires that each station listen to the wire and wait until no transmissions are being made before being allowed to transmit. This is half-duplex operation, and as explained before, it is very inefficient. A switch takes care of this problem by allocating a dedicated bandwidth to each of its ports.

A switch can remove the possibility of collisions, and stations don't have to listen before transmitting, but can actually transmit and receive at the same time. This mode of operation is called ***full duplex***. Full duplex increases network performance dramatically, effectively doubling the net throughput. For example, in 10Mbps Ethernet, you get 10Mbps in each direction, for a total combined theoretical throughput of 20Mbps on each port.

Fast Ethernet

Fast Ethernet is defined in the IEEE 802.3u standard. Fast Ethernet maintains the same layer 2 (frames remain the same) components, and merges them with new layer 1 components (physical cabling and wiring). The whole operation is essentially the same, but the speed is increased to 100Mbps (half duplex).

Fast Ethernet is currently often found in the access and distribution layers of the campus network, when there are no other higher speed links available. Fast Ethernet is most commonly used to connect user stations to the networks (access layer switches) and also to increase throughput in the connection to servers.

Fast Ethernet can also support full duplex operation, giving network devices a combined theoretical throughput of 200Mbps. This maximum speed is only possible when a device is directly connected to a switch port and all devices (router, endpoint, another switch) support full duplex operation.

Fast Ethernet is fully compatible with older Ethernet technology. Switch ports are often referenced as 10/100Mbps, to denote compatibility with both specifications. This capability is provided by the possibility of auto-negotiation. When Auto-negotiation is configured at each end, devices will select the maximum possible speed (of the slowest device) and the duplex operation. Switches will exchange information to determine the duplex mode. If for any reason this process fails, both switches will use half duplex, the default switch port setting.

When auto-negotiation is not set, or when one side is hard-coded and the other side is set for auto-negotiation, a duplex mismatch can occur. This is the cause of major network problems, such as communication slowdowns, which will be explained later. This can also cause other network instabilities because if one switch is in full duplex mode, it won't stop to hear if there are transmissions, while the other end, in half duplex, does, and keeps waiting for the other end to stop transmitting.

The priority in the auto-negotiation process is as follows:

Priority	Ethernet Mode
7	100BASE-T2 (full duplex)
6	100BASE-TX (full duplex)
5	100BASE-T2 (half duplex)
4	100BASE-T4
3	100BASE-TX
2	10BASE-T (full duplex)
1	10BASE-T

Figure 4: Ethernet Auto-Negotiation Priorities

Cisco recommends that switch ports are configured with speed and duplex mode manually. This prevents the very troubling duplex mismatch and link speed issues. You must remember to set both ends with the same speed and duplex mode. To configure the duplex mode use the **duplex [auto** | **full** | **half]** interface configuration command. The command is pretty much self-explanatory.

Gigabit Ethernet

Gigabit Ethernet is defined in the IEEE 802.3z standard. It provides 1000Mbps (1Gbps) using the same data link layer (same frame format) and new layer 1 capabilities and specifications. This means new connections and hardware but interoperability with older standards, like fast Ethernet and Ethernet. The standard was the product of the merger of the 802.3 standard (layer 2 characteristics) and the American National Standards Institute (ANSI) X3T11 FiberChannel (provided the layer 1, hardware mechanisms).

In campus networks Gigabit Ethernet is commonly found connecting end devices such as servers to switches and switch to switch uplinks. It can be found in all three layers of the campus network.

10-Gigabit Ethernet

Just as Fast Ethernet and Gigabit Ethernet, 10-Gigabit Ethernet preserves all layer 2 characteristics (frame format and size, MAC addresses). 10-Gigabit Ethernet is also known as 10GbE and is defined in the IEEE 802.3ae standard. It operates exclusively in full duplex. Many large Enterprise networks and ISPs are running 10-Gigabit in their distribution and core.

Switch Port Configuration

To enter switch configuration mode, enter the following command:

```
Switch(config)# interface type module/number
```

Type refers to the Ethernet types defined: Ethernet, fastethernet, gigabitethernet, tengigabitethernet or vlan, if we are configuring remote access to the switch.

Module is the slot in which the interface is within the switch and number is the actual port number. Here is an example:

```
Switch(config)# interface fastethernet 0/1

Switch(config)# interface gigabitethernet 1/2
```

There we enter interface configuration mode for the fastethernet module 0 port 1. There are certain types of switches, like the Catalyst 3750 that can be stacked with other switches of the same family. In that scenario you might see interface ethernet 2/0/10. That means the switch in position 2 of the stack, module 0 and port 10.

In the second line after pressing enter we go to the interface configuration mode for the Gigabit Ethernet module 1 port 2.

You can also enter interface configuration mode for more than one port or a *range* of ports with the **interface range** global configuration mode command. This command will prove very useful when implementing a lot of identical switching functions.

This is the command to be used in global configuration mode:

```
Switch(config)# interface range fastethernet 1/0/1 – 48
```

This will select the switch stacked in position one, module 0 and fast ethernet ports from 1 to 48. You can also select a group of ports that are not in the same switch or not exactly in a range, by dividing the ports you want by commas. Here is an example:

```
Switch(config)# interface range fastethernet 1/0/1, fastethernet 1/0/3,
fastethernet 1/0/5, fastethernet 1/0/24 – 36
```

The previous command will enter interface configuration mode for ports 1, 3, 5, and 24 to 36 of stacked switch 1, slot 0. Notice the "range" is always a closed range, meaning both ends are included. In this case, port 24 and 36 will be included in the range.

Another very important and useful feature is the range macro. It allows you to save certain ranges that you will likely need to use often (like all ports belonging to certain VLAN).

```
Switch(config)# define interface-range macro-name type module/number [
type module/ number ...] [type module/first-number - last-number] [...]
```

Macro-name is the name you will assign to the macro. The other range definitions are exactly as you would enter in an interface range command. An example:

```
Switch(config)# define interface-range vlan10ports fastethernet 1/0/1,
fastethernet 1/0/3, fastethernet 1/0/5, fastethernet 1/0/24 - 36

Switch(config)# interface range macro vlan10ports
```

This is very useful when you need to configure non-contiguous ports with the same settings, like VLAN associations or port aggregation in the form of EtherChannel bundles. At that point you will enter interface configuration mode for the range defined with the define interface-range command.

Describing Ports

You can describe ports using the description interface configuration mode command. This is helpful to network administrators when they want to identify the function of the port, describe the device attached to the port, or when taking notes during maintenance outages. The description is locally significant to the port.

In the following example we will add a description to the interface fast Ethernet 0/2, saying "Application Server, Building C":

```
Switch(config)# interface fastethernet 0/2

Switch(config-if)# description Application Server, Building C
```

Port Speeds

You can manually assign specific port speeds to some Ethernet interfaces using interface configuration command speed. Fast Ethernet ports can be assigned speeds of 10Mbps, 100Mbps and Auto, the default, for auto-negotiation. Gigabit Ethernet GBIC ports are always set to 1000Mbps. It is not possible to configure them with another speed. 1000BASE-T ports, on the other hand, can be configured as 10, 100 and 1000Mbps and Auto, the default. Here is an example:

```
Switch(config-if)# speed {10 | 100 | 1000 | auto}
```

If a 10/100 or a 10/100/1000 port is configured with "Auto" as its speed, both speed and duplex mode will be auto-negotiated. This can cause problems on your network.

Port Duplex Mode

You can assign the link mode of specific switch ports. The possibilities offered in the configuration are half duplex, full duplex and auto-negotiation. Auto-negotiation is only allowed in Fast Ethernet and Gigabit Ethernet ports. In this mode, the ports try to use full duplex operation first, and if that fails, fall back to half duplex mode. Auto-negotiation starts every time one of the links' status changes. Cisco recommends that you set both speed and duplex mode manually in each switch port. Here are a couple configuration examples:

```
Switch(config)# interface fastethernet 0/1
Switch(config-if)# speed auto
Switch(config-if)# duplex auto
Switch(config-if)# interface fastethernet 0/10
Switch(config-if)# speed 100
Switch(config-if)# duplex full
```

In this case Fast Ethernet 0/1 was configured to auto-negotiate both speed and duplex mode. Fast Ethernet 0/10 was assigned a link speed of 100Mbps and full duplex operation.

Errors on Switch Ports

Switches have mechanisms to detect errors in practically any possible way. You can configure a switch port to be shut down automatically when certain error conditions are detected. Network management applications can also be used to inform network administrators of the occurrence of certain types of errors.

When a switch detects an error in a port, it is put in the **errdisable** state and is shutdown. This can be manually tuned so that the switch only puts the port in errdisable and shut down condition for certain types of errors that the network administrator determines is important. You can also configure this function with the following command:

```
Switch(config)# errdisable detect cause [all | cause-name]
```

"*Cause-name*" is one of the several possibilities that could occur. They are the following:

- **all —** Detects every possible cause.
- **arp-inspection —** Detects errors with dynamic ARP inspection.
- **bpduguard —** Detects when a spanning-tree bridge protocol data unit (BPDU) is received on a port configured for STP PortFast.
- **channel-misconfig —** Detects an error with an EtherChannel bundle.
- **dhcp-rate-limit —** Detects an error with DHCP snooping.
- **dtp-flap —** Detects when trunking encapsulation is changing from one type to another.
- **gbic-invalid —** Detects the presence of an invalid GBIC or SFP module.
- **ilpower —** Detects an error with offering inline power.
- **l2ptguard —** Detects an error with Layer 2 Protocol Tunneling.
- **link-flap —** Detects when the port link state is "flapping" between the up and down states.

- **loopback —** Detects when an interface has been looped back.
- **pagp-flap —** Detects when an EtherChannel bundle's ports no longer have consistent configurations.
- **psecure-violation —** Detects conditions that trigger port security configured on a port.
- **rootguard —** Detects when an STP BPDU is received from the root bridge on an unexpected port.
- **security-violation —** Detects errors related to port security.
- **storm-control —** Detects when a storm control threshold has been exceeded on a port.
- **udld —** Detects when a link is seen to be *unidirectional* (data passing in only one direction).
- **unicast-flood —** Detects conditions that trigger unicast flood blocking on a port.
- **vmps —** Detects errors when assigning a port to a dynamic VLAN through VLAN membership policy server (VMPS).

The most common violations are security violations such as a MAC address learned in a port with the maximum amount of MAC addresses learned through it. Another very common cause is the BPDUguard violation. This one will be explained in detail later in this domain.

By default, switch ports need to be manually re-enabled when they are put in the errdisable state by issuing shutdown and then no shutdown on the offending switch port using the CLI. You can configure a switch so it doesn't put the port in the errdisable state when certain situations happen. The switch actually puts the port in errdisable state and then it re-enables the port, after a specified period of time. 300 seconds is the default time, but it can also be modified. Here are the commands that achieve this:

```
Switch(config)# errdisable recovery cause [all | cause-name]

Switch(config)# errdisable recovery interval seconds
```

In the first line, the cause-name is the list of possible error conditions described above, and all means exactly that: all conditions will be recovered after the recovery interval. The second line is to specify the amount of time the port will remain in the errdisable state. As stated earlier, the default period is 300 seconds.

To troubleshoot port states, speed and duplex mode use the show interfaces EXEC mode command. It will show you a lot of information, such as layer 1 and 2 operational status, speed of the link, duplex mode, MTU, encapsulation, Ethernet type and several other errors.

Virtual LANs

A virtual LAN or VLAN is switched network provided by the logical segmentation of a network, consisting of a single broadcast domain, regardless of the physical location of the hosts. VLANs have the same characteristics as a physical LAN as if they were on their own separate switch. If a switch port belongs to a VLAN, unknown destination unicasts, multicasts and broadcasts for that VLAN (network segment) are sent out that port, just as if it were a physical LAN. Any switch port can be assigned to a VLAN. Unicast, multicast and broadcast sent from a port assigned to certain VLAN will only be flooded to ports belonging to the same VLAN. This segmentation of broadcast domains helps to make the network more efficient.

In order for a host to communicate with another outside its VLAN, the packet must be routed with a router or Layer 3 switch. As an independent logical network, it contains its own Management Information Base (MIB) information and has the capability to support its own implementation of spanning tree protocol. Spanning tree protocol will be covered in detail later.

VLANs have come to solve all the problems associated with the problems of the flat network topology, which is nothing more than a big switched network, a single broadcast domain. This type of network is very inefficient and not exactly scalable. In the flat network, broadcasts are received by all hosts, dramatically affecting network performance by consuming unnecessary bandwidth and provoking an unnecessary use of router and switch processing resources. VLANs segment the network so hosts that don't require constant access to others stay in different VLANs, and are prevented from receiving broadcasts that might not be necessary for them. Equally, hosts in the same VLANs are generally performing similar functions and therefore require and work constantly with the same resources. For a host to communicate with another in a different VLAN, routing must take place, exactly as it would be required if the communication was between hosts in different physical networks.

As stated, any port in a switch can belong to a VLAN. You can configure a port for a certain type of VLAN membership, which determines the kind of traffic the port will carry. These memberships are static access, dynamic access, trunk and tunnel (dot1q tunnel).

A static-access port can belong to one VLAN and is configured manually (hence the term static). By default, all ports belonging to VLAN 1, the native or administration VLAN, are set to type Ethernet VLAN and have a maximum transmission unit (MTU) of 1500. It is also important to know that VLAN memberships are kept in hardware, which provides great efficiency because it doesn't require complex table lookups. But it also means that you can configure a finite amount of VLANs on a switch. Every switch model has a maximum limit as to the number of VLANs that it can maintain.

To assign a VLAN, you must first create it. Issue the vlan vlan-id global configuration mode command, to create the VLAN. Be aware that Cisco IOS Switches support VLANs from 1 to 1005, with 1002 to 1005 reserved and automatically created for special uses. Cisco IOS switches can also use extended VLANs, with IDs from 1006 to 4094. In order for the switch to accept VLANs in this range, VLAN Trunking Protocol (VTP) must be configured and the mode should be set to transparent. VTP will be covered in detail later. After you create the VLAN, you assign the port to the VLAN. Here is an example:

```
Switch(config)# vlan 2
Switch(config-vlan)# name Engineering
Switch(config-vlan)# vlan 3
Switch(config-vlan)# name Accounting
Switch(config)# interface fastethernet 0/1
Switch(config-if)# switchport
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 2
Switch(config-if)# interface fastethernet 0/2
Switch(config-if)# switchport
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 3
```

Figure 5: Assigning Ports to a VLAN

In the above example, we created VLAN 2 and 3, and then told the switch the fast Ethernet 0/1 was going to be a switch port (not used for routing, this is only necessary in multilayer switches) with the **switchport** interface configuration mode command. Then we configured the interface to be an access port with the **switchport mode access** interface configuration mode command, and finally assigned the port to VLAN 2 with the **switchport access vlan** 2, interface configuration command.

Dynamic VLANs provide membership to hosts based on their MAC addresses. VTP and VLAN Membership Policy Server (VMPS) are required to use this type of VLAN assignment.

Dynamic VLANs and their configuration are out of the scope of the SWITCH exam and course, but it is a very important feature worth learning. Cisco has a very detailed introduction to the concept of VMPS, along with step-by-step instructions on setting one up, at their website.

Some considerations you must take before deploying VLANs are: the number of VLANs depending on traffic patterns, application types, segmentation of common workgroups and network management requirements.

There are also important design considerations that must be observed. Cisco recommends that VLAN data shouldn't move beyond the distribution layer switch. If you are dragging VLANs between distribution blocks, you have a poorly designed network. Also, VLANs should never reach the network's core according to current best-practices. In essence, we must design our VLANs to keep broadcast and multicast traffic away from the core. Core switches are meant to switch packets at very fast speeds. By eliminating broadcast traffic, we make our core extremely efficient.

The IP addressing scheme must allow for all hosts in the VLAN. That is, if you have a 255.255.255.0 network mask, you shouldn't require more than 254 hosts in that VLAN. Also consider network continuity, since it allows more efficient route summarization, which allows for a much more efficient routing and general resource utilization.

There are two major VLAN designs: **End-to-End** and **Local.** They follow the 80/20 rule and the 20/80 rule respectively.

1. **End-to-End VLANs** will span the entire network regardless of the physical location of the user. This type of VLAN should be designed with the 80/20 rule in mind, which means that 80% of the traffic will be local and 20% will traverse the network core to a remote destination. End-to-End VLANs must group users with common network resource requirements. They must be accessible from every access layer switch to accommodate mobile users.

   End-to-End VLANs are rarely found in today's enterprise networks because their data must be allowed to traverse the core network. That means broadcasts and multicast traffic will pass through it, creating the possibility of a nasty broadcast storm or switching loop that spans the entire network. It also unnecessarily uses network resources in the core, which can cause major network disruptions. These types of problems are very difficult to troubleshoot because of their potentially huge scope. Cisco recommends avoiding End-to-End VLANs whenever possible because of these possible issues.

2. **Local VLANs** consider the 20/80 rule, 20% of the traffic remains local while 80% will traverse the network core. Local VLANs group users by physical location, and their size varies from one switch to an entire building. This kind of VLAN allows for easier management and better scalability.

Trunk Links

Trunk links are switch ports that allow transit from several VLANs. Trunk links are useful and most often found connecting switches to switches and switches to routers. The trunk can support one, several or all active VLANs in the switch. Cisco switches support trunking on the fast Ethernet interfaces and up. You cannot trunk a 10 Mbps Ethernet interface. When creating trunks that transport multiple VLANS, switches need a method to identify data from different VLANs passing through the trunk. This is known as VLAN tagging. Switches on both ends must have the same method of identification. Cisco uses 802.1Q protocol and ISL protocol to identify these frames with their corresponding VLANs.

Inter-Switch Link Protocol (ISL) is Cisco's proprietary protocol that encapsulates Ethernet frames between a header and a trailer, adding 26 bytes in the header and 4 bytes in the trailer. The source VLAN is identified in the 15-byte VLAN ID field in the header, while the trailer contains a cyclic redundancy check for the new re-encapsulated frame. The switch doesn't add ISL encapsulation when the frame arrives at its destination port (ISL is only used in trunk links). ISL is only supported in older IOS versions. It is no longer supported in Cisco Catalyst Switches. ISL is almost never found on networks today and newer switch IOS trains have completely removed the use of the tagging protocol.

IEEE 802.1Q Protocol is the industry standard used to identify VLANs passing through a trunk link. 802.1Q doesn't encapsulate the Ethernet frame as ISL does; it simply "tags" the frame with the VLAN information. This method is also called frame tagging, single tagging or internal tagging. The VLAN information is tagged next to the address field. 802.1Q adds 4 bytes to the Ethernet frame.

802.1Q also uses the concept of a ***native VLAN***, which is basically a way to allow hosts to be connected to a trunk link. When a frame is received in the switch and sent to a host connected to a trunk link, the switch forwards the frame out the trunk port without adding any tags. When traffic is sent along a trunk that belongs to a native VLAN, those frames pass through the trunk untagged.

Since both VLAN identification protocols increase the size of the Ethernet frames, this creates a problem related to the MTU, maximum transmission unit, which cannot exceed 1518bytes. ISL has Cisco proprietary mechanisms to deal with this issue, and there is an IEEE standard, the IEEE 802.1ac, that extends the maximum frame size to 1522 bytes.

Dynamic Trunking Protocol (DTP) is another Cisco proprietary protocol that negotiates a common trunking mode between two connected switches. The negotiation involves the encapsulation (ISL or 802.1Q) and whether the link becomes a trunk or not. If the two switches are ISL-capable this will be the encapsulation preferred.

DTP is activated by default on Cisco switch ports, and DTP frames are sent out every 30 seconds. DTP is turned off when you configure a port as an access port, or set it to non-negotiate with the interface configuration mode command **switchport nonnegotiate.**

You should disable DTP in ports connected to non-trunking routers, firewall interfaces or hosts. This saves bandwidth and switch resources. Switches can only perform auto-negotiation if both belong to the same VLAN trunking protocol domain (VTP will be covered later) or if the VTP domain hasn't been defined.

Trunk Configuration

Switch ports are access ports by default, but they try actively to form a trunk (switchport mode dynamic desirable) as long as the other agrees on forming the trunk. The following are the commands required to configure a port as a trunk:

```
Switch(config)# interface type mod/port
Switch(config-if)# switchport
Switch(config-if)# switchport trunk encapsulation {isl | dot1q | negotiate}
Switch(config-if)# switchport trunk native vlan vlan-id
Switch(config-if)# switchport trunk allowed vlan {vlan-list |
all | {add | except | remove} vlan-list}
Switch(config-if)# switchport mode {trunk | dynamic {desirable | auto}}
```

Interface type mod/port global configuration mode command enters interface configuration mode. **Switchport** defines the port as a layer 2 port, meant for switching.

Switchport trunk encapsulation {**isl** | **dot1q** | **negotiate**} defines the encapsulation. If negotiate is chosen, and both formats are available, ISL is preferred. Negotiate is the default.

Switchport trunk native vlan *vlan-id* defines the native VLAN. Be careful with this option! The native VLAN is crucial information for the trunk and its negotiation. Improper configuration can result in VLAN leakage and several types of security vulnerabilities that will be explained in detail in the security section of this guide.

Switchport trunk allowed vlan {*vlan-list* | **all** | {**add** | **except** | **remove**} *vlan-list*}. By default all active VLANs are allowed in a trunk. That can be changed with this command. vlan-list is a list of VLANs, and is defined exactly as we defined a range with the interface range command: to add VLANs 1, 3, 5, 7 and from 50-100, use: 1, 3, 5, 7, 50-100. Since all active VLANs are allowed by default, use **add** or **except** to specify a range of VLANs not allowed to pass through the trunk. Use **remove** to remove one or a list of VLANs from the allowed list of VLANs. Remember all active VLANs are allowed by default. An active VLAN is one that has ports assigned to it in the switch.

The trunk can be formed with any of the following commands:

- **Switchport mode trunk** interface configuration command: The port automatically becomes a trunk. The encapsulation is negotiated by default, but if you want to prevent this negotiation from taking place, manually configure the encapsulation. Remember that if both encapsulation protocols are available, ISL will be preferred. Also remember current versions of the IOS don't support ISL. DTP frames are sent out by default if the port is not configured with the **switchport nonnegotiate** interface configuration command.
- **Switchport mode dynamic desirable** interface configuration command: This is the default. This port will try to actively negotiate a trunk with the other end. The only time this port won't become a trunk is when the other end is configured as an access port.
- **Switchport mode dynamic auto** interface configuration command: A port with this configuration will form a trunk but it will not actively attempt to do it. This means that the port will form a trunk with another port configured as a trunk or as dynamic desirable, but not with another dynamic auto or an access port.

It is a Cisco recommended practice that you configure all non-trunking ports as access ports. It is also a best practice to avoid the "auto" settings, or any other setting that involves some kind of automatic negotiation. It widens the platform of attack (gives possible attackers new vulnerabilities that can be exploited) and can result in suboptimal network performance if the negotiation is affected and the best options are not agreed.

Troubleshooting VLANs and Trunk ports

There are several useful commands when verifying and troubleshooting VLANs and trunk ports. Here are the ones you need to know for the exam:

The **show interface** *type mod/number* **switchport** interface configuration mode command provides a lot of information that should be used for VLAN configuration verification. Here is an example:

```
Switch# show interface fastethernet 0/2 switchport
Name: Fa0/2
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Voice VLAN: none (Inactive)
Appliance trust: none
Switch#
```

Figure 6: VLAN Verification

Let's run through some of the key points in this output:

- **Switchport: Enabled**, refers to the layer 2 operational status of the port (for MLS).
- **Administrative Mode: Dynamic auto**, refers to the trunking mode configured on the port.
- **Operational Mode: static access**, refers to the actual operation mode of the port. In this case, even after this port was configured as dynamic auto, the port became an access port. This is because of several causes. The most common is when the port at the other end is an access port or was also configured as dynamic auto (remember this kind of dynamic negotiation doesn't actively try to form a trunk, but forms one if the other end is trying, for which it must be configured as a trunk or as dynamic desirable).
- **Administrative trunking encapsulation: dot1q**, verifies that the port is configured to use 802.1q as its trunking encapsulation.
- **Negotiation of trunking: on**, verifies that DTP frames are being sent every 30 seconds on the interface.
- **Trunking Native mode VLAN: 1 (default)**, simply verifies that the native VLAN for 802.1q trunking is 1, which is the default.
- **Trunking VLANs enabled: All**, this is also a default setting. All VLANs are currently allowed on the trunk port. No exceptions have been configured.

Some of the other information provided with this output will be examined later in the guide.

Another very useful command is the show vlan id vlan-id EXEC command. Here is a sample output:

```
Switch# show vlan id 2
VLAN  Name                             Status    Ports
----  -------------------------------- --------- -------------------------------
2     Engineering                      active    Gi2/1, Gi2/2, Gi2/3, Gi2/4
                                                 Gi4/2, Gi4/3, Gi4/4, Gi4/5
                                                 Gi4/6, Gi4/7, Gi4/8, Gi4/9
                                                 Gi4/10, Gi4/11, Gi4/12

VLAN  Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
----  ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------

2     enet  100002     1500  -      -      -        -    -        0      0

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------

Switch#
```

Figure 7: Output from the show vlan id Command

The output shows that various ports on module 2 and 4 are configured to run on VLAN 2.

You can also see a lot of valuable information and verify configuration with the **show interface** *[type mod/number]* **trunk** EXEC command. Here is a sample output:

```
Switch# show interface fastethernet 0/2 trunk

Port        Mode         Encapsulation  Status        Native vlan
Fa0/2       auto         802.1q         not-trunking  1

Port        Vlans allowed on trunk
Fa0/2       1

Port        Vlans allowed and active in management domain
Fa0/2       1

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/2       1

Switch#
```

Figure 8: Output from the show interface trunk Command

To see if Dynamic Trunking Protocol (DTP) is being used and how it is being used, use the **show dtp** [**interface** *type mod/num*] command. You can issue the command as **show dtp**, but specifying an interface will display a lot more DTP information.

VLAN Trunking Protocol (VTP)

VTP was developed by Cisco to manage VLANs belonging to switches in the enterprise network or within the same administration domain. VTP allows every switch in their domain to have an overall view of the active VLANs. VTP also allows network administrators to create, edit, delete or restrict the VLANs and propagate this information to all switches in the network. It also allows the network administrator to define which switches in the network will be allowed to create, modify and edit VLAN settings. A switch can only participate in one domain or management domain. Switches belonging to different domains do not share VTP information.

VTP allows network administrators to make configuration changes centrally in certain switches and have those changes replicated to all other switches in the domain. Careful planning of which switches are server, client or transparent is needed to properly use this protocol.

VTP uses layer 2 frames to communicate VTP information between switches in the same domain. These frames are called VTP advertisements and they let other switches in the same management domain know about active VLANs and specific VLAN parameters.

There are three VTP operating modes.

1. **Server Mode** – The default mode. Allows a switch full control of the VTP domain. A switch in server mode is allowed to create, modify or delete VLANs and then propagate these changes to the rest of the switches in the management domain. There must be at least one switch running in server mode in every VTP management domain. A switch in server mode should be physically secure.
2. **Client mode** – Switches in this mode receive and propagate VTP information, but are not allowed to create, modify or delete VLANs.
3. **Transparent mode –** Switches in this operating mode will be allowed to create, modify or delete VLANs but they will be locally significant, meaning they will not send VTP advertisements to the domain with their VLANs information. There are two versions of VTP, version 1 and version 2. The main difference between the two is that in version 2, a switch in transparent mode receives and propagates VTP advertisements from other switches (even if the management domain name doesn't match), while in version 1 they don't.

VTP advertisements are multicasts sent out the switch uplink ports. The multicast MAC address is 01-00-0C-CC-CC-CC. VTP advertisements carry the VTP management domain name, VLANs information and **VTP configuration revision number**, which is used to verify if the switch is receiving an updated advertisement. If the switch has the configuration revision number 3, and receives a VTP advertisement with configuration number 1, it doesn't accept the changes reflected in the advertisement. If the number is 4, it accepts the changes in the advertisement and updates the switch VLAN configuration (VLAN database), while updating the configuration revision number to the match the one received in the advertisement. The higher the revision number, the more trusted the information is.

When you install a new switch, you must make sure the configuration revision number is set to a number below what is currently being advertised. This prevents the ***VTP Synchronization problem***, which happens when a new switch (with incorrect VLAN database configuration) has a higher configuration revision number than the one in other switches in the VTP domain. To reset the configuration revision number to zero, you can take one of the following steps:

1. Change the VTP domain name to an arbitrary name and then change it back to the VTP domain name. At that point the configuration revision number should be reset to zero (0).
2. Change the switch to transparent mode and then change it back to server or client mode.

By default, advertisements are sent in non-secure mode. You can setup a password in order to make the transmissions secure. You need to configure the password in every switch in the management domain. Keep in mind that the VTP passwords shared between switches are sent in clear text.

VTP advertisements are sent out when VLAN changes occur in a switch in VTP server mode, or when a switch configured as a client requests the advertisement when it boots. The different advertisements are:

- **Summary advertisements –**These advertisements are sent periodically every 300 seconds and they are also triggered when a change is made to the VLAN configuration in the switch. The summary advertisement contains information necessary for other switches to make the proper changes and also security features. The information found in summary advertisements is the following; domain name, configuration revision number, time stamp, password, MD5 code, and the number of subset advertisements (defined next) that will follow.
- **Subset advertisement** – These are sent when a VLAN configuration change occurs, such as the creation of a new VLAN, port assignments, renaming of a VLAN, MTU size, VLAN type (Ethernet, token ring, etc.), VLAN number, security association identifier (SAID) and VLAN name. Information from each VLAN is sent independently in sequential subset advertisements.
- **Advertisement requests from clients –** When a client switch hears a summary advertisement with a higher revision number, or it has been reset, or the VTP domain name has been changed, it sends an advertisement request to server switches. The server then responds with a summary advertisement and subsequent subset advertisements for each existing VLAN.

Catalyst switches configured with VTP in server mode store VLAN and VTP information in the vlan.dat file in the flash memory file system. This is to prevent the switch from losing its VLAN and VTP configuration when it is restarted. Even if the running config is erased and the router is rebooted, the VLAN configuration remains. In order to clear out the VLAN configuration, the vlan.dat file must be deleted. For most fixed configuration switches, such as the 3560, issue the delete flash:vlan.dat privileged Exec command.

VTP Configuration

By default, Cisco Switches are in VTP server operational mode for the management domain NULL, which means the mode is left blank, with no password configured. If it hears a VTP advertisement, it learns the VTP domain name, VLANs and configuration revision number. The problem arises when the new switch for some reason comes with a configuration revision number that is higher than the one running in the configuration revision number of the VTP domain of the network. To avoid this, it is a recommended practice that you first start your switch out of the production network and configure the VTP domain name and also reset the configuration revision number to 0, prior to connecting the switch to the production network.

The **show vtp status** EXEC command gives a lot of useful information to verify the switch VTP parameters such as vtp domain name, vtp configuration revision number, maximum VLANs supported, number of existing VLANs, VTP operating mode (server, client or transparent), if the VTP pruning is on or off (concept defined later), MD5 digest, and the IP address of the sender of the last advertisement that produced a configuration change. The following is a sample output:

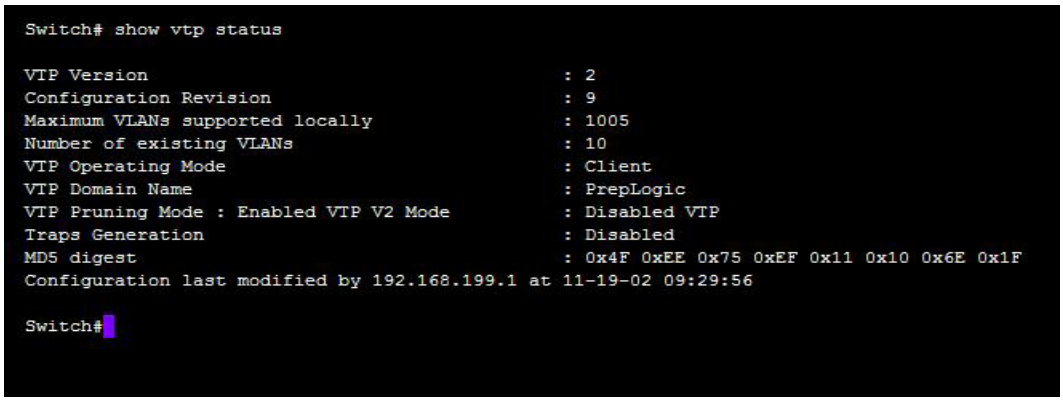

Figure 9: Output from the show vtp status Command

Here is a sample configuration:

```
Switch(config)# vtp domain preplogic
Switch(config)# vtp mode server
Switch(config)# vtp password cisco
Switch(config)# vtp version 2
```

The first line configures the VTP domain **preplogic**. The second line sets the VTP mode as server (this is the default). The third line sets the password to **cisco**. Remember that VTP uses MD5 as its hashing type for security. The last line sets the VTP to version 2.

There are a few considerations with VTP version 2:

1. VTP transparent mode propagates received advertisements from server and client switches. The VTP transparent switch doesn't check the revision number prior to forwarding the VTP advertisements.
2. VTP v2 is not interoperable with v1. A v2 capable switch can coexist with v1 but it will operate in v1 mode.
3. If you decide to run v2 in a management domain, after making sure all your switches are v2 capable, you only need to configure the version 2 in one of the server switches in the management domain, and the information of the new setting will be propagated to the remaining switches, automatically enabling v2 in all capable switches.
4. VTP v2 supports Token Ring switching and Token Ring VLANs.

If two switches are not passing VTP information between each other, verify that the VTP domain and passwords are identical. To view the VTP password, you can issue the **show vtp password** command.

Finally there is a very useful troubleshooting command that presents VTP messages and error counters. The show vtp counters EXEC command provides this information. The following is a sample output:

```
Switch# show vtp counters
VTP statistics:
Summary advertisements received         : 1
Subset advertisements received          : 2
Request advertisements received         : 1
Summary advertisements transmitted      : 1630
Subset advertisements transmitted       : 0
Request advertisements transmitted      : 4
Number of config revision errors        : 0
Number of config digest errors          : 0
Number of V1 summary errors             : 0
VTP pruning statistics:

Trunk   Join            Join        Summary advts
        Transmitted     Received    received from
                                    non-pruning-capable
                                    device
---------------------------------------------------------
Gi0/1   14590           15100       0

Switch#
```

Figure 10: Output from the show vtp counters Command

VTP Pruning

Trunk ports belong to all VLANs by default. Trunks forward all broadcasts and multicasts received in the switch. Sometimes a broadcast or multicast gets forwarded out a trunk port even when the switch on the other end doesn't have the VLAN where the broadcast was generated. VLAN pruning was created to address this issue. When VTP pruning is turned on, broadcast, multicast and unknown unicast from certain VLANs are transmitted through a trunk link only if the VLAN is present in the switch in the other end. This offers enhanced security because user data won't be travelling across the network in parts where it has no use. A possible attacker won't be able to reach portions of the network that would be accessible to him if VTP pruning is not in use.

It's important to notice that even when VTP pruning is on and has determined a certain VLAN is not needed in a trunk, an instance of Spanning Tree Protocol must be run per VLAN allowed on the trunk. To prevent an unnecessary instance of STP to run in the link you must manually eliminate that VLAN from the trunk with the **switchport trunk allowed vlan remove** *vlan-list*.

VTP pruning is disabled by default on Cisco switches. To enable it, you must type **vtp pruning** global configuration command. You should use this command in a VTP server, and it will advertise that pruning has been enabled, which enables VTP pruning in the rest of the switches in the domain, except switches in transparent mode. For switches in transparent mode you need to manually prune VLANs from unneeded trunks because VTP pruning is transparent to them.

Use **switchport trunk pruning vlan** {{{**add** | **except** | **remove**}*vlan-list*} | **none**} interface configuration command to add or remove VLANs from trunk links. By default all VLANs are eligible for pruning when you activate VLAN pruning with **vtp pruning** global configuration command.

VLANs 2-1001 are eligible for pruning by default. VLAN 1 is commonly used as the management VLAN and for this it's not eligible for pruning because it is often propagated throughout the distribution block. VLANs 1002-1005 are also not eligible for pruning since these are special, reserved VLANs for other types of networks.

VTP Troubleshooting

It is really important to make sure you don't introduce a switch configured in VTP server mode with a higher configuration revision number than the current in the domain. As we said before, it is recommended that you test the switch out of the production network and reset the configuration revision number with one of the two methods described earlier before plugging it into your production network. Introducing a server switch to the domain that is configured with a higher revision number can delete all of your production VLANs if you aren't careful.

There are several reasons why a switch might not receive VTP advertisements and information. These are the most common:

- The VTP domain name is incorrect or misspelled compared to the other switches within the domain. The client switch needs to have the same VTP domain name as the VTP Server. The domain name is case sensitive.
- The VTP version is not the same as the VTP server.
- The VTP password does not match the one configured in the domain.
- The switch is configured as a VTP client and there are no VTP servers in the domain.
- One of the links connected to the VTP server is not a trunk port.
- The switch is configured in transparent mode. Remember that in transparent mode the switch is basically inoperative in the VTP domain. If you are using VTP version 2 the switch will forward advertisements but it won't create, delete or modify VLAN information based on the advertisements received.

Spanning Tree Protocol (STP)

Spanning Tree Protocol was created to solve the problem of bridging loops, defined in the "Switching Operation" section earlier. STP is defined in the IEEE 802.1D standard.

Without STP, switches are transparent to each other (they don't modify Ethernet frames), and they can't do anything about redundant paths. That means that they will send frames out all uplink ports to the same destination. STP makes switches aware of each other and allows them to negotiate and block some ports from forwarding frames which creates a loop-free path to locations with redundant links or paths.

STP communicates with all switches in the network and selects a switch as a reference point (root bridge), and detects all redundant paths to it. STP then selects the best paths and blocks the less than optimal one or ones. STP maintains communications between all switches in the network segment (subnet) in order to compute the best path among blocked links, when the active path link fails.

STP uses Bridge Protocol Data Units (BPDU) to communicate switches. A switch sends a BPDU frame out a port and uses its unique MAC address as the source address and the well-known STP multicast address 01-80-c2-00-00-00 as destination.

There are two types of BPDUs:

1. **Configuration BPDU** – Used to compute the spanning-tree.
2. **Topology change Notification (TCN) (BPDU)** – Announce topology changes in the network.

By default, BPDUs are sent out every two seconds to ensure topology changes and network conditions are updated promptly and loops are identified, prevented or corrected.

BPDUs carry all information necessary for the election of the common reference points of the network segment, along with information such as switch identification, link information helpful to determine the optimal path (path cost) when more than one is available.

The central point we have mentioned a couple times; it's a switch called the *root bridge*. The term bridge remains from the times when bridges where doing the functions now performed by switches. When you hear root bridge think of *root switch*.

The main characteristic of a root bridge is that all its ports are forwarding frames, which is the same as saying that none of its ports are in the blocking state. STP port states (Blocking, Listening, Learning and Forwarding) are defined later. The root bridge is elected for a network segment by analyzing the ***bridge ID,*** which is an 8 byte value composed by the bridge priority and the MAC address of the port. **The lowest Bridge ID is elected as the root bridge.**

The bridge priority is a 2 byte value that is used to establish the priority of a switch in relation to the others switches in the domain. The default on Cisco Catalyst switches is 37,768 or 0x8000.

The **Bridge ID** takes the form of Bridge Priority: MAC address. As you can see, since the bridge priority is the same in all switches by default, the MAC address is the value used to determine the root bridge. The lowest MAC address wins the election. As you should remember from the CCNA studies, the MAC address is a unique 48 bit value used to identify network adapters and devices. In Cisco Switches, the MAC address used comes from the supervisor module or the backplane, from a pool of 1024 addresses assigned to one of these parts.

The default selection method is far from optimal because we cannot control MAC addresses as they are set by the manufacturer. This is why the network administrator must configure all switches and decide which one is elected as the root bridge.

The bridge ID can also be configured as an **extended system ID**, defined in the **IEEE 802.1t** standard. It is a 4bit priority multiplier, plus a 12bit VLAN ID followed by a non-unique MAC address for the VLAN. If the switch doesn't support 1024 unique MAC addresses for its own use, the extended system ID is used by default. To use the extended system ID use the following command:

```
Switch(config)# spanning-tree extend system-id
```

When every switch boots up, it automatically assumes it is the root bridge. Then it starts sending BPDUs with its bridge ID as the root bridge ID. It also receives BPDUs from other switches in the same process, and updates the root bridge ID in its BPDUs when it receives one with the lowest bridge ID. All switches in the segment do the same and as BPDUs with lowest root bridge ID are received, the BPDUs are sent with an updated root bridge ID. Until they all agree with a root bridge ID and root bridge. At that point, STP is considered *converged*.

The election process is continuous, with BPDUs sent every 2 seconds. If a new switch with a lower Bridge ID enters the network, BPDUs with a lower root bridge ID are propagated and the computation begins, ending with the election of the switch as the root bridge.

Root ports are ports in non-root switches that provide the lowest cost to the root bridge/switch. A switch selects its root port using the concept of the ***root path cost***, which is the cumulative cost of all links leading to the root bridge. The cost of a certain link is called path cost. These ports are also always forwarding.

You must be careful to learn the difference between path cost and root path cost. Root Path Cost is a cumulative value, that is added to the BPDUs and they move through switches in the network segment, while path cost is simply the cost associated with a port in a switch.

The term cost was originally defined in the IEEE 802.1D standard as 1000Mbps divided by the link bandwidth in Mbps. Since the popularity of Gigabit Ethernet the IEEE now defines cost with a nonlinear equation. Note that the old scale is rarely used, and can easily be calculated. The new costs values are the ones shown in the following table:

Link Bandwidth	New STP Cost
4 Mbps	250
10 Mbps	100
16 Mbps	62
45 Mbps	39
100 Mbps	19
155 Mbps	14
622 Mbps	6
1 Gbps	4
10 Gbps	2

Figure 11: STP Costs

The official SWITCH exam guide establishes that old cost values are now legacy and that the new values are the ones that must be used in both production networks and on the exam.

The Root Path Cost is determined in the following manner:

1. The root bridge originates a BPDU with a root path cost value of 0. This is because its port doesn't have to travel through a link in order to reach the root, which is the switch itself.
2. The next switch that receives the BPDU adds the cost of the link based on the configured bandwidth interface parameter.
3. This second switch sends out BPDUs with this new cumulative root path cost value.
4. The process is repeated every time a neighboring switch receives a BPDU: The cost of the traversing link is added to the cost advertised by the neighboring switch.

It is very important to note that the root path cost is incremented as the BPDUs are received, not as they are sent.

When the switch receives a BPDU with a root path cost and adds its link cost, it saves the value in volatile memory so it can later compare other BPDU's root path cost to the one it has saved. When a BPDU with lower root path cost is received, it knows that path is better to the root, immediately becoming the new root path cost. The process constantly looks for better paths to the root and that is how the **root port** is elected.

Designated Ports: These are forwarding ports connected to a network segment. If more than one port is connected to a network segment or two different switches share the same network segment, only one port will be elected to forward frames in order to avoid loops. This port is the *designated port*. The other port will be placed into a blocking state as described below.

Designated ports election. The process is as follows:

1. Lowest root bridge ID.
2. The lowest root path cost.
3. Lowest sender bridge ID.
4. Lowest sender port ID.

STP States

When a switch is powered on, before it is able to forward frames transparently as it should, its ports go through several stages or states:

- **Disabled** – These are ports that are shut down by the administrator or that have been disabled by some kind of error condition (errdisabled state). This is a special state and is not a part of the regular STP progression.
- **Blocking –** In this state a port is unable to send or receive data. It only receives BPDUs to start learning about the spanning tree topology. Ports that are put in the standby mode also enter the Blocking state.
- **Listening** – A port transitions from blocking to listening when the switch thinks the port can become a designated or root port. At this state the port starts accepting frames but it doesn't populate the MAC Address table or forwards frames received. The port is allowed to actively participate in the spanning tree process, by sending and receiving BPDUs. The port is also allowed to become a designated or root port because it sends BPDUs of its own.
- **Learning** – After a period of time called Forward Delay, the switch transitions from the listening state to the learning state. The port still sends and receives BPDUs, but now also populates the MAC address table with the source address of frames received. The function of this state is to give the switch a little time of participation in the spanning tree process and computations without giving the ability to forward frames.
- **Forwarding** – After another Forward Delay period, the port is allowed to move to the forwarding state. Now the port sends and receives BPDUs, populates the MAC address table, forwards frames. It is a fully functional port in the spanning tree loop free topology.

You can watch the STP process as it transitions between the different STP states by issuing the show spanning-tree interface type mod/port EXEC command multiple times.

STP Timers

STP uses timers to guarantee that a network converges before a loop can occur. These timers are:

- **Hello Time** – This is the time the root bridge takes to send a configuration BPDU. The default as we said earlier is 2 seconds. You only need to configure this setting in the root bridge because non-root bridges only relay root bridge configuration BPDUs.
- **Forward Delay –** The period of time, a default of 15 seconds, during which a switch spends in listening and learning states.
- **Max Age –** This is the time that a switch uses a BPDU before considering it dated. While STP is being executed a switch keeps a BPDU from the best source or "best BPDU". The max age indicates to the switch the maximum time the switch will keep the BPDU after it last received it. The default Max Age value is 20 seconds.

These timers should only be changed after serious consideration. These changes should be made on the root bridge switch and it will propagate the changes throughout the switches participating in the spanning tree process. If you change STP timers trying to improve convergence time switching loops can occur.

Topology Changes in STP

Switches send a topology change notice (TCN) BPDU when they detect a topology change. There are two types of topology changes. One refers to the changes that occur when a port goes from the forwarding or learning state to blocking state or when it moves a port to the forwarding state. When any of this happens, the switch sends a TCN BPDU out its root port so that the root bridge switch receives it. Every time an uplink switch receives the TCN BPDU it forwards towards the root switch and sends back to the sender an acknowledgement. Finally, the root bridge receives the notification that a topology change has occurred and sends the acknowledgement to the originating switch. It then marks the topology change flag in the configuration BPDU, which is then propagated to all the switches in the network or switched domain.

When the configuration BPDU is received by every non-root switch in the network, they shorten their bridge table aging times from the default to the Forward Delay time. This is done to force a faster purging of the MAC table in order to prevent problems associated with the change in the topology. Here are the three different types of topology changes:

- **Direct Topology Change** – The change produced because of link failure. What happens here is a recalculation of the spanning tree, but the changes will be mostly in the switches that have a problem with the link. The whole STP process doesn't go through a massive recalculation.
- **Indirect Topology Change** – This is the topology change that is detected in a link that appears up in both ends. The kind of disruption is usually created by a firewall, another switch, a service provider's switch, etc. This kind of failure is associated with the use of the switch's aging timers. When the BPDUs are not received in the port suffering from the frame filtering that will end up producing the "indirect topology change", the non-root bridge will flush its "best" BPDUs after the Max Age timer is up. This makes it accept a better BPDU in a port that is in the blocking state. This allows the port to transition from Blocking, Listening, Learning and finally Forwarding state.
- **Insignificant Topology Change** – This is a topology change that doesn't cause the spanning tree to make any kind of recalculation. This happens when hosts are turned on and off. What happens is the link changes and sends the TCN BPDU to the root bridge. The root bridge responds with the configuration BPDU with the topology change flag, that causes a faster than usual flush of the CAM table contents. When you think of this, there is really no problem with STP as no computation is required. The problem here is that in a large network, this constant flushing of the MAC address table significantly impacts network performance, because you will have a lot of unknown unicast flooding. Portfast is a spanning tree protocol feature that allows a switch connecting to hosts to be excluded for the active participation in STP. Ports configured with Portfast don't send BPDUs. Portfast will be explained and configured later in this section.

STP Types

Spanning Tree Protocol (STP) was developed to prevent switching loops by providing a loop-free logical network topology and to provide rapid recovery from topology changes. Provisions have been taken to adapt STP to the changing and evolving structures and features of the switched networks. STP was originally developed to work in bridged networks, with basically one instance of STP for the whole segment. Today networks offer features that require more powerful solutions and STP has evolved offering several possibilities. The different types of STP encountered in today's networks are:

- **Common Spanning Tree (CST)** – This is the IEEE 802.1D standard. The biggest limitation of the original STP is that there is a single instance of it running for all VLANs. This keeps CPU loads low during STP calculations but has several limitations, like the inability to load balance based on traffic for different VLANs. CST BPDUs are transmitted with untagged frames over the native VLAN.
- **Per VLAN Spanning Tree (PVST)** – Cisco's proprietary implementation of STP. It provided the powerful feature of one instance of STP for each VLAN. Load Balancing is possible. PVST requires ISL Cisco proprietary VLAN identification frame encapsulation. Since ISL is no longer supported in current Cisco IOS switches, you won't find PVST in a lot of networks.
- **Per VLAN Spanning Tree Plus (PVST+)** – A more advanced Cisco Proprietary version of STP. It offers the advantages of PVST and also interoperates with CST. It offers the flexibility of working with both ISL and 802.1Q encapsulation methods.

STP Configuration

Switches run an instance of STP by every active VLAN on them by default. In some cases you can find STP has been shut down for a specific VLAN or interface. Use the following commands to enable STP:

```
Switch(config)# spanning-tree vlan 5
Switch(config)#interface fastethernet 0/1
Switch(config-if)#spanning-tree vlan 10
```

In the first line we use spanning-tree vlan 5 global configuration command to enable spanning tree protocol in the switch for VLAN 5. This effectively starts an instance of STP for VLAN 5 in case it had been previously shut down.

The third line enables STP at the interface level for the VLAN 10. We used spanning-tree vlan 10 interface configuration command.

Root bridge selection should not be left to the methods of the protocol as was discussed earlier. It should be controlled by the network administrator. As we explained earlier, the criterion is far from optimal. Remember the root bridge switch is the one with the lowest bridge ID in the network segment. The Bridge ID is the combination of the bridge priority and the MAC address. Default bridge priority is 32,768, which makes the deciding factor the lowest MAC address. With this criterion the slowest switch could end up becoming the root switch, which is far from optimal. We can configure the switch and manipulate the election so we can have the root bridge of our choice. There is also a secondary root bridge, which is nothing more than the switch that would take the place of the root bridge if the root fails.

Before we enter configuration commands, it is important to know the placement of the root bridge switch should be as close to the center or core layer of the switched network. In case we have a server farm, the maximum load will be in the exit/entry point to the farm and that is the point where the root should be.

Root Bridge Switch Placement and Configuration

There are two methods to force the election of a root bridge:

1. Assign a bridge ID lower than the default bridge ID if the rest of the switches in the network are left to their default or simply make sure the switch you want to become the switch has a lower bridge ID than the rest of the switches in the network. The command to achieve this is the following:

   ```
   Switch(config)# spanning-tree vlan vlan-list priority bridge-
   priority-value
   ```

   Vlan-list uses the same ranges used with the range global configuration command and *bridge-priority-value* is simply the priority and the value ranges from 0 to 61,440 in multiples of 4096.

2. Set the switch to automatically select a priority that makes it the root switch:

   ```
   Switch(config)# spanning-tree vlan vlan-id root {primary |
   secondary} [diameter diameter]
   ```

   This command is a macro that executes several other commands. The switch modifies the values of the bridge ID depending on the values received from other switches. It will use the lowest ID if you set it to be the primary root and the second lowest if set to secondary. Notice the switch will use one 4096 multiple below the lowest bridge ID advertised in the network. The **diameter** specifies the length (the amount or switches) of the switched network from the core to the access layer switch connecting endpoints. The default diameter is 7 and this is the value used to calculate the BPDU timers.

To display the STP Bridge Priority values use the **show spanning-tree vlan** vlan-id EXEC command. The following is a sample output:

```
Switch# show spanning-tree vlan 5
VLAN0005
    Spanning tree enabled protocol ieee
Root ID Priority 4200
        Address     000c.1004.9fef
        Cost    4
        Port    1 (GigabitEthernet0/1)
        Hello Time   2 sec      Max Age 20 sec      Forward Delay 15 sec

Bridge ID Priority 32868 (priority 32768 sys-id-ext 5)
          Address 000c.1000.9f1c
          Hello Time  2 sec     Max Age 20 sec      Forward Delay 15 sec
          Aging Time    300

Switch#
```

Figure 12: Output from the show spanning-tree vlan Command

When the switch is the root switch for the STP process, you will see the "**This is the Root Bridge**" message under MAC address on some Cisco switches. Otherwise, you can compare the Root ID MAC address and the Bridge ID MAC address. If these two match, than you are looking at the root bridge.

Notice that when you use the **spanning-tree vlan** *vlan-id* **root** global configuration command the macro that runs doesn't guarantee the switch will become the root. It is possible that another switch has already been configured with a lower bridge priority. In this case you need to manually configure the switch to bridge priority of zero (0) or a value lower than the current root. It is recommended that you do this and set the priority to a very low value in order to prevent other switches from being elected as root. **Spanning-tree vlan** *vlan-id* **priority** *priority-value* is the global configuration command you should use to achieve this.

Configuring Cost and Port-Priority to Manipulate Path Selection

You can manipulate how the frames are forwarded in a per-VLAN and/or per-interface basis with spanning-tree commands. This allows the network administrator or designer to load balance traffic between available links.

You can modify the **path cost** associated with a link if you need to make it the preferred route to a destination with the **spanning-tree** [**vlan** *vlan-id*] **cost** *cost* interface configuration command. Notice the optional condition of the **vlan** command. IF you don't specify the VLAN the cost assigned will be used in all instances of STP, which is the same as saying it will be used in all VLANs. The cost value can range from 1 to 65,535.

```
Switch(config-if)# spanning-tree vlan 10 cost 100
```

In that line we configure the interface to set its path cost to 100 for VLAN 10 transit.

You can see the cost of an interface with the **show spanning-tree interface** type mod/num [**cost**] EXEC command.

The **Port ID** is a 16 bit value composed of the port priority and the port priority plus the port number. Remember it is used as a tie breaker in the criteria STP uses to choose a path. The other important command we mentioned is the **spanning-tree** [**vlan** *vlan-list*] **port-priority** *port-priority*.

Configuring STP Timers

You can also manipulate the amount of time the switch spends in different STP states and the interval between the Hello BPDUs. These are the STP timers and they can be customized to the different requirements of networks. The default timers are determined considering a network diameter of 7. If this value is different, you can manipulate both the diameter and the timers to improve convergence. If the network diameter is bigger, the timers should be increased, while smaller networks can have improved STP convergence time by reducing the STP timers.

Remember the network **diameter** must be **determined and considered prior to modifying these timers**. The network diameter is the quantity of switches from the network core to the outmost access layer switch. Here are the commands:

```
Switch(config)# spanning-tree [vlan vlan-id] hello-time seconds
Switch(config)# spanning-tree [vlan vlan-id] forward-time seconds
Switch(config)# spanning-tree [vlan vlan-id] max-age seconds
```

If you don't use the vlan portion the commands are applied to all instances of STP running on the switch.

You can also change all STP timers on a single switch with the following command:

```
Switch(config)# spanning-tree vlan vlan-list root {primary | secondary}
[diameter] diameter  [hello-time hello-time]]
```

After you enter this command and use the optional **diameter** and **hello-time** configuration, the Max Age and Forward Delay timers will be reconfigured automatically to an appropriate value. Notice the **vlan** is mandatory in order to have this automatic STP timers configuration feature take place.

Redundant Link Convergence

Spanning Tree Protocol offers several features that allow faster convergence in certain special situations.

PortFast allows a switch to immediately transition a port from the blocking state to forwarding state. This feature can only be used in access ports in the access layer switches. The switch keeps running STP in the port and it is put in the blocking state if a loop is detected. Portfast is disabled by default, and you can change to make it the default in all non-trunking ports with the **spanning-tree portfast default** global configuration command. To enable Portfast at the interface level, use the **spanning-tree portfast** interface configuration command. Another big benefit comes from the optimization provided by the command because it doesn't send TCN BPDUs when the endpoints come up or down, saving valuable bandwidth and switch resources (CPU utilization). You can also use the **switchport host** interface configuration command to start a macro that will enable port fast at the switchport level.

Uplinkfast allows a switch with more than one uplink to the root switch to transition one of the blocked ports to forwarding state in case the root port fails. The uplink that will transition to the forwarding state will be the one with the lowest root path cost. There are three things to keep in mind when using Uplinkfast:

1. The transition from blocking to forwarding doesn't happen instantly. It actually takes 1-3 seconds, but when compared to the standard process without the uplinkfast configured, which takes approximately 50seconds, it sure feels like it's instant.
2. Uplink is enabled globally and can't be configured in a per-VLAN or per-interface basis.
3. Uplinkfast cannot be configured in a root switch.

When the original root port comes back up it will take its place as root port and the uplink port activated by the Uplinkfast feature will go back to the blocking state. This doesn't happen automatically. The switches use the following formula to determine how long they should wait when they detect the root port has come up:

(2 x Forward Delay) + 5 seconds.

When Uplinkfast enters and a blocked port assumes the function of the root port because of a root port failure, the switch takes actions to prevent becoming the root switch:

1. The switch priority is set to 49,152 to ensure that all other switches with their default priorities need to go down before this can become the root bridge.
2. STP port cost is increased by 3000. This makes it very unlikely that this switch will be actively forwarding frames to the root bridge.

To enable Uplinkfast use the following command:

```
Switch(config)# spanning-tree uplinkfast [max-update-rate update-value]
```

The **max-update-rate** establishes how many multicast frames are sent to the 01-00-0c-cd-cd-cd MAC address in packets per seconds (pps). These frames are sent with the source address of all neighbors that happen to be in the CAM table at that time. The intention is to make these frames flow and go through the new uplink port, letting the other switches know about the newly activated path to those source addresses.

You should use UplinkFast in access layer switches connecting to distribution layer switches. It should never use them in core layer switches. The risk of having a switching loop across the network core and the impact of its occurrence makes it a bad risk management proposition to use uplinkfast in the network core. It is better to let the root switch or a designated switch in the network core recalculate the STP process and converge, even if this means a few seconds of downtime. A switching loop crossing the network core will mean a lot more trouble and downtime than the 50 seconds that STP will take to converge.

If you want to see the status of the STP UplinkFast use the EXEC command **show spanning-tree uplinkfast.**

BackboneFast is the feature used in the network core switches to speed up STP convergence. When a switch loses its indirect connection to the root switch, it starts sending BPDUs letting the domain know it is the root switch. When another switch receives this inferior BPDU, it must wait until the Max Age timer in the port leading to the inferior BPDU expires, before it starts sending the superior BPDUs it has received or generates if it is the root switch. When BackboneFast is enabled, the switch doesn't have to wait for the Max Ager timer to expire before it sends the superior BPDU.

BackboneFast uses Root Link Query (RLQ) protocol. RLQ uses a series of requests and responses to detect link failures. There are two types of RLQ messages: RLQ Query and RLQ Response.

The purpose of the RLQ request is to check connectivity to a root bridge. The RLQ Request is, for this reason, almost always sent out ports receiving BPDUs. The root bridge is specified in the RLQ Request. When the RLQ response is received, it specifies the root bridge that originated the response. If the two roots (the one in the RLQ request and the one in the RLQ Response) are the same, connectivity is still alive, else, it is lost.

The RLQ query is only replied by the root. If a non-root switch receives an RLQ query it forwards the frame towards the root switch, flooding the frame through its designated ports. To prevent the RLQ response frames from being propagated to segments where it is not necessary, the switch originating the RLQ query adds its bridge ID to the frame. That way, when it receives a response it knows the information is only useful to itself and the frame is not forwarded to the other designated ports. RLQ must be running in all switches in the network segment participating in the STP process. That is why BackboneFast must be configured in all switches in the network segment.

To enable BackboneFast use **spanning-tree backbonefast** global configuration command in all switches. Make sure you run the command in all switches or BackboneFast will not work properly.

You can check if BackboneFast has been enabled with the show spanning-tree backbonefast EXEC command. The switch will display a message indicating whether or not the feature has been enabled.

```
Switch# show spanning-tree backbonefast
BackboneFast is enabled
Switch#
```

Protecting the STP Process

In this section we'll learn the features that help us maintain a switched network and STP processes running without disruptions caused by security breaches like rogue switches.

Root Guard is used at the port level to prevent a switch downstream of the port from becoming the root switch for the network segment. When a switch receives a superior BPDU in a port that has been configured with Root Guard, the switch discards the BPDU frame and puts the port in the ***root-inconsistent*** STP state. Root-inconsistent is operationally equal to the listening state, and no frames are forwarded.

Since Root Guard is configured at the port level, you need to use it in switches uplink ports that will never be allowed to become a root, because you cannot disable the root guard feature in a per-VLAN basis. Configure root guard at the interface level, as follows:

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

You can also check the ports in the root-inconsistent state using the show spanning-tree inconsistent ports privileged EXEC command.

BPDU Guard is used to prevent rogue switches connected to production switches. Typically this command is used on access ports because this is likely where a rogue switch will be placed on the network. BPDU Guard is used in ports configured with Portfast. If a BPDU is received in the port configured with BPDU Guard, the port is automatically shut down and put in the err-disabled state. You can configure BPDU Guard at both the interface level and in global configuration mode. The following configures BPDU Guard in all ports in the switch (remember that it will only be activated in portfast enabled ports):

```
Switch(config)# spanning-tree portfast bpduguard default
```

To configure at the interface level:

```
Switch(config-if)# spanning-tree bpduguard [enable | disable]
```

It is very important to know and understand the difference between configuring BPDUGUARD at the interface level or at the global level:

- BPDUGUARD at the global level: configures all PortFast enabled ports to shut down if a BPDU is received.
- BPDU configured at the interface level: the port is shut down if a BPDU is received, but the port doesn't have to be portfast enabled.

In both cases the shut-down port needs to be manually re-enabled, from the errdisabled state.
PortFast BPDU Filtering is a very useful feature because it allows you to filter BPDUs in PortFast-enabled ports without effectively shutting down the port or putting it in the errdisabled state. You need to be very careful because this feature works very differently when configured locally and when configured globally.

When configured globally it will disable PortFast ***from PortFast-enabled ports*** if a BPDU is received in the port or ports. You have to be very careful with this command because it can create switching loops if a trunking switch is connected. As a general rule, bpdufilter should not be globally enabled in physically insecure switches.

When configured at the interface level it will simply quietly drop the BPDU frames received. The port will not send or receive BPDUs.

To enable PortFast BPDU filtering at the global level use the following command:

```
Switch(config)# spanning-tree portfast bpdufilter default
```

To enable BPDU filtering at the interface level use the following command:

```
Switch(config-if)# spanning-tree bpdufilter enable
```

There is a very useful show command that displays a lot of valuable information regarding Spanning Tree Protocol and its features, including if the BPDU Filter has been enabled. That is the **show spanning-tree summary totals** EXEC command. We'll come back to this command later and will see a sample output.

Instabilities Due of Loss of BPDUs

BPDUs are used by STP to know about the switched network topology. STP relies on BPDUs to maintain a loop-free topology and network integrity. There are conditions where one or more BPDUs might not be received or relayed at certain switches, causing the switch to recalculate or try to converge to a different network topology even when a change hasn't really occurred. This is commonly a physical layer problem with copper and even more so with fiber optic interfaces. To prevent this from happening we have Loop Guard and Unidirectional Link Detection (UDLD).

Loop Guard monitors the BPDU activity in non-designated ports (blocking state). While BPDUs are being received, everything stays the same. When BPDUs are not received and the Max Age timer is up, Loop Guard prevents the port from going through the STP port states and puts it in the loop-inconsistent state, which is operationally the same as Blocking. When BPDUs are received again, the port goes back to normal STP states without any manual intervention.

You can enable Loop Guard at the interface level or globally. To enable it at the interface level, use the following command:

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

To enable Loop Guard in all non-designated ports globally, use the following global configuration command:

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

Loop Guard is configured at the interface level but it works in a per-VLAN basis. This means that if the port has Loop Guard enabled, it will only use it for STP instances or VLANs for which the port is in the Blocking state.

As a network designer or administrator you must decide if you require loopguard enabled in all STP instances running in the switch (in this case you will enable it globally) or if you want to select specific interfaces and their associated VLANs (in this case you will enable it at the interface level).

Unidirectional Link Detection (UDLD) Is a Cisco proprietary STP feature that detects links with unilateral connectivity problems. This problem is common in fiber optic Ethernet links where two circuits work independently in transmission or receiving. If one of those circuits is damaged and communication is only working unidirectionally, the STP topology can be damaged because the port not receiving BPDUs will try to transition to the forwarding state and a loop can form.

The worst problem that can arise with this situation is that if the bidirectional communication is damaged, the rest of the topology might never know what is happening at the other end of the link. To prevent this situation UDLD constantly monitors the bidirectional situation of a link. It sends a special UDLD frame and expects to receive a response from the other end. If the reply is received, the link is operating as it should, but if a response is not received, the link is unidirectional and action must be taken.

UDLD must be enabled at both ends and its Message interval time should be set to a value less than the Max age timer. This is the whole point of the feature: to prevent a loop from forming because of action taken in a port before it purges its BPDUs because of a link failure.

You can configure UDLD globally and also at the interface level. If you enable UDLD globally all gigabit Ethernet fiber ports will enable UDLD. Since the nature of the twisted pair and copper wire don't suffer from the layer 1 conditions that are suited for the unidirectional link problems, the UDLD is not enabled on them by default. You CAN configure UDLD for such ports but you need to enable it manually. Here are examples of globally enabling UDLD and interface level UDLD configuration:

```
Switch(config)# udld {enable | aggressive | message time seconds}
Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# udld {enable | aggressive | disable}
```

In the first line we enable UDLD globally. Remember that at this point all the fiber optic ports will enable UDLD. Copper Ethernet ports require that you enable it manually at the interface level, as was done in the third line.

UDLD has two operating modes: Standard and aggressive.

- **Normal mode**: The port continues its operation normally after detecting a link failure. The port is only marked as having a problem and a syslog message is created.
- **Aggressive mode**: When a bidirectional link is broken, UDLD detects the issue and attempts to restore bidirectional activity. Every 8 seconds, UDLD sends messages that, if not received and replied to put the port in errdisabled state. The port cannot be used until it is administratively shut down and then turned back on.

The **message time** is the timer interval between UDLD frames. Remember the message timer must be lower than the Max Age timer.

Earlier we mentioned a show command with a very interesting output. It's the show spanning-tree summary totals EXEC command. Here is a sample output:

```
Switch#show spanning-tree summary totals
Switch is in rapid-pvst mode
Root bridge for: VLAN0001, VLAN0010, VLAN0020, VLAN0030, VLAN0050
Extended system ID                  is enabled
Portfast Default                    is disabled
PortFast BPDU Guard Default         is disabled
Portfast BPDU Filter Default        is disabled
Loopguard Default                   is disabled
EtherChannel misconfig guard        is enabled
UplinkFast                          is disabled
BackboneFast                        is disabled
Configured Pathcost method used     is short

Name            Blocking    Listening   Learning      Forwarding       STP Active
--------------  ----------- ----------- ------------  ---------------  ---------------
5 vlans         0           0           0             24               24
Switch#
```

Figure 13: Output from the show spanning-tree summary Command

Advanced Spanning Tree Protocol

The IEEE 802.1D standard, CST (Common Spanning Tree Protocol), is now considered a legacy protocol because of its convergence periods, which are now considered too long for most network applications. The new protocols are based on the 802.1D standard technology but focused on faster convergence to suit the increasing demands of modern networks. These new standards are the Rapid STP (RSTP), and Multiple STP (MST or MSTP). These will be defined in this section.

Rapid Spanning Tree Protocol (RSTP) IEEE 802.1w

RSTP (802.1w) was developed with STP 802.1D concepts to attend the needs of smaller convergence periods. That is why it is considered an extension of the 802.1D standard.

RSTP elects a root switch with the same criteria used in STP 802.1D, using the Bridge ID concept. In addition, many of the legacy STP features we discussed that were manual options with STP are built-in to RSTP by default as you will see by reading below.

Most of the concepts used in STP are used in RSTP, but the port roles are different between both protocols. The port roles in RSTP are:

- Root port.
- Designated port.
- Blocking port.

The root port has the same function that it had in STP, which was frame forwarding to the root switch using the lowest root path cost available. The root switch doesn't have any root port.

The designated port is the port in a network segment that has the best root path cost. This might be confusing. The key to understand the difference between root port and designated ports is the concept network segment. The root port is the port **in the switch** with the best root path cost. The designated port is the port with the best root path cost in a network segment. Think of two switches both connected to the same network segment. RSTP will block one of them. The concept of the designated port is related to the segment, not the switch. It's the best route from a network segment to the root port. When several switches have ports in the same network segment, only one will be the designated port if the network segment is only receiving traffic from one VLAN (think of the network segment as a subnet).

The blocking port or ports: This can be further divided into two new roles: The alternate port and the backup port.

1. **Alternate port**: This is an alternate route to the root port. This is the port with the second best root path cost. Remember that the root path cost refers specifically to the best route the switch itself has to the root. This port is effectively in the blocking state and will forward frames only when the root port fails. It is the second best alternative to the root bridge.
2. **Backup port**: This role relates to the designated port. This is a backup route that frames will take in case the designated port fails. Think of it as the backup route from a network segment. This port is effectively in the blocking state and will forward frames only when the designated port fails.

RSTP port roles are root port, designated port, alternate port and backup port (the alternate and backup are blocking ports and root and designated are in the forwarding state).

The port states are also different in RSTP. RSTP classifies the port states depending on what the switch does with the frames received. The following are the port states in RSTP:

- **Discarding** – This state resembles the disabled, blocking and listening port states of CST.
- **Learning** – In this state, frames are dropped but MAC addresses are learned using the source MAC address of the frames.
- **Forwarding** – In this state the port is operationally active, which means frames are being forwarded and MAC addresses are being learned.

BPDUs in RSTP

The main difference with STP and RSTP, when it comes to BPDUs, is that in STP, the BPDUs are sent by the root bridge and relayed by other switches. In RSTP, BPDUs are part of an interactive process that takes place between switches in order to negotiate a port role. BPDUs are sent at Hello Time intervals regardless of if superior BPDUs from the root switch have been received or not. This new behavior gives all the switches in the RSTP process the power to influence and maintain the switched network topology.

Since BPDUs are expected every two seconds (the default Hello Timer for both STP and RSTP) action can be taken faster in case of a link failure. By default, the link is consider down when the switch doesn't hear from the other end in three Hello intervals (6 seconds) and the information regarding that assumed unreachable end is aged out. This is a great enhancement with respect to STP which takes the Max Age timer (20 seconds by default) to be able to react to network changes.

RSTP can coexist with BPDUs generated by switches operating STP 802.1D. A port operates in the mode of the first received BPDU and by default locks that mode for a configurable time interval. This is to prevent a constant change from one mode to another when a migration is in place or simply when there are two modes in operation in the STP domain.

RSTP Convergence

Convergence is the state in which all switches in the switched network have agreed on certain functions and paths to provide a loop free topology. As we know, RSTP was developed to converge faster than its predecessor, STP 802.1D.

A STP process has converged when a root switch has been elected, all switches in the STP domain know about it and all ports in all participating switches are either in the forwarding or blocking state.

RSTP speeds up the convergence process by eliminating the known STP timers that make the switches wait for a certain period before transitioning to another port state. RSTP eliminates this process and instead makes decisions about port states based on **port types. The convergence time is usually 30 seconds faster than in CST, because Forward Delay timer is not used.**

Port Types

- **Root port** – The port with the best path to the Root switch of the network and there can only be one per switch.
- **Edge port –** A port located in the edge of the network, connecting end users. This port operates exactly as a portfast enabled port in STP. This port assumes that a loop cannot be formed because there is only one host directly connected. If a BPDU is received in an edge port, its edge port condition is lost. **The port then becomes a point-to-point port.**
- **Point-to-point port –** A port that connects two switches operating in full duplex and is a designated port.

RSTP achieves faster convergence by propagating handshakes over point-to-point links. The handshakes take place with the nearest neighbor and once successful, the process moves outward towards the edge switches. While this process unfolds every participating switch must take actions to prevent loops from forming. This is done with a synchronization process.

Synchronization is the process that takes place in order for RSTP to converge. The non-edge ports start in the discarding state, and after receiving BPDUs decide the root port. The process unfolds from the core to the edge. When the superior BPDU is received, the switch sends an agreement BPDU letting the other know it has agreed to have the connected port become the root port. While this happens, other non-edge ports are in the blocking state, and when the decision about forwarding or blocking the port towards the root bridge is finished, the other non-edge ports leading towards the access switches or network edges start the selection process, leading to the election of designated ports and blocked ports. This happens until the process reaches the edge ports, where this process is not required, since these ports are in the forwarding state as soon as they detect the link is up.

Topology Change Detection in RSTP

In RSTP, a topology change is only generated when a non-edge port moves to the forwarding state. All other port changes are not considered a topology change anymore. When an RSTP bridge detects a topology change, the following happens:

- It kicks off the topology change "while timer" with a value equal to twice the hello-time for all its non-edge designated ports and its root port, if necessary. The switch starts the topology change while timer, with a default time of twice the hello-time. The timer runs for all the switch's non-edge designated ports and its root port.
- It clears all CAM table entries (MAC addresses) associated with those ports.
- BPDUs sent out while the topology change while timer is running have the TC bit set. BPDUs are also sent out the root port.

Topology Convergence Propagation in RSTP

When a bridge receives a BPDU with the TC bit set from a neighbor, the following occurs:

- The switch clears all learned MAC addresses except the one that receives the topology change.
- It starts the TC while timer and sends BPDUs with TC set for as long as the timer is running, on all its designated ports and root port. RSTP no longer uses the specific TCN BPDU, unless a switch running a legacy STP needs to be notified.

This new topology change procedure makes the propagation convergence much faster compared to 802.1D. In STP the switch that suffered the link outage had no direct involvement in the propagation process, as only the root bridge did this, while now in RSTP the switch that detects the problem is in charge of the propagation of the information. Since there is no need to wait for the root bridge to be aware of the changes, the convergence to a new topology is faster.

RSTP Configuration

Cisco switches operate in Per-VLAN Spanning Tree Plus (PVST+) by default, which is why before RSTP can be used it has to be enabled globally with MST or RPVST+.

In order to enable RPVST+ you need to use the following global configuration command:

```
Switch(config)# spanning-tree mode rapid-pvst
```

To revert back to the default PVST+, with the now legacy 802.1D STP, you only need one global configuration command:

```
Switch(config)# spanning-tree mode pvst
```

You can see in which STP mode the switch is operating with the **show spanning-tree vlan** *vlan-id* EXEC configuration command. The following is an actual switch output:

```
Switch# show spanning-tree vlan 10
VLAN0010
Spanning tree enabled protocol rstp
Root ID    Priority    32778
           Address     000d.29ad.ae00
           This bridge is the root
           Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
Bridge ID  Priority    32778  (priority 32768 sys-id-ext 10)
           Address     000d.29ad.ae00
           Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time 300

Interface          Role Sts Cost      Prio.Nbr Type
------------------ ---- --- --------- -------- --------------------------------
Fa0/4              Desg FWD 19        128.4    P2p
Switch#
```

Figure 14: Verifying STP Mode Operation

You also need to configure edge ports and point-to-point ports in RSTP.

To configure an edge port, simply enable PortFast:

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

To configure a port as a point-to-point interface, use the following interface configuration command:

```
Switch(config-if)# spanning-tree link-type point-to-point
```

In this output you can see the STP mode, Bridge ID, timers, ports that belong to the VLAN, their roles, port priorities and port types.

Multiple Spanning Tree Protocol (MST)

CST had the limitations of having only one process of STP for all VLANs in the domain. A small problem in single instance of STP can cause network problems in multiple areas. PVST was developed to acknowledge this issue and allow one STP instance for every VLAN in the domain. PVST solved one problem but also created another in some very large deployments. The PVST can create issues related to switch CPU and memory resources. As the network grows and requires more and more VLANs, it might be less than optimal and redundant to have one instance of STP per VLAN, since it is likely that only a few redundant paths are available for even dozens of VLANs. To solve this problem, Multiple Spanning Tree (MST) was developed.

MST allows a network administrator to use one or more STP instances with every STP instance able to handle a group of VLANs as opposed to one instance for a single VLAN. The network administrator can group VLANs into one STP instance or more. MST is defined in IEEE 802.1s.

There are important design considerations prior to implementing MST in a network:

- Determine the amount of possible logical topologies with the existing physical network and connections, and determine how many instances of STP are required to support such topologies.
- Determine which VLANs will be mapped to which STP instance. The criterion is based in security and traffic type and amount considerations.

MST Regions are groups of switches running MST with a set of common parameters. Cisco compares a MST region with a BGP Autonomous System, which is basically a group of network devices under a common administration.

Most networks don't need more than one region, but multiple regions are also supported. All switches within the region must run the instance of MST with the following attributes:

- MST configuration name (32 characters)
- MST configuration revision number (0 to 65535)
- MST instance-to-VLAN mapping table (4096 entries)

The network administrator should propagate the configuration and attributes throughout the region. This has to be done manually or with Simple Network Management Protocol (SNMP). SNMP is outside the scope of the SWITCH exam but as a network engineer it is extremely important to be proficient in that technology. You can find a lot of information and very good books on the subject.

If for any reason two switches have a MST different attribute, the switches believe that they belong to different regions.

Most of the regions information is propagated inside BPDU frames. The revision number, region name and a digest number of the VLAN-to-instance mapping are sent. The digest is numerical value calculated using a mathematical function (the details of this calculation are outside the scope of the SWITCH 642-813 exam). When a neighboring switch receives the BPDU it checks for this digest value and compares it to its own. If they are the same, the BPDU belongs to the same region. Else it knows the port on which it was received is a region boundary. This means the switch belongs to more than one region.

An MST switch can handle only one Internal Spanning Tree (IST) and one or more Multiple Spanning Tree Instances (MSTI). This is defined in the IEEE 802.1s standard. The Cisco implementation of MST happens to currently support 16 instances: One IST instance and 15 MSTIs.

IST Instances

MST was designed to interoperate with legacy STP protocols. The IST instance is simply an RSTP instance inside the MST region. Its function is to represent the entire MST region as a CST switch to the outside world.

The IST instance communicates by sending BPDUs through the native VLAN of the CST trunks to the rest of the CST switches/topology.

The exact mechanism through which IST makes the participating switches appear as one CST bridge is out of the scope of the SWITCH exam and this guide.

MST Instances

MST Instances (MSTIs) are RSTP instances that exist exclusively inside a MST region. Unlike ISTs, MSTIs never interact with switches outside the region. MST uses only one Spanning Tree instance outside of the region. MSTIs don't have an outside counterpart, and they never send BPDUs to the outside; that is an IST role.

Inside the MST region, MSTIs don't send independent BPDUs. They send a RSTP BPDU for the whole region (IST BPDU or MSTI 0 BPDU) and append particular information about a specific instance additionally. This information for specific MSTIs is placed in the M record, which is where additional information pertaining to specific MSTI is placed inside the BPDU. Remember that a MSTI instance is only valid inside the MST region. It doesn't matter if an adjacent region has the same MSTI in use, The MSTI information is only locally significant.

MST can only run with RSTP inside the region and CST on the outside. A MST can only interact with one CST instance on the outside. That creates a problem when the outside switches are running PVST+. Cisco addressed that problem by having the IST replicate its CST BPDU to all PVST+ instances, to simulate a PVST+ behavior. Cisco switches automatically detect the PVST+ neighbor when they receive multiple BPDUs for several instances.

The following are the commands required to configure MST in a switch:

```
Switch(config)# spanning-tree mode mst
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# name name
Switch(config-mst)# revision version
Switch(config-mst)# instance instance-id vlan vlan-list
Switch(config-mst)# show pending
Switch(config-mst)# exit
```

In the first line MST is selected as the STP mode. In the second line we enter MST configuration mode.

The **name** MST configuration mode command sets the region configuration name.

The revision MST configuration command is used to track changes to the MST configuration. Just like VTP, the configuration name, and revision number must match in all switches in the MST region.

Instance *instance-id* **vlan** *vlan-list* maps VLANs to a MST instance. *Instance-id* takes the value from 0-15 and carries topology information for the VLANs specified with the *vlan-list*.

The **show pending** MST configuration commands show all the changes you have made and have not yet saved. Please remember that all configuration commands and changes made in the MST configuration mode only take effect and are saved to the running-configuration file when the **exit** command is used.

After MST is configured, PVST+ ceases and RSTP starts operating, since a switch cannot operate with both Spanning Tree versions at the same time.

All standard STP parameters are also present in MST and their respective commands are practically the same, with the difference that you should now use the **mst** command and instance-id before most parameters. Here is a list of the most common commands:

To set the root bridge:

```
Switch(config)# spanning-tree mst instance-id root {primary |
secondary} [diameter diameter]
```

To set the bridge priority:

```
Switch(config)# spanning-tree mst instance-id priority bridge-priority
```

To set the port cost:

```
Switch(config)# spanning-tree mst instance-id cost cost
```

To set port priority:

```
Switch(config)# spanning-tree mst instance-id port-priority port-priority
```

To set STP timers:

```
Switch(config)# spanning-tree mst hello-time seconds
Switch(config)# spanning-tree mst forward-time seconds
Switch(config)# spanning-tree mst max-age seconds
```

Aggregating Switch Links

EtherChannel is the technology developed to allow network administrators to scale link bandwidth by aggregating, or combining, up to 8 Fast Ethernet, Gigabit Ethernet or 10-Gigabit Ethernet links. This allows for easier expansion and growth without requiring expensive equipment every time more throughput is necessary.

EtherChannel also solves the issues with redundant parallel paths and switching loops, because it creates a logical link out of two to eight individual links. These logical links can be used as access ports or trunk ports. And if there is a physical layer problem on one of the connections, the end-user is not affected because traffic will simply be redistributed over the remaining links in the EtherChannel bundle.

According to Cisco, the main benefits of the EtherChannel technology are:

- **Standards-based** –EtherChannel was developed to be compatible with the IEEE 802.3 standard. It uses Ethernet mechanisms to provide some of the features present in this technology.
- **Multiple platforms** – EtherChannel is flexible and can be used anywhere in the network that bottlenecks are likely to occur. It can be used in network designs to increase bandwidth between switches and between routers and switches—as well as providing scalable bandwidth for network servers, such as large UNIX servers or PC-based Web servers.
- **Flexible incremental bandwidth** – EtherChannel provides bandwidth aggregation in multiples of 100 Mbps, 1 Gbps, or 10 Gbps, depending on the speed of the aggregated links. For example, network managers can deploy EtherChannel technology that consists of pairs of full-duplex Fast Ethernet links to provide more than 400 Mbps between the wiring closet and the data center. In the data center, bandwidths of up to 800 Mbps can be provided between servers and the network backbone to provide large amounts of scalable incremental bandwidth.
- **Load balancing** – EtherChannel is composed of several Fast Ethernet links and is capable of load balancing traffic across those links. Unicast, broadcast, and multicast traffic is evenly distributed across the links, providing higher performance and redundant parallel paths. When a link fails, traffic is redirected to the remaining links within the channel without user intervention and with minimal packet loss.
- **Resiliency and fast convergence** – When a link fails, EtherChannel provides automatic recovery by redistributing the load across the remaining links. This is done in less than one second. This convergence is transparent to the end user—no host protocol timers expire, so no sessions are dropped.
- **Ease of management** – EtherChannel takes advantage of Cisco experience developed over the years in troubleshooting and maintaining Ethernet networks. Existing network probes can be used for traffic management and troubleshooting, and management applications such as CiscoWorks and other third-party network management applications are now EtherChannel-aware.
- **Transparent to network applications** – EtherChannel does not require changes to networked applications. When EtherChannel is used within the campus, switches and routers provide load balancing across multiple links transparently to network users. To support EtherChannel on enterprise-class servers and network interface cards, smart software drivers can coordinate distribution of loads across multiple network interfaces.
- **Compatible with Cisco IOS** – EtherChannel connections are fully compatible with Cisco IOS virtual LAN (VLAN) and routing technologies. The Inter-Switch Link (ISL) VLAN Trunking Protocol (VTP) can carry multiple VLANs across an EtherChannel link, and routers attached to EtherChannel trunks can provide full multiprotocol routing with support for hot standby using the Hot Standby Router Protocol (HSRP).

- **100 Megabit, 1 Gigabit, and 10 Gigabit Ethernet-ready** – EtherChannel is available in all Ethernet link speeds. EtherChannel technology allows network managers to deploy networks that will scale smoothly with the availability of next-generation, standards-based Ethernet link speeds.
- **Interoperability with Coarse Wavelength Division Multiplexing (CWDM) Gigabit Interface Converters (GBICs)** – By simultaneously implementing Gigabit EtherChannel and CWDM technologies, network managers can increase the bandwidth of their links without having to invest in new long runs of fiber. CWDM technologies allow the traffic aggregated by the Cisco EtherChannel link to be multiplexed on to a single strand of fiber.

EtherChannel must be formed bundling up to eight links of the same type meaning the same speed and media type. Fast Ethernet bundles form Fast EtherChannels, Gigabit Ethernet form Gigabit EtherChannels, etc.

All aggregated ports must belong to the same VLAN if they are access ports. If they are trunks, they must have the same native VLAN and allow the same set of VLANs to traverse them. All ports in the bundle must be configured with the same duplex mode (full duplex) and speed. They also must use the same STP type and have identical STP settings.

Distributing Traffic in EtherChannel

Load is not automatically balanced across all links in the EtherChannel. Frames are forwarded based on the result of a hashing algorithm. This hashing algorithm can use a wide variety of information to calculate load and act on it, including IP addresses, physical addresses and port numbers.

The hashing algorithm selects the specific port to be used for forwarding specific frames to specific ports. If two addresses or ports are hashed, the selection will require an exclusive-OR (XOR) operation with the rightmost bit(s) of the address(es) or port(s). If only one address is to be hashed, the rightmost bits will dictate the port in the bundle that is to be used to forward frames.

This is important and must be understood. The amount of bits selected are the ones required to represent the port number (inside the bundle) in binary. If you have 8 ports bundled, the first port is port 0 and last is port 7. You need 3 bits to represent 7 in binary; you will need those bits to identify which port is going to be forwarding the frames or packets. For example, 010 in binary is 2. That means that traffic will be sent across Channel 3 in the bundle. If we assume you are using source IP address as the load balancing method, and the source IP address is 192.168.100.19.

19 = 0001 0011. We need to use the three rightmost bits which are 011 (3 in decimal). The port that will be used is link 3 in the bundle.

In case two addresses are being hashed, the XOR operation must be done. Remember that in XOR two different bits will produce 0 and equal bits will produce 1.

As you can see, this form of load balancing/distribution based on source and destination addresses can cause a link to handle a lot more transit than others, creating a load imbalance. This is what happens, for instance, when you have an EtherChannel connected to a server and use the destination address as the load balancing method. All the traffic on the heavily used server will always go across one specific link and never be load balanced. To solve this issue it is recommended that you use a combination of source and destination addresses and port numbers. That way you can distribute traffic according to not only hosts or destination but also applications in certain hosts.

The hashing operation can be performed with MAC or IP addresses or the combination of both. The command to configure load balancing is the following:

```
Switch(config)# port-channel load-balance method
```

Notice the load balancing is set globally. You cannot set the load balancing method on a per port basis. The method variable can take several values, and they are listed in the following table:

***method* Value**	**Hash Input**	**Hash Operation**	**Switch Model**
src-ip	Source IP address	bits	All models
dst-ip	Destination IP address	bits	All models
src-dst-ip	Source and destination IP address	XOR	All models
src-mac	Source MAC address	bits	All models
dst-mac	Destination MAC address	bits	All models
src-dst-mac	Source and destination MAC	XOR	All models
src-port	Source port number	bits	6500, 4500
dst-port	Destination port number	bits	6500, 4500
src-dst-port	Source and destination port	XOR	6500, 4500

Figure 15: Types of EtherChannel Load-Balancing Methods

The default method for pure Layer 2 switches is source MAC address (**src-mac**). If Layer 3 switching is being used on the switch, the source and destination **IP address** (src-dst-ip) method is recommended.

When IP is not the Layer 3 protocol in use, you must use MAC addresses to determine the link to be used to forward traffic.

As network administrator, you should check if the current configurations are producing load (traffic) imbalances and correct them with one of the methods provided. To see what load balancing method is in use and the amount of traffic that has gone through each link, use the show EtherChannel port-channel EXEC configuration command.

An important situation to notice is when EtherChannels are configured to connect routers. Both MAC addresses and IP addresses will always be the same, making both methods forward through the same links. To address this issue, you should configure the channels to load balance using port numbers to forward frames based on applications.

Please note that when IP load-balancing is selected and there are no IP packets to forward, the switch or router will fall back to MAC address indexing.

As we mentioned earlier, EtherChannels also help network administrators deal with switching loops and multicast/broadcast traffic. When an inbound multicast or broadcast traffic is received in a link of an EtherChannel, the multicast/broadcast is never sent back through any of the bundled links. This is because the multiple bundled links are treated as if they were a single physical link. Equally, outbound multicast or broadcast traffic is load balanced like any other frame or packet: the broadcast or multicast frame or packet is part of a hashing calculation to determine the link through which it is going to be forwarded.

EtherChannel Negotiation Protocols and Configuration

There are means to provide EtherChannels negotiation and dynamic link configuration. There are two protocols available to negotiate bundled links between switches: **Port Aggregation Protocol** (PAgP), which is a Cisco proprietary, and **Link Aggregation Control Protocol** (LACP).

Port Aggregation Protocol

This is Cisco's Proprietary link aggregation protocol. Cisco switches exchange PAgP packets over Ether Channel capable links. The protocol learns dynamically the capable ports in the LAN and then informs the other LAN ports. Once it has identified the links, it provides the means to group the ports into an Ether Channel. The Ether Channel is then added to the spanning tree topology as a single link.

PAgP packets are only exchanged between ports configured as desirable or auto modes.

Ports with the same neighbor ID and port capability are grouped together as a bidirectional point-to-point Ether Channel. These capabilities are trunking state, duplex mode and speed. If the ports are trunking, they must have the same native VLAN, and must allow the same VLANs through them.

Switch ports can form Ether Channels if they are in a compatible mode (these modes are very similar to the dynamic trunking modes). A port configured as **desirable** actively tries to form an ether channel, sending PAgP packets and initiating the negotiation, with other ports configured as **desirable** and also with ports configured as **auto**.

Ports configured as auto don't form Ether Channels with other ports in **auto** because they don't initiate negotiation. **Auto** is the default.

PAgP negotiation is the default. To configure it on Cisco Catalyst Switches use the following commands:

```
Switch (config) # interface type mod/num
Switch (config-if) # channel-protocol pagp
Switch (config-if) # channel-group number mode {on | {{auto |
desirable} [non-silent]}}
```

The aggregation protocol is defined with the channel-protocol interface configuration command.

The EtherChannel must be configured with a unique number, from 1 to 64, and the operation mode must be defined too. On makes the Ether Channel unconditionally and no negotiation takes place; auto waits for the other end to ask to form the Ether channel and accepts if PAgP packets are received and finally desirable actively tries to form the Ether Channel starting negotiation and sending PAgP packets.

The non-silent submode means the port will be required to hear PAgP packets in order to attempt to form an Ether Channel. The default submode in auto or desirable is silent, in which the port doesn't have to wait for PAgP packets in order to attempt to become a part of an Ether Channel.

The following is a sample configuration:

```
Switch# configure terminal
Switch (config)# port-channel load-balance src-dst-port
Switch(config )# interface port-channel 1
Switch(config-if)# ip address 172.32.52.10 255.255.255.0
Switch(config-if)# interface range fastethernet 0/1 - 5
Switch(config-if)# no ip address
Switch(config-if)# channel-protocol pagp
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# end
```

We first establish the load-balancing method with port-channel load-balance src-dst-port global configuration command and then create the port-channel or EtherChannel 1.

In the fourth line we assign an IP address to the EtherChannel. In the following line we use the interface range to enter interface configuration command for several interfaces. We then define the EtherChannel negotiation protocol. Finally we add them to the channel-group 1 and use the desirable mode.

The individual links cannot have IP addresses assigned to them. Again, this is because the multiple bundled links are treated as if they are a single physical link. The interface that can be configured for this bundle is the virtual port-channel interface and not the actual individual links within the bundle.

Remember the default sub-mode is silent, and doesn't have to be specified.

Link Aggregation Control Protocol

LACP is the IEEE standards-based link aggregation protocol. LACP is defined in the IEEE 802.3ad and is also known as the 802.3 clause 43, link aggregation. Just like PAgP, LACP neighbors are identified and port group capabilities are compared, with the exchange of LACP packets. LACP also assigns roles to EtherChannel's ports.

The switch with the lowest system priority, a 2-byte value followed by a 6-byte switch MAC address, is allowed to make decisions about what ports are actively participating in the EtherChannel at a given time.

The LACP port priority is used to select which port becomes active in the bundle. The port priority is a 4 byte value, 2 byte priority and 2 byte port number, and in the aggregated link, a lower value means a more preferred priority. This is very important because up to 16 links can be defined in the EtherChannel bundle but only 8 will be actively forwarding frames. The 8 links with the lowest priority values are selected as active and the remaining links are put in a standby state and become active if one of the active links fails at the physical layer.

After the EtherChannel is up and running with the best links bundled it enters the STP topology as one single port.

Similar to PAgP, LACP can select ports to actively try to become part of the EtherChannnel (**active** mode in LACP and **desirable** in PAgP), and **passive** (**auto** in PAgP and **passive** in LACP) in which switches only negotiate an EtherChannnel if the other end starts the negotiation.

The LACP configuration is very similar to PAgP. An example:

You can use the following configuration commands to accomplish this:

```
Switch(config)# lacp system-priority 100
Switch(config)# interface range gig 2/1 – 4 , gig 3/1 – 4
Switch(config-if)# channel-protocol lacp
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# lacp port-priority 100
Switch(config-if)# exit
Switch(config)# interface range gig 2/5 – 8 , gig 3/5 – 8
Switch(config-if)# channel-protocol lacp
Switch(config-if)# channel-group 1 mode active
```

We first set the system priority, in this case 100. Then we enter interface configuration mode with the int range command, and use LACP as the EtherChannel negotiation protocol with the **channel-protocol lacp** command. The port priority is set to 100 in ports 1 through 4 of slot 2, and 1 through 4 of slot 3. Those same interfaces belong to the EtherChannel 1 and are set to active mode (the ports will try actively to negotiate an EtherChannel by sending LACP packets).

The Gigabit interfaces 2/5 – 8 and 3/5 – 8 are configured in the same manner with the exception that the port priority is left to the default, 32,768. In this scenario, the port numbers will be the tie breakers in the election of the active links in the EtherChannel. The lower port numbers will be more preferred.

Troubleshooting EtherChannels

The most common cause of problems in EtherChannels comes from links with different port settings and abilities. If you encounter problems, check that all the required settings match on both ends.

The following are important points to have in mind when troubleshooting EtherChannel links:

- PAgP when set to desirable mode, actively tries to form the EtherChannel, but the other end needs to be configured as desirable or auto.
- LACP tries to bring up an EtherChannel when one side of the bundle is configured as active. The other end of the bundle must be configured as passive or active in order for the EtherChannel to come up.
- EtherChannel auto (PAgP) or passive (LACP) modes participate in channel protocol passively, which means they need to hear from a neighbor trying to actively (desirable or active modes) bring an EtherChannel up.
- PAgP desirable and auto modes default to the silent sub-mode, which means they will actively try to form an EtherChannel without hearing PAgP packets. If the non-silent submode is selected, PAgP packets must be received in order for the EtherChannel to form.

There are several important commands when troubleshooting EtherChannels:

- **Show etherchannel summary** EXEC command will show you the ports in the channel. They will be flagged to indicate the current port state.

```
Switch# show etherchannel summary
Flags:  D - down            P - in port-channel
        I - stand-alone     s - suspended
        H - Hot-standby (LACP only)
        R - Layer3          S - Layer2
        u - unsuitable for bundling
        U - in use          f - failed to allocate aggregator
        d - default port
Number of channel-groups in use:    1

Number of aggregators:              1
Group   Port-channel    Protocol     Ports
--------+---------------+------------+---------------
1       Po1(SU)         PAgP         Fa0/1(P) Fa0/2(P)
                                     Fa0/3 Fa0/4(P)
                                     Fa0/5(P)

Switch#
```

Figure 16: Verifying EtherChannel Operation

Under port channel, the (SU) message should appear (layer 2, in use) when the EtherChannel is operational. The "I "(stand-alone) means the link is up but it is not part of the EtherChannel. Basically the remaining flags and parameters are self-explanatory.

- The show etherchannel port EXEC command gives a lot of information, from negotiation protocol (PAgP or LACP) and mode, to priority settings. You can also see information of the other end, such as the MAC addresses of the other end's port, module and port number, and partner's name.
- The show interface type mod/num etherchannel EXEC command shows all active EtherChannel configuration settings for a port. This command will display errors regarding the EtherChannel formation, such as duplex or speed mismatches, different native VLANs set in both ends, different set of allowed VLANs in the trunk or trunking state mismatch.
- The show etherchannel load-balance shows the hashing algorithm or EtherChannel load-balancing method in use.

The following table shows a list of very useful show commands when troubleshooting EtherChannels and their display:

Display Function	Command Syntax
Current EtherChannel status of each member port	**show etherchannel summary** **show etherchannel port**
Time stamps of EtherChannel changes	**show etherchannel port-channel**
Detailed status about each EtherChannel component	**show etherchannel detail**
Load-balancing hashing algorithm	**show etherchannel load-balance**
Load-balancing port index used by hashing algorithm	**show etherchannel port-channel**
EtherChannel neighbors on each port	**show {pagp | lacp} neighbor**
LACP system ID	**show lacp sys-id**

Figure 17: EtherChannel Troubleshooting Commands

Domain 2: Implementing a Security Extension for a Layer 2 Solution

As networks have grown, so have their requirements. Corporations now run mission critical applications in a network environment and the means to protect data have become crucial. Securing network information while guaranteeing access of said data to allowed users has now become a major field in the realm of Information Technology, Network Engineering and Administration.

As technology develops, attacks to steal data and disrupt operations become increasingly more dangerous and even easier for attackers. Protecting the network environment from attacks from the inside and from the outside is an everyday challenge for Information Security Analysts and engineers.

As a CCNP you are required to know how to protect the network from common and sophisticated attacks. Cisco switches are equipped with powerful means to prevent such attacks, and we will be reviewing some of them in this section. We'll review the methods available to secure switches in general, from best practices to specific prevention and mitigation of some of the most common and disrupting attacks.

Port Security

Cisco Catalyst switches offer the port security feature to control access to switch ports based on MAC addresses. Port security is most often configured in access layer switches, where users connect to the network. It is enabled in a per-interface basis.

After the port is configured with port security, it will learn and keep track of one or more MAC addresses and will expect only them to connect to the switch port. This is called the *sticky* feature. By default, port security enabled interfaces will only accept one MAC address, but the port can be configured to accept up to 1024 MAC addresses. Learned addresses can also be aged out if they are not heard on the interface after a specified period. Aging does not occur by default. MAC addresses can also be statically defined.

A violation occurs when more than the allowed maximum MAC addresses are learned on the port or when an unspecified, unauthorized MAC address is learned on the port. When a violation occurs, three actions are possible:

- **Shutdown –** The port is put in the "errdisable" state which effectively shuts the port down. The network administrator must be activated with a **shutdown** and then **no shutdown** interface configuration command or with the errdisable recovery configuration feature.
- **Restrict –** The port is kept up but all frames from the violating MAC address are dropped. The switch keeps track of the amount of violations and can be configured to register the violations in SNMP and a syslog message.
- **Protect –** This works exactly as **restrict** but no record from the violation is kept.

The following are the commands used to configure the different features of port security.
These enable port security on the interface:

```
Switch(config-if)# switchport port-security
```

The following command specifies how many addresses are allowed in the interface. Remember by default, port security enabled interfaces will only learn one address and consider the appearance of a second MAC address a violation:

```
Switch(config-if)# switchport port-security maximum max-addr
```

The *max-addr* value ranges from 1 to 1024.
To statically configure a port security allowed MAC address use the following interface configuration command:

```
Switch(config-if)# switchport port-security MAC-address MAC-addr
```

The *MAC-addr* value must be given in triplet-dotted (xxxx.xxxx.xxxx) format.

You need to define the port security violation action. The default option is shutdown. To change it, use the following command:

```
Switch(config-if)# switchport port-security violation {shutdown |
restrict | protect}
```

You can encounter a situation when you need to clear the port cache in order to allow a new set of hosts and their MAC addresses to be allowed in the port. You can do this with the following command:

```
Switch# clear port-security dynamic [address MAC-addr | interface type mod/num]
```

The following is a sample configuration where we will use the port security feature in a Fast Ethernet port. We'll statically configure an allowed MAC address and will set the maximum hosts allowed at 5. The violation will be set to restrict, which will cause the unauthorized MAC address frames to be dropped and a log message issued to the console or syslog server if one is configured, while keeping the link up for authorized hosts:

```
Switch(config)# fastethernet 0/1
Switch(config-if)# switchport access vlan 10
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security violation restrict
Switch(config-if)# switchport port-security maximum 5
Switch(config-if)# switchport port-security MAC-address 0022.ff7d.b77c
```

In this configuration the port is an access port that belongs to VLAN 10. Port security will learn 4 additional MAC addresses besides the **0022.ff7d.b77c** host that was statically configured, making a total of 5 allowed MAC addresses. If a fifth MAC address appears in the source address of a frame, the frame will be dropped and a log message will appear on the console. The log message is the following:

```
Apr 5 10:18:41.888 EDT: %PORT_SECURITY-2-PSECURE_VIOLATION: Security
violation occurred, caused by MAC address 0000.5455.4e01 on port
FastEthernet0/1.
```

There are several troubleshooting commands that you can use to verify the port security status (violation penalty, Aging Time if any, configured MAC addresses, ports in the errdisabled state, and a summary of port security status in the system (switch)). The following are the commands and their outputs:

```
Switch# show port-security interface fastethernet 0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 5
Total MAC Addresses        : 0
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address        : 0022.ff7d.b77c
Security Violation Count   : 0
Switch#
```

Figure 18: Verifying Port Security Status with show port-security interface

For a system-wide summary of ports with port security enabled, use the **show port-security** EXEC command. This command is useful to see the maximum allowed MAC addresses per port and the violation setting. You can also see how many addresses have been learned on the port:

```
Switch# show port-security
Secure Port     MaxSecureAddr   CurrentAddr     SecurityViolation       Security Action
                (Count)         (Count)         (Count)
-------------------------------------------------------------------------------------
Fa0/1           5               1               0                       Restrict
Fa0/10          1               0               0                       Shutdown

Total Addresses in System (excluding one mac per port)      : 0
Max Addresses limit in System (excluding one mac per port)  : 6096
Switch#
```

Figure 19: Verifying System-Wide Port Security

You can also display a summary of ports in the errdisabled state with the following command:

```
Switch# show interfaces status err-disabled

Port    Name        Status          Reason
Fa0/10  Test port   err-disabled    psecure-violation
Switch#
```

Figure 20: Verifying Ports in the *errdisable* State

There are two ways to re-enable a port in the errdisable state. You can manually shut it down and then enable it with the **no shutdown** interface configuration command, or you can configure the switch to automatically recover errdisabled ports after a specified period of time. To manually make an errdisabled port active, use the following commands:

```
Switch(config)# interface type mod/num
Switch(config-if)# shutdown
Switch(config-if)# no shutdown
```

Port-Based Authentication

AAA stands for authentication, authorization and accounting within Cisco IOS devices. Port-based authentication is a catalyst switch function that integrates AAA authentication and port security features. When it is enabled, the user must be authenticated before a port can forward any kind of traffic. After the user is authenticated the switch makes the port fully functional.

Port-based authentication is defined in the IEEE 802.1x standard. In order for hosts to use features, both the switch and end user's device must support the 802.1x standard using the Extensible Authentication Protocol over LAN (EAPOL). In order for a port-based authentication enabled switch to allow a user to transmit to the network the user must authenticate. If the switch doesn't support the 802.1x and the PC does, the user will be able to connect, as the PC will stop the protocol. If for example, the switch has port-based authentication and the endpoint doesn't support it, the port will remain in the unauthorized state and the endpoint (the switch port to which the host is connected) won't be allowed to transmit.

802.1x EAPOL is a layer 2 protocol, and that is why the PC or endpoint requires compatibility before being able to connect to the network. The PC won't get an IP address from a DHCP server or access any other network service before it is authenticated in the switch.

802.1x EAPOL configuration:

The 802.1x protocol uses Remote Authentication Dial-In User Service (RADIUS) servers to handle authentication. The Cisco access control server (ACS) can be used as a RADIUS server. RADIUS server configuration is out of the scope of the SWITCH exam. Cisco defines a six step procedure to configure 802.1x port-based authentication:

- **Step 1**:Enable AAA on the switch:

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

AAA is not enabled by default. This command enables AAA and disables any old AAA models on the switch. This will wipe out any existing AAA configurations on the switch.

- **Step 2**: Define external RADIUS servers.

First, define each sever with its secret shared password. This key is transparent to the port-based authentication client. It is only known by the switch and the server. Use the following command to define the RADIUS server:

```
Switch(config)# radius-server host {hostname | ip-address} [key string]
```

You can define redundant RADIUS servers by repeating the previous command.

- **Step 3**: Define the authentication method for 802.1x:

The following command makes all RADIUS authentication servers defined in the switch to be used for 802.1x authentication:

```
Switch(config)# aaa authentication dot1x default group radius
```

- **Step 4**: Enable 802.1x on the switch:

```
Switch(config)# dot1x system-auth-control
```

- **Step 5:** Configure each switch port that will use/require 802.1x authentication:

```
Switch(config)# interface type mod/num
Switch(config-if)# dot1x port-control {force-authorized | force-
unauthorized | auto}
```

The possible 802.1x states are:

- **Force-authorized –** The port is forced to always authorize any connected device without any authentication required. This is the default state for switch ports when 802.1x is enabled.
- **Force-unauthorized** – The port is never allowed to authorize a connected client. The port cannot move to the authorized state regardless of the connected host or hosts, which makes it the same as if it was shut down.
- **Auto –** The port uses 802.1x to authenticate a connected host and move from the unauthorized state to the authorized stated. This requires an 802.1x capable client.

Notice that the default port state is force-authorized, which allows any client to pass traffic and access the network without any authentication. To effectively require authentication, you need to explicitly set each port to the **auto** state by issuing the **dot1x port-control auto** command.

- **Step 6**:Allow multiple hosts in a switch port:

If the switch is going to expect more than a host in a certain switch port (the port is connected to another switch or hub), you need to use a command to allow this behavior. By default the switch port is considered an access port with a single host connected. If you have an additional hub hanging off the access port you need to modify the default behavior. To change this, use the following command:

```
Switch(config-if)# dot1x host-mode multi-host
```

To verify 802.1x operation in each switch port you should use the show dot1x all EXEC command.
The following is a configuration example:

```
Switch(config)# aaa new-model
Switch(config)# radius-server host 192.168.1.1 key PrepLogic
Switch(config)# radius-server host 192.168.1.2 key PrepLogicSWITCH
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# dot1x system-auth-control
Switch(config)# interface range FastEthernet0/1 - 10
Switch(config-if)# switchport access vlan 10
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x port-control auto
```

Mitigating Spoofing Attacks

One of the most common forms of attacks in today's networks are man in the middle attacks, performed by spoofing information to make switches and network devices believe there are authorized, known hosts or network devices. The attacker tries to spoof a router and then becomes the "man in the middle", receiving packets destined to the router and then forwarding them, in order to make the attack transparent to the user.

DHCP Spoofing; Description and Mitigation

Attackers take advantage of the DHCP process when a client broadcasts a DHCP request out on their subnet. What can happen is that the attacker will attempt to reply to the DHCP request before the real DHCP server does. The attacking device will send a response to the client making the request specifying its IP address as the default gateway. By doing that, the attacker receives every packet destined to another network and can inspect and manipulate information before forwarding it.

Cisco switches use DHCP snooping to prevent this type of attacks. DHCP snooping consists of categorizing switch ports as trusted or untrusted. When a DHCP reply message is received in an untrusted port, the packet is discarded and the port put immediately in the errdisabled state, effectively shut down. DHCP reply messages are only allowed on defined trusted ports when DHCP snooping is enabled.

DHCP snooping keeps track of MAC addresses, IP addresses leased, lease time and other information about trusted, legitimate replies. This information can be used to keep track of a lot of network activity between authorized hosts and network devices. DHCP snooping is generally used at the access layer. DHCP snooping can be activated in a per-VLAN basis. When it is active in a VLAN, the switch builds a table of IP addresses to MAC-address bindings for the DHCP clients on that VLAN.

To enable DHCP snooping, you need to first enable it globally, and then identify the VLANs where the DHCP snooping will be implemented. Finally, you need to define the trusted ports. Notice that the default state is untrusted, which means that if you don't set certain ports as trusted and connect a DHCP server, or DHCP traffic goes through the switch, the ports will be put to the errdisable state.

DHCP option 82, Subscriber Identification, is a very helpful feature, defined in the RFC3046. When a DHCP request is heard in an untrusted port, the switch adds its own MAC address and the port identifier in the Option 82 field and forwards the frame out a trusted port so that it can reach a trusted DHCP server. The Option 82 is enabled by default when DHCP snooping is enabled.

To configure DHCP snooping use the following commands:

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan vlan-id [vlan-id]
Switch(config)# interface type mod/num
Switch(config-if)# ip dhcp snooping trust
```

You can use the show ip dhcp snooping [binding] EXEC command to display the DHCP snooping status. Here is a sample output:

```
Switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
104

Insertion of option 82 is enabled
Interface                  Trusted            Rate limit (pps)
-------------------------  -----------------  --------------------
FastEthernet0/10           no                 5
FastEthernet0/12           no                 5
FastEthernet0/1            yes                Unlimited

Switch#
```

Figure 21: Verifying DHCP Snooping

If you want to display all the DHCP bindings that have been overheard use the **show ip dhcp snooping binding** EXEC command.

IP Source Guard

IP source guard is a security feature that restricts IP traffic on non-routed, Layer 2 interfaces by filtering traffic based on the DHCP snooping binding database and on manually configured IP source bindings. You can use IP source guard to prevent traffic attacks caused when a host tries to use the IP address of another device on the same VLAN to trick others into sending traffic destined to the real PC. This is a classic spoofing scenario.

IP Source Guard can be enabled on interfaces with DHCP snooping enabled. What it does is block any IP address that is not part of a mapping in the DHCP Snooping database or a static entry. A switch with IP Source guard enabled will filter both layer 3 or layer 2 unknown addresses. The mechanism to achieve this is a port access control list (PACL) applied to the interface.

Note that the port ACL (PACL) takes precedence over any router ACLs or VLAN maps that affect the interface. This is because the PACL is applied to the layer 2 interface which is the first point the network has a chance to apply any type of access control.

Basically when IP Source Guard is enabled in the port, the switch tests packets received in the port against one or both of the following conditions:

- The source IP address must be identical to the IP address learned by DHCP snooping or a static entry. A dynamic port ACL is used to filter traffic. The switch automatically creates this ACL, adds the learned source IP address to the ACL, and applies the ACL to the interface where the address is learned.
- The source MAC address must be identical to the MAC address learned on the switch port and by DHCP snooping. Port security is used to filter Layer 2 traffic.

If the address is something else than addresses learned by DHCP snooping or statically configured, the switch drops the packet or frame.

To statically configure an address binding (this is done for hosts with static IP addresses –those that are not using DHCP) use the following command:

```
Switch(config)# ip source binding mac-address vlan vlan-id ip-address
interface type mod/num
```

IP Source Guard configuration is very simple. You only need to enter one command in the interface you need to perform the IP source guard:

```
Switch(config-if)# ip verify source [port-security]
```

The **port-security** optional keyword is used to inspect source MAC addresses too. Remember IP Source Guard inspects only source IP addresses if this option is not used.

IP Source Guard Configuration Guidelines as proposed by Cisco Systems.

This section describes the guidelines for configuring IP source guard in your network:

- IP source guard is supported on the policy feature card (PFC) 3 and later versions.
- IP source guard is not recommended on trunk ports.
- IP source guard cannot coexist with PACLs.
- IP source guard is not supported on EtherChannel-enabled ports, and EtherChannel is not supported on IP source guard-enabled ports.
- VLAN-based ACL features, such as static ARP inspection, are disabled when you enable IP source guard.
- It is recommend that you enable high availability when using dynamic ARP inspection (DAI), DHCP snooping, and IP source guard. If high availability is not enabled, clients have to renew their IP addresses for these features to work after a switchover. A switchover is the manual process of transferring the switching functions to a second redundant switch. This is done mostly to perform network upgrades.
- DAI will be explained in more detail, next. High Availability will be explained in detail in its own domain in this guide.

There are two troubleshooting commands that you should know:

```
Switch# show ip verify source [interface type mod/num]
```

This command shows the IP Source Guard status. The next command should be used when the network administrator needs to know the information in the IP source binding database. There you can see both dynamically and statically configured bindings:

```
Switch# show ip source bindng [ip-address] [MAC-address] [dhcp-
snooping | static] [interface type mod/num] [vlan vlan-id]
```

Dynamic ARP Inspection (DAI)

Address Resolution Protocol (ARP) is the protocol used when a host has a neighbor IP address and needs its MAC address to communicate in Layer 2. The host transmits an ARP request and waits for an ARP reply from the host with the announced IP address. This functions perfectly between trusted users, but also gives a chance for attackers to reply with their own MAC address and start receiving frames that were supposed to be sent to another network user. This is another form of man-in-the-middle attack.

This attack is called ARP poisoning or ARP spoofing. The attacker receives frames destined to another user and forwards them after using for different purposes, making the attack transparent to end users. To prevent this, Cisco Catalyst Switches use Dynamic ARP Inspection (DAI).

DAI works very similarly to DHCP snooping. It classifies ports as trusted and untrusted and keeps a database of MAC-IP bindings. If an ARP reply is received in an untrusted port, the packet is inspected and if the MAC-IP binding doesn't appear as a known and trusted binding, the packet is dropped and a console log message is generated. No inspection takes place when ARP replies are received in trusted ports.

A switch gathers information from trusted MAC-IP bindings, from manually configured bindings and also from the DHCP snooping database.

To enable DAI, use the following command:

```
Switch(config)# ip arp inspection vlan vlan-range
```

By default, all switches associated with the VLAN-range specified will be in the untrusted state. You need to define the trusted ports with the following commands:

```
Switch(config)# interface type mod/num
Switch(config-if)# ip arp inspection trust
```

The manually configured bindings that we mentioned earlier refer to those bindings that are not obtained from the DHCP snooping, or hosts with static IP addresses not leased by a DHCP server. In order to allow these kinds of MAC-IP bindings and ARP replies, you need to define an ARP ACL that defines the static MAC-IP bindings that are permitted. Use the following commands to do this:

```
Switch(config)# arp access-list acl-name
Switch(config-acl)# permit ip host sender-ip mac host sender-mac [log]
```

[The previous command must be repeated as many times as necessary until all permitted MAC addresses are permitted]

```
Switch(config-acl)# exit
Switch(config)# ip arp inspection filter arp-acl-name vlan vlan-range [static]
```

The **ip apr inspection filter** command is used to apply the ARP ACL to the DAI. The **static** optional keyword is used to prevent the router from checking the DHCP snooping database against the ARP Reply, and use only the ARP ACL.

Finally, you can further validate the contents of the ARP Reply. The ARP Reply has MAC and IP fields, and by default, these are the only fields checked and validated. This is a problem because you can have an attacker sending corrupt MAC-IP bindings, that don't belong to them and could somehow get inserted in the DHCP snooping database. To prevent this, you can use DAI to inspect the actual source address of the Ethernet frame in which the ARP Reply is encapsulated. To do that, use the following command:

```
Switch(config)# ip arp inspection validate {[src-mac] [dst-mac] [ip]}
```

The **src-mac** checks the source MAC address of the frame and verifies it corresponds with the MAC address inside the ARP Reply.

The **dst-mac** keyword checks the destination MAC address in the Ethernet header and verifies it corresponds with the target MAC address in the ARP reply.

The **ip** keyword checks the sender's IP address in all ARP requests. It verifies the sender's IP address against the target IP address in all ARP Replies.

Best Practices for Securing Cisco Switches

As a CCNP candidate, you are required to be very knowledgeable about network security. Since the point of entry to our networks is generally a switch, it is extremely important to abide by certain rules that harden our security and make the job of the attackers a painful one. These rules are as much logical, software based security, as they are physical.

Two basic recommendations are that you physically secure switches in secure closets where only the network administrator has access. Secondly, you should limit the number of services running on your production Cisco equipment. This refers to limiting the "platform of attack" that broadens when a new service is provided. For instance, if network administrators are proficient at the command line interface, and there is no need to use the switch as a web server, you should not set the switch as a web server. Allowing the switch as a web server provides a service, and also opens ports or allows certain ports to be listening for information. This condition can be exploited by attackers, and that is what the concept of "platform of attack" refers to, and that is why it is important to limit it, or, better yet, not make it unnecessarily bigger.

Cisco has some recommendations for securing Cisco Switches that you must be aware of as a CCNP and for the SWITCH exam:

- **Configure secure passwords**: Use enable secret as opposed to enable password to set the privileged-level password. Enable secret offers a stronger password encryption, and when used with the service password-encryption command, the password cannot be seen in clear text when you use the **show run** command. You should also use the service password encryption on passwords that are not using enable secret passwords.
- The use of **external** AAA servers is recommended. AAA servers should be used to authenticate administrative users, and to keep passwords and usernames safe in a secure location. This also provides for more centralized, scalable network management than having all user credentials locally in every switch.
- AAA is a very **powerful** tool. Its configuration and details are out of the scope of this guide and the SWITCH exam, but you should learn about it. The CCNA Security Official Certification Guide by Cisco Press has a very comprehensive introduction to it.
- **Use system banners**: They should be used to let users know about acceptable use policies and also to let unauthorized users know they are not welcome, and that they are breaking the law and might be prosecuted. You need to avoid welcoming messages. There are judicial precedents of hackers being acquitted because of a poorly crafted system banner.
- The **banner motd** global configuration command is used to present a "message of the day" to authenticated users, after logon. Never divulge information about your network that a malicious user could use to the organization's dismay.
- You should advise possible intruders that unauthorized access is a violation of a law and that violators will be prosecuted. You must never use welcome messages with the banner motd command. There have been cases where attackers have been acquitted or simply not prosecuted because of a friendly MOTD that implies that access to the device is allowed.
- **Secure web interface**: Remember that if you don't need the web interface to manage and monitor the switch, you should simply not activate it or deactivate the service. Remember the concept of attack platform, and how offering more services than necessary makes your system more vulnerable. You disable the web interface with the **no ip http server** and/or the **no ip http secure server** global configuration command.

- If you need to run a web server for whatever reason, it is recommended that you run a secure web server (https). It offers data encryption, which makes it a much more secure protocol than traditional HTTP. You start the secure web server with the **ip http secure server** global configuration command, instead of the **ip http server**.
- Another security measure recommended by Cisco is that you authorize web access only from certain authorized hosts or networks. You do that by defining an access list permitting traffic from the previously established authorized IP address or addresses. Let's make a sample configuration that allows only hosts with IP address 192.168.1.2 to access the secure web server:

```
Switch(config)# access-list 10 permit 192.168.1.2 0.0.0.0
Switch(config)# ip http secure server
Switch(config)# ip http access-class 10
```

- **Secure Switch console**: Regardless of physical security, you should always configure a password for console access to the switch.
- **Secure virtual terminal access:** Always configure authentication to access through terminal (vty) lines. This type of access uses either Telnet (clear-text) or SSH (encrypted) tools to remotely access the Cisco IOS command line. It is also recommended that you allow access only to certain authorized IP addresses using access lists, just like we did in the web server example. Be sure to apply the access list and restrictions to ALL vty lines. The following is a sample configuration:

```
Switch(config)# access-list 10 permit 10.0.0.1
Switch(config)# line vty 0 4
Switch(config-line)# access-class 10 in
Switch(config-line)# password 0 PrepLogic
Switch(config-line)# login
Switch(config-line)# login-timeout 30 seconds
Switch(config-line)# motd-banner enabled
```

- **Use SSH whenever possible**: Telnet is not secure. Every transmitted character is sent without encryption, in the clear. That is why it is easy for an attacker to intercept packets and get a hold of usernames and password. SSH offers strong encryption and protects from this kind of attack. You should use the highest available version of SSH in the switch. Version 1 and 1.5 are now considered weak with known security flaws. It is recommended that you use version 2 whenever possible.

```
Switch(config)# crypto key generate rsa
Switch(config)# access-list 10 permit 10.0.0.1
Switch(config)# line vty 0 4
Switch(config-line)# access-class 10 in
Switch(config-line)# password 0 PrepLogic
Switch(config-line)# transport input ssh
```

In the first line of this configuration we generate the keys used to provide encryption to the SSH connection.

- **Secure SNMP access** – Read/write access must be disabled in order to prevent unauthorized configuration changes. It is also recommended that SNMP access be restricted to certain known IP addresses. Also remember that SNMP data is transmitted without encryption, in the clear.
- **Secure unused ports** – Unused ports ***must*** be administratively shut down. They should also be configured as access ports to prevent the described VLAN hopping attacks. Their native VLAN must be set to some bogus number too, so that in case a user gains access to the port, ***it*** can be isolated to a bogus VLAN and therefore can't access the network.

  The switchport host macro is useful, because it sets the port as an access port and also uses Portfast in the interface, which also provides certain STP security.
- **Securing the STP Process** – You should always use STP BPDU guard to protect from a bogus switch trying to disrupt the STP process by inserting BPDUs and trying to become the root switch for the STP domain. Remember the BPDU guard feature is configured in portfast interfaces and puts the port in the errdisabled state, which is effectively shut down, if a BPDU is received in the port.
- **Secure the use of CDP** – Cisco Discovery Protocol (CDP) is a very handy feature for network discovery of neighboring devices. It also has other uses in more advanced situations. While information that can be a life saver for network administrators or a requirement for some network equipment, it can also be used by malicious users to craft their attacks on your network platforms. CDP should only be running in ports connecting other trusted network devices, preferably trunk ports. CDP must also be enabled in interfaces connecting Cisco IP phones. Remember you can disable CDP globally using the **no cdp enable** global configuration command or on a per-interface basis with the **no cdp enable** interface configuration command.

VLAN Security

In the past, traffic inspection and security measures where exclusively setup at the router boundaries, where packets could be inspected before being forwarded, but now we can also inspect and apply filtering and other security measures inside our logical networks, VLANs. This is achieved with the use of VLAN Access Control Lists (VACLs).

Cisco Catalyst Switches can also logically divide a VLAN into multiple groups that share the same subnet and default gateway and are able or unable to communicate with each other based in a set criteria. Private VLANs (PVLANs) provide this capability and are a very powerful means to secure and isolate certain users and resources from others without requiring a different logical network (inside the same VLAN). They also allow for more efficient network resource utilization, because broadcasts and multicasts are not transmitted to isolated hosts in the VLAN.

VLAN trunks are used to connect switches carrying different VLANs, and this provides a dangerous and vulnerable point of attack when they are not physically secure. In this domain we will examine some methods to prevent the most common attacks to trunk ports and VLANs.

VACLs: VLAN Access Control Lists are access lists used to filter traffic that doesn't move beyond the VLAN. They are merged into the TCAM just as standard ACLs, which means they are ways to filter traffic with no switching penalty, as they are performed at wire speed, in hardware. VACLs are configured like route maps, meaning they are a series of match statements and then action statements that are the route map equivalent of the set statements.

When configuring a VACL, the first thing you must do is define the VACL with a name. Next you must define one or more matching conditions and then the subsequent action, with an action statement. Finally you must attach the VACL to a specific VLAN, just like you attach regular ACLs to terminal lines, services, interfaces, etc. Here is a sample configuration:

```
Switch(config)# vlan access-map map-name [sequence-number]
Switch(config-access-map)# match ip address {acl-number | acl-name}
Switch(config-access-map)# match ipx address {acl-number | acl-name}
Switch(config-access-map)# match mac address acl-name
Switch(config-access-map)# action {drop | forward [capture] | redirect
type mod/num}
Switch(config)# vlan filter map-name vlan-list vlan-list
```

It is important to understand that VACLs are applied globally to one or more VLANs, not VLAN interfaces (SVIs). These are layer 2 ACLs and SVI's are layer 3 logical router interfaces. Remember that VLAN interfaces are points where data leaves the VLANs, and since VACLs are meant to filter data within the VLANs, it doesn't make sense to apply them to the SVI. VACLs don't have any inbound or outbound direction because they operate at Layer 2.

Private VLANs

Private VLANs (PVLAN) are simply the logical segmentation of regular VLANs. This means that a private VLAN will have hosts in the same subnet that will not receive broadcasts from each other. In some instances the hosts will be able to communicate with each other and in some instances they simply won't.

Think of a datacenter or a server farm. The network is using a single subnet for the farm. Each server should be able to communicate with the router or L3 switch, its default gateway, but it would be very handy and a resource saver if all servers didn't have to listen to each other's broadcasts. That is exactly what private VLANs provide: PVLANs provide Layer 2 isolation between ports within the same broadcast domain, and the security inherent to blocking the access to areas of the network that don't require it. The regular VLAN is divided in several logical groups or segments. The resulting network is the Private VLAN, composed of a primary VLAN and one or more secondary VLANs. Think of the primary VLAN as the regular VLAN before segmentation, and secondary VLANs are the new logical segments that you need to provide security or simply prevent traffic and broadcasts from being transmitted to. Hosts in secondary VLANs can communicate with hosts in the primary VLANs, but cannot communicate with hosts in another secondary VLAN.

The secondary VLANs can be configured as one of the following types:

- **Isolated –** This type of PVLAN can only communicate with hosts in the primary PVLAN. Hosts connected to an isolated secondary PVLAN cannot communicate with hosts in the same secondary PVLAN. Isolated ports are isolated from the rest of the network, except the primary PVLAN.
- **Community –** Any switch port belonging to a common community can communicate with each other and with the primary PVLAN. They cannot communicate with ports/hosts belonging to another PVLAN.

Isolated secondary PVLANs are used frequently in the ISP to isolate clients. One client doesn't have to hear a broadcast from another customer of the ISP, thus making them ideal for this type of configuration.

The Community PVLAN is used in the ISP to provide connectivity between a client's remote sites.

Secondary VLANs must be associated with a primary VLAN. VTP doesn't transmit PVLAN information to switches in the VTP domain. PVLANs are only locally significant to a switch.

There are two types of PVLAN ports:

- **Promiscuous –**A promiscuous port effectively ignores the PVLAN rules, because it can communicate with the primary VLAN and anything connected to it, be it a secondary VLAN, isolated or community.
- **Host –** This switch port connects to a host and is only able to communicate with ports in primary VLAN, its own community VLAN or promiscuous ports.

Private VLAN configuration Steps:

1. Determine how many secondary VLANs are required and which ones are going to be communities and which will be isolated. Also determine if there will be hosts connected to promiscuous ports, that is hosts that will require communication with all VLANs, primary and secondary (communication with all hosts in the subnet, regardless of PVLAN configuration limitations). Create the VLANs and define them (isolated or community):

```
Switch(config)# vlan vlan-id
Switch(config-vlan)# private-vlan {isolated | community}
```

2. Define the Primary VLAN and make the associations with secondary VLANs:

```
Switch(config)# vlan vlan-id
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# private-vlan association {secondary-vlan-
list | add secondary-vlan-list | remove secondary-vlan-list}
```

3. Associate Switch ports with their corresponding PVLAN. First define the port as promiscuous or host and then you must associate host ports with their primary and secondary VLANs. If the port is promiscuous, you need to make a mapping of the primary to secondary VLANs allowed. Here are both possibilities:

```
Switch(config-if)# switchport mode private-vlan {host | promiscuous}
```

Now if you configured a host port, you need to associate the host's primary and secondary VLANs with the following command:

```
Switch(config-if)# switchport private-vlan host-association
primary-vlan-id secondary-vlan-id
```

If the port was configured for promiscuous mode operation with the **switchport mode private-vlan promiscuous interface** configuration command, you need to map the ports to a primary and one or more secondary VLANs. You can also remove or add new secondary VLANs in case they are needed. Use the following command to achieve this:

```
Switch(config-if)# switchport private-vlan mapping primary-vlan-
id secondaryvlan-list | {add secondary-vlan-list} | {remove
secondary-vlan-list}
```

The following is a configuration example:

```
Switch(config)# vlan 10
Switch(config-vlan)# private-vlan community
Switch(config)# vlan 20
Switch(config-vlan)# private-vlan community
Switch(config)# vlan 30
Switch(config-vlan)# private-vlan isolated
Switch(config)# vlan 100
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# private-vlan association 10,20,30
Switch(config-vlan)# exit
Switch(config)# interface range fastethernet 1/1,2
Switchconfig# switchport private-vlan host
Switch(config-if)# switchport private-vlan host-association 100 10
Switch(config)# interface range fastethernet 1/4 - 5
Switchconfig# switchport private-vlan host
Switch(config-if)# switchport private-vlan host-association 100 20
Switch(config)# interface fastethernet 1/3
Switchconfig# switchport private-vlan host
Switch(config-if)# switchport private-vlan host-association 100 30
Switch(config)# interface fastethernet 2/1
Switch(config-if)# switchport mode private-vlan promiscuous
Switch(config-if)# switchport private-vlan mapping 100 10,20,30
```

In the previous configuration, VLANs 10, 20 and 30 are secondary VLANs, with 10 and 20 being communities, which simply means that hosts in the same community will be able to communicate with each other (hosts in VLAN 10 will be able to communicate with each other regardless of their location, and they will be transparent to hosts in any other VLAN inside the PVLAN). Since VLAN 30 is an isolated VLAN, a host that belongs to it will only be able to communicate with the primary VLAN and promiscuous ports.

We then create the VLAN 100 and define it as the primary VLAN, and then associate it with the secondary VLANs that we have already created.

After that we go to the specific ports and define them as host ports and associate them with their corresponding primary and secondary VLANs.

Finally we define the promiscuous ports and ***map*** the primary VLAN and all the secondary VLANs that the port will be able to communicate with.

The terminology can look confusing at first because of the multiple associations and mappings, but it is quite simple. The first thing you must have in mind is that you need to associate secondary VLANs to their primary VLAN. Then you need to define the ports that only require communication with the primary VLAN (and the promiscuous ports) and define them as host ports, while associating them with the proper primary and secondary VLAN. Finally the ports that will effectively be inside the PVLAN but will not abide by its rules, the promiscuous ports, need to be defined, and mapped with those VLANs that they'll be allowed to communicate to.

Private VLANs and Switched Virtual Interfaces

When you have a SVI (a VLAN configured with a Layer 3 address on a multi-layer switch) routing traffic from private VLANs, you are required to map additional private VLANs with the SVI, because the mappings and associations defined so far have been done at the Layer 2 level.

Everything is configured for Layer 2 operation as we described earlier. You are only required to add the mapping for the VLAN interface or SVI, with the following command:

```
Switch(config-if)# private-vlan mapping {secondary-vlan-list | add
secondaryvlan-list | remove secondary-vlan-list}
```

If we assume the secondary VLANs 10 and 20 have been created, the layer 2 associations and mappings defined, and the SVI VLAN 100 is properly configured with its IP address, the required configuration to map the secondary VLANs to the SVI and allow routing is the following:

```
Switch(config)# interface vlan 100
Switch(config-if)# private-vlan mapping 10,20
```

Trunk Security

As we said in the introduction to this domain, it is common for network administrators to grow overconfident about trunk ports because they are usually physically secure. There are several weak points that attackers can exploit to gain unauthorized access to network resources. We will examine these possibilities and see how we can prevent and mitigate these attacks.

Switch Spoofing

The most common exploit of this kind happens when a switch is left to its default DTP status, which is **auto**. In this state the port will wait for another port to start negotiation to form a trunk. If a PC is connected the port will become an access port with access only to the native VLAN. But this opens the possibility of security breach if the attacker emulates or spoofs a switch and starts sending DTP frames and a trunk is formed. In the default setting, all VLANs in the switch are allowed through the trunk, which in essence gives access to the intruder to all networks that cross the switch.

The solution to this problem is to configure the switch port to not send DTP frames, and this is achieved by configuring it as an access port. Trunk ports connecting the access switch to the distribution switch must be configured with a bogus native VLAN that is not in use in the network. Let's assume Fast Ethernet ports from 1 to 24 are left unused. The following configuration puts all ports in VLAN 10 in access mode:

```
Switch(config)# interface range fastethernet 0/1 – 24
Switch(config)# switchport access vlan 5
Switch(config)# switchport mode access
```

In this configuration the Fast Ethernet ports 1 – 24 are defined as access ports and will not negotiate a trunk under any circumstance.

VLAN Hopping

In this attack frames are sent with multiple 802.1Q tags, which make the switch or switches send malicious frames to hosts in different VLANs without the use of a router. The attacker uses double tagging with its own VLAN tag on the outside. When the frame is forwarded out of a trunk, the first tag is stripped off and the switch on the other end receives the frame with the 802.1Q tag for the VLAN the attacker intended to reach.

Several conditions must be met in the network and switch configuration in order for this attack to be possible. The mitigation and prevention of this type of attack is given by preventing this situation.

First, the attacker must be connected to a switch port and the trunk uplink must be an 802.1Q trunk with its native VLAN being the same as the one used by the attackers access port. So if the native VLAN is 10, the attacker must also be connected to a switch port configured for VLAN 10.

As we explained, the first tag given to the frame is the one with the VLAN the attacker wants to reach. Then a second tag uses the attacker's VLAN, the native VLAN. When the switch receives this frame, it realizes the frame is tagged with its native VLAN, and when it forwards the frame out the trunk it strips off the first tag and assumes the frame is going out untagged, as it should be, to the trunk, when in reality the spoofed tag is left exposed to the trunk and will be received by the other trunk port in the other switch. If the switch has a host in this VLAN it will forward the frame out the corresponding port and hosts will receive the frames as if they were from legit users.

The solution to this type of attacks is to use to set the native VLAN in trunk ports to an unused VLAN. You should also prune the native VLAN from the trunk, which confines a possible attacker (who finds out about the unused VLAN) to the trunk link.

A second method to prevent the double tagging VLAN Hopping attack is to force the switch to tag frames destined to the native VLAN. As you know, the native VLAN frames are untagged by default. You can change this behavior and make the switch tag frames for the native VLAN, which renders the double tagging attack useless, as the switch will put the frame in the trunk exactly as it was sent by the attacker, with the native VLAN tag on the outside. To make the switch tag native VLAN traffic, use the following command:

```
Switch(config)# vlan dot1q tag native
```

Domain 3: Implementing Switch-based Layer 3 Services InterVLAN Routing

In the past, a router was required to provide layer 3 connectivity. The router needed to have one physical or logical interface connected to the subnets that it had to provide communication to. Now this Layer 3 communication can be performed within the LAN by a multilayer Switch.

When the router or multilayer switch connects to a trunk port in the switch and has multiple logical interfaces, the interVLAN routing is often called "router on a stick" because it has a single branch from the multi-VLAN switch to the router that has multiple logical layer 3 interfaces configured on a single physical port. All VLANs go through the trunk and the router/Layer 3 switch is configured with subinterfaces that are meant to receive traffic from the different VLANs/subnets.

As we explained in the introduction in domain 1, a multilayer switch can perform the function of communicating with hosts in different VLANs, routing, in hardware, with the **Application Specific Integrated Circuits (ASICs)** at wire speed, just as if it were performing Layer 2 switching.

A multilayer switch can forward traffic from Layer 2 or Layer 3. Layer 3 forwarding can be implemented assigning a physical interface or a logical interface named switched virtual interface (SVI). When an interface is given a Layer 3 address, an IP address, it becomes the default gateway of any host connected to the interface or VLAN.

InterVLAN routing requires that you define the switch interface as a Layer 3 interface. By default, switch ports are configured to be Layer 2 interfaces. You configure ports for Layer 3 operation with the **no switchport** interface configuration command. Equally, if you need to change a layer 3 port to operate as a Layer 2 interface, use the **switchport** interface configuration command. Not all Cisco Catalysts switches are multilayer switches.

You can display the current operation mode of an interface with the show interface type mod/num switchport EXEC command. The output of the command is the following:

```
Switch# show interface fastethernet 0/1 switchport
Name: Fa0/1
Switchport: Disabled
Switch#
```

Figure 22: Verifying a Switchport's Operation Mode

Switchport refers to Layer 2. If you see **Switchport: enabled**, this means the interface is operating at Layer 2, the default. In this case, the **no switchport** command was used and the port is a Layer 3 interface, that must be configured with an IP address and network mask.

Since Layer 2 operation is transparent to the user (no configuration such as IP address or routing protocol is necessary in order to make the port operational), no additional configuration is required to connect hosts or another switch to the port. On the other hand, if you set the port as a Layer 3 configuration, the following is the minimum configuration required in order for the port to be able to communicate with hosts:

```
Switch(config)# interface type mod/num
Switch(config-if)# no switchport
Switch(config-if)# ip address ip-address mask [secondary]
```

If several ports are part of an EtherChannel, and it is configured as a Layer 3 interface, the IP address must be configured under the port-channel interface. You should never give an individual port an IP address if it is part of a bundled, EtherChannel. For instructions on how to set an IP address to an EtherChannel refer to domain 1.

SVI Ports

As we mentioned earlier, you can give Layer 3 forwarding functionality to a VLAN with a MLS. For this you must assign an IP address to the VLAN. This is especially useful when you have several ports under the same VLAN and routing is necessary out of the VLAN. Instead of requiring several interfaces with their own default gateways or requiring an additional router or Layer 3 device, you use the SVI as your point of entry and exit in and out of the VLAN for all hosts.

In the following example we will create the VLAN 10 and will name it Sales. We will then define it as a SVI by assigning an IP address to it:

```
SwitchA(config)# vlan 10
SwitchA(config-vlan)# name Sales
SwitchA(config-vlan)# exit
SwitchA(config)# interface vlan 10
SwitchA(config-if)# ip address 192.168.1.1 255.255.255.0
SwitchA(config-if)# no shutdown
```

Now, to better understand the function of a SVI, suppose you have 8 ports using VLAN 10, Sales in the Switch A. If the SVI didn't exist, we would require an additional router or Layer 3 switch connected to a trunk port of Switch A, in order to route traffic in or out of the subnet assigned to VLAN 10. The SVI allows the switch to forward all Layer 3 traffic outside the IP subnet within the VLAN by configuring the hosts with the SVI as their default gateways.

Multilayer Switching: Cisco Express Forwarding

Cisco Express Forwarding (CEF) is Cisco's proprietary method of layer 3 packet forwarding. It is a route cache switching method. CEF provides wire speed performance with the use of dynamic lookup tables that are kept in hardware and Application Specific integrated Circuits (ASICs).

CEF performs packet switching with the use of two functional blocks: The Layer 3 Engine and the Layer 3 Forwarding Engine. The Layer 3 Engine acts as a router keeping the routing information based in information manually configured or learned from dynamic routing protocols. The Layer 3 Forwarding Engine is used to forward packets *in hardware* to the destinations learned and kept by the Layer 3 Engine.

The **Forwarding Information Base** (FIB) is basically the new routing table as we understand from traditional routers, with a new format and with a couple minor differences. The FIB contains an ordered list of IP destinations with the most specific, longest-prefix address first and its associated next hop IP address. The most specific possible route is the host route, or routes with the 255.255.255.255 network mask. That means the FIB knows the exact route to the destination. These are present in the FIB, and differently from the traditional routing table, they don't have to be manually configured. They are used for directly connected routes. This mechanism provides for higher efficiency in the table lookup and forwarding process.

The FIB receives the routes from the Layer 3 Engine, which is the one running the routing protocols. Changes made to the IP routing table or ARP Table in the Layer 3 Engine must be immediately reflected in the FIB, because it is the one that provides the packet forwarding at wire speed and the preferred functional block for this task, whenever possible.

You can display the contents of the FIB table entries to a specific VLAN or interface with the **show ip cef** [*type mod/num* | **vlan** *vlan-id*] [**detail**] EXEC command. Here is a sample output:

```
Switch# show ip cef vlan 10
Prefix          Next Hop        Interface
192.168.1.0/24  Attached        Vlan 10
192.168.1.10/32 192.168.1.10    Vlan 10
192.168.1.15/32 192.168.1.15    Vlan 10

Switch#
```

Figure 23: Displaying the Contents of the FIB Table

You can also view FIB entries for specific IP addresses and network masks with the following command:

```
Switch# show ip cef [prefix-ip prefix-mask] [longer-prefixes] [detail]
```

Can you change configuration settings that alter the FIB? Yes, but it is outside the scope of the SWITCH exam, as is the mechanism used to add certain and specific IP routes to the FIB. There are situations where some entries are not processed by the ASICs, and are handled by the CPU for conditions that are explained later.

The **longest-prefixes** optional parameter within the show command is used to display longest match entries in the FIB table. The **detail** optional parameter is used to display additional route information, such as the version number, epoch and other information. The *version* number shows the amount of times the router has been updated since the FIB was created. The *epoch* number is the number of times the FIB table has been cleared and built again.

There are certain instances when packets cannot be forwarded by the Layer 3 Forwarding Engine with the Application Specific Integrated Circuits that provide "wire speed". In these cases, the packet is marked as "CEF punt" and is sent (or punted) to the Layer 3 Engine to be processed and routed using traditional software-based processor-intensive processing. Some of the most common conditions that can cause the multilayer switch to mark a packet as CEF-punt include:

- An entry for the destination network cannot be found in the FIB table.
- The FIB table is full. The remedy has more to do with proper network design and an IP addressing scheme (contiguous networks that allow proper route summarization and such things), that allows a more efficient dynamic routing and even the addition of default routes to big network segments.
- The packet has to be fragmented because the MTU has been exceeded.
- The IP Time-To-Live (TTL) has expired.
- The encapsulation type is not supported. The only Ethernet encapsulation supported by hardware switching is ARPA.
- The packet has been subject of security mechanisms such as tunneling, encryption, compression or has triggered the log option in a local access list.
- The packet is destined out an interface configured with the ip nat outside interface configuration command, which means a Network Address Translation (NAT) operation is required. Only a few high-end MLSs can provide NAT operation in hardware.
- In short, a packet is marked as CEF-punt when its route to the destination is not found in the FIB table or when the packet requires some special handling or features that are not currently supported or performed in hardware at wire speed.

So far we've discussed the scenario where the FIB is maintained completely in one MLS platform. There is the possibility of splitting the load among several switches in order to improve efficiency when forwarding packets. Specialized hardware is required for this and there are two CEF methods that allow this behavior:

1. **Accelerated CEF (aCEF) –** In this mode CEF is distributed in multiple Layer 3 Forwarding Engines, currently in the form of line cards. Only a portion of the FIB is downloaded to the engine because they generally don't have the capability to receive and hold the entire table. The routes that are kept in the "partial" FIB tables are the ones that are more likely to be used again. If a route is not found a request is made to the Layer 3 Engine and the FIB table is updated. As you can see, the resulting operation is very fast forwarding but not *necessarily* at wire speed.

2. **Distributed CEF (dCEF)** – In this method the CEF is fully distributed among several Layer 3 Forwarding Engines. This provides for considerably increased performance. Since the whole FIB table is replicated, there can be as many Layer 3 Forwarding Engines as necessary. A central Layer 3 Engine is used with this method to maintain a routing table and populate the FIB tables in Layer 3 Forwarding Engines in the line cards.

Adjacency Table

The CEF adjacency table is the Layer 3 Forwarding Engine equivalent of the ARP table in a standard router. It's the part of the FIB table where the Layer 3 to Layer 2 mappings are kept. This is specifically for each Layer 3 next hop address, which can always be accessed at Layer 2.

To display the adjacency table contents use the following command:

```
Switch# show adjacency [type mod/num | vlan vlan-id] [summary | detail]
```

You can view the adjacencies in a certain interface or a VLAN. You can also display the number of adjacencies stored of both physical and VLAN interfaces using the **summary** optional keyword as follows:

```
Switch# show adjacency summary
```

In addition, you can see a lot more information with the **detail** keyword. The following example shows an actual output:

```
Switch# show adjacency vlan 10 detail
Protocol    Interface     Address
IP          Vlan 10       192.168.1.10(5)
                          7 packets, 672 bytes
                          000CEE45B2F9000F444E64220800
                          ARP 03:54:55
                          Epoch: 0

IP          Vlan 10       192.168.1.15(5)
                          4 packets, 416 bytes
                          000CE44FF4C8333932EEa1210800
                          ARP 01:01:25
                          Epoch: 0

Switch#
```

Figure 24: Showing Interface Adjacencies

As you can see, there are both Layer 3 and Layer 2 addresses in that output of this show command. The MAC addresses are the equivalent of the first 3 octets in the long hex string below each IP address. The remainder of the string is the hex values that are a combination of the device IP address and EtherType value.

As we mentioned, the adjacency table is built from the ARP table of the Layer 3 Engine. If the FIB table doesn't have an ARP associated to an IP address entry, the switch can't forward the packet at wire speed, and the packet is sent to and processed by the Layer 3 Engine. When the Layer 3 Forwarding Engine can't forward frames because the ARP entry doesn't exist in the adjacency table, the FIB entry is marked as "CEF Glean". The packet is sent to the Layer 3 Engine which will send an ARP request. After receiving an ARP reply it will be able to forward the packet.

While in the CEF glean state, and in order to prevent the Layer 3 Engine from being overwhelmed with ARP request duplicates, the multilayer switch drops all packets destined for entry in which adjacency is in the glean state. This feature is called **ARP throttling or throttling adjacency**. The ARP throttling makes the Multi-layer switch wait for an ARP reply for 2 seconds before sending a new ARP request. When an ARP Reply is received, the ARP throttling is released and ARP requests are sent when needed. The FIB table is updated and the Layer 3 packet forwarding with the Layer 3 Forwarding Engine, at wire speed, is resumed.

There are several types of adjacency possible in the adjacency table. They are helpful to the forwarding process. Here are some of the more important types:

- **Null adjacency –** Used for the routes pointing to the null interface.
- **Drop adjacency –** Used for packets meant to be dropped by the switch, due to an unsupported protocol, an encapsulation failure, unresolved address, or a checksum error among others.
- **Discard adjacency –** Used when packets must be discarded because of an access list deny match or other policy.
- **Punt adjacency –** Used when the packet cannot be forwarded with the Layer 3 Forwarding engine and must be punted to the Layer 3 Engine.

You can analyze the CEF punt activity with the **show cef not-cef-switched** EXEC command. You will find the reasons of the **cef punt** condition.

The possible CEF punt reasons are the following:

- **No_adj –** An incomplete adjacency.
- **No_encap –** An incomplete ARP resolution.
- **Unsupp'ted –** Unsupported packet features.
- **Redirect –** ICMP redirect.
- **Received –** Packets destined for Layer 3 Engine interfaces.
- **Options –** IP options that cannot be analyzed in hardware present.
- **Access –** Access list evaluation failure.
- **Frag –** Fragmentation failure.

After the FIB entry has been found in the table, one final step is taken before actually forwarding the packet. The Layer 2 frame header contains the MAC address of the receiving switch interface, and it must be rewritten as the MAC address found for the destination entry in the adjacency table. This is the same thing that a standard layer 2 switch would do when forwarding frames. Remember that during the transport of a packet, the IP addresses never change while traversing the network but the layer 2 MAC addresses are constantly being rewritten to indicate the next source and destination MAC address along the path. The same happens with the source address that has to become the MAC address of the port through which the packet is going to be forwarded. After the frame addresses are changed, the checksum needs to be recalculated, and the same happens at Layer 3, because the TTL value needs to be decreased by one, and the checksum needs to be recalculated accordingly.

The ***packet rewrite*** is done very efficiently with the used of specialized hardware, the application Specific Integrated Circuits (ASICs).

Configuring CEF

CEF is enabled by default in all CEF capable Catalyst Switches. In some Catalyst Switches, the CEF runs with the IOS and can never be disabled. You can disable CEF on a per-interface basis on switches that allow it, with the no **ip route-cache cef** or **no ip cef** interface configuration commands, depending on the Catalyst switch model.

The first thing to do is verify that the port is configured as a Layer 3 port with the following command:

```
Switch# show interface type mod/num switchport
```

Remember, when you see "switchport" think "Layer 2". In this case, disabled means the port is disabled for Layer 2 switching, and enabled for Layer 3 switching.

If the port is configured for Layer 2 operation, the command will display VLAN information, if it is an access or trunking port, trunking mode and native VLAN.

To verify the configuration in an SVI, use the following command:

```
Switch# show interface vlan vlan-id
```

If there is no output that means the VLAN interface is shutdown or the VLAN hasn't been created in the switch.

Use the **show vlan** EXEC command to view all configured VLANs. You will see all the VLAN names and the ports associated to each VLAN.

To display information regarding the IP configuration of an interface use the **show ip interface** command.

Here is a sample output:

```
Switch# show ip interface vlan 10
Vlan10 is up, line protocol is up
Internet address is 192.168.1.1/24
Broadcast address is 255.255.255.255
Address determined by setup command
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Outgoing access list is not set
Inbound access list is not set
Proxy ARP is enabled
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
ICMP redirects are always sent
ICMP unreachables are always sent
ICMP mask replies are never sent
IP fast switching is enabled
IP fast switching on the same interface is disabled
IP Flow switching is disabled
IP CEF switching is enabled
IP Feature Fast switching turbo vector
IP Feature CEF switching turbo vector
IP multicast fast switching is enabled
IP multicast distributed fast switching is disabled
IP route-cache flags are Fast, Distributed, CEF
Router Discovery is disabled
IP output packet accounting is disabled
IP access violation accounting is disabled
TCP/IP header compression is disabled
RTP/IP header compression is disabled
Probe proxy name replies are disabled
Policy routing is disabled
Network address translation is disabled
WCCP Redirect outbound is disabled
WCCP Redirect inbound is disabled
WCCP Redirect exclude is disabled
BGP Policy Mapping is disabled
Sampled Netflow is disabled
IP multicast multilayer switching is disabled

Switch#
```

Figure 25: Output from the show ip interface Command

You can also use the **show ip interface brief** EXEC command to display some information about Layer 3 interfaces. This is a very well-known command from the CCNA studies. It displays the physical interfaces and SVIs IP addresses and their operational status at both Layer 1 and Layer 2.

To verify CEF operation use the **show ip cef** EXEC configuration command. The following is a sample output:

```
Switch# show ip cef
Prefix              Next Hop        Interface
0.0.0.0/32          receive
192.168.1.0/24      attached        Vlan10
192.168.2.0/32      receive
192.168.2.1/32      receive
192.168.1.2/32      192.168.199.2   Vlan1
192.168.199.255/32  receive
Switch#
```

Figure 26: Verifying CEF Operation

The entries marked with the next hop "receive" are the ones that are directly connected and will be handled by the Layer 3 Engine (CPU, not wire speed). The ones "attached" are those packets that must be routed through an SVI. You usually need to check this information if you are having performance issues and you think the MLS is not routing at wire speed. You can confirm this by checking the FIB table and/or by checking the previous output.

You can also display the FIB table contents based on the interface, with the following command:

```
Switch# show ip cef type mod/num [detail]
```

Using DHCP with a Multilayer Switch

When a port of a multilayer switch is configured as a Layer 3 interface, hosts connected to it should be configured with the IP address of the interface as their default gateway. Hosts can be manually configured, but that is very inefficient from a network management perspective as the network grows. That is when the Dynamic Host Configuration Protocol (DHCP) plays a role. As we learned in the CCNA studies, DHCP is a service where a remote server assigns IP addresses, default gateway and DNS servers' IP addresses to hosts requesting the configuration with a DHCP Discover message. Other parameters can also be sent to the requesting device. When a host needs an IP address it attempts to contact a DHCP server and the procedure is as follows:

1. **The client sends a DHCP Discover broadcast message**. The client sends the broadcast at Layer 2, using its MAC address as the source address and since the message is a broadcast, ffff.ffff.ffff is the destination address.

2. **A DHCP server present in the subnet responds with a "DHCP Offer" message**: This offer message contains an IP address, subnet mask, default gateway, DNS server or servers and other parameters. The DHCP server also sends its own IP address to identify itself because there could be more than one DHCP server in the subnet. Since the client doesn't have an IP address, the DHCP offer is sent as a broadcast.

3. **The client sends a "DHCP Request":** In this message the client is accepting the parameters for configuration that the DHCP server sent in the DHCP Offer message. This message is also sent as a broadcast because the client still doesn't have a valid IP address.

4. Finally, **the DHCP server replies with a DHCP ACK message**: The offered IP address and parameters in the DHCP Offer message are sent again as a confirmation that they are available and approved for the host to use in the subnet. This message is also sent as a broadcast.

DHCP servers were originally designed to operate in the same broadcast domain (subnet/VLAN) of the hosts they were meant to serve. Now you can configure DHCP Relay agents on layer 3 devices such as a multilayer switch. You can then configure one (or a pair for redundancy) centrally located DHCP server for dynamic configuration of hosts in several subnetworks.

Without the use of the DHCP Relay agent, DHCP servers would only be able to provide configuration parameters inside their broadcast domain.

A router or a multilayer switch can also be used as a DHCP Server. The configuration is as follows:

```
Switch(config)# ip dhcp excluded-address start-ip end-ip
Switch(config)# ip dhcp pool pool-name
Switch(config-dhcp)# network ip-address subnet-mask
Switch(config-dhcp)# default-router ip-address [ip-address2] [ip-address3] ...
Switch(config-dhcp)# lease {infinite | {days [hours [minutes]]}}
Switch(config-dhcp)# exit
```

We first configure the excluded range of addresses. These are the IP addresses that will be used in hosts that require a static IP address in the subnet or if you simply want to reserve a group of addresses on the subnet that you don't wish to hand out to end devices. Network devices such as switches, routers, access points and servers all require static IP addresses. We define and name a DHCP pool and in DHCP configuration mode define the network we will be granting IP addresses from, the default gateway, and lease time.

The lease time if not defined is 1 day by default. Hosts negotiate their IP addresses in about half the lease expiration time and decide if they will keep the leased IP address.

DHCP Relay Agent

As was explained earlier, most production DHCP server deployments are centralized for all or most subnets where clients require DHCP services. As we explained, DHCP messages are sent as broadcasts, which mean they are contained by router or Layer 3 MLS interfaces and are kept inside the VLAN. The DHCP Relay Agent allows us to place the DHCP server in a centralized place in the network and provide DHCP parameters to hosts in more than one IP subnet.

The DHCP relay agent listens to the DHCP Discover broadcast from clients, intercepts it and then forwards a packet as a unicast to the DHCP server. There is a field where the router adds the IP address of the interface that received the DHCP broadcast. That is how the DHCP server knows from what IP subnet (DHCP pool) it should send the DHCP offer. The response is obviously sent back to the relay agent and it forwards the offer to the host that issued the DHCP Discover message.

To configure a MLS as a DHCP Relay agent use the **ip helper-address** interface configuration command in the Layer 3 interface connected to the IP subnet that hosts the endpoints that require DHCP configuration. This will either be a router interface or a logical layer 3 VLAN interface on a multilayer switch. You can use the **ip helper-address** command as much as you want, and this causes the router or MLS acting as a DHCP relay agent to forward the DHCP request to all configured addresses under the command. All DHCP servers will reply to the relay agent and it will forward all DHCP offers to the host, which will have to decide which one to accept and use.

The following is a sample configuration of a DHCP relay agent:

```
Switch(config)# interface vlan 10
Switch(config-if)# ip address 192.168.1.1 255.255.255.0
Switch(config-if)# ip helper-address 10.0.1.5
Switch(config-if)# exit
```

In this configuration the SVI acts as the default gateway of hosts connected to ports that belong to VLAN 10. A DHCP server must be properly configured with the address 10.0.1.5.

Domain 4: Preparing the Infrastructure to Support Advanced Services

Today's networks not only carry data as they used to; there are now more requirements as campus networks carry a great deal of voice and video communications. The switched campus network must be designed and properly configured to handle the current demands for voice, video and wireless traffic. With VoIP – IP telephony -- the main consideration involves network congestion avoidance with a set of techniques known as Quality of Service (QoS). VoIP traffic cannot compete with data for available bandwidth because of the nature of live voice communications. There are also security considerations that must be addressed. In this domain we will explore the options and features in the Cisco Catalyst switches that allow our switched networks to operate efficiently carrying and delivering data, voice and video.

Voice over IP (VoIP) – IP Telephony

Cisco Telephony devices, IP phones and the like, require power to function. Most Cisco IP phones can be plugged directly into a wall's power outlet. Many times this is not the best solution as it requires every phone to have an available AC power outlet nearby. A better alternative is Power over Ethernet (PoE). **PoE** is how low voltage electricity is sent to the IP phones and other network devices over standard Ethernet cables.

PoE can also resolve the problem of VoIP phones being out during a power outage. Now with PoE there is a centralized point, the wiring closet where the IP phones connect to the access layer switch, to offer a backup in case of electricity failures. Instead of requiring a UPS in every single point where an IP phone is present, you can guarantee that all IP phones will remain operational in cases of a power outage by providing a backup to the access layer switch.

The PoE also saves money because an additional AC adapter is not required.

Power over Ethernet is available in two methods in Cisco Catalyst switches:

- **Cisco Inline Power (ILP)** – The Cisco-proprietary method that was the first method of delivering power to network devices through data cables.
- **IEEE 802.3af –** The IEEE standard allows Cisco switches to provide power over Ethernet to non-Cisco network devices.

Some older Cisco phones and wireless access points only operate using ILP. Therefore Cisco switch ports can detect either ILP-only PoE devices or ones that are capable of 802.3af. In addition, some older switches only offer ILP power. If that is the case, they may not be able to power some 802.3af devices. You will need to do some research to make sure your PoE switch operates with the PoE end device. Fortunately, ILP is quickly becoming a thing of the past and most switches and end devices in production today are 802.3af capable.

Cisco switches don't offer power constantly to the line. They try to detect if there is a connected device that requires power using two different methods, for each of the PoE methods.

When using the IEEE 802.3af standard, the switch applies a small voltage to both sending and receiving twisted pair cables, and if it measures a 25K ohm in the line, it means an IP phone is connected and then proceeds to apply power. The IEEE 802.3af defines 4 power classes, and there is a new one that newer Cisco Switches can identify and use, defined in the IEEE 802.3at standard. Depending on the resistance measured, the Cisco switch applies the proper power to the network device, in this case an IP phone. Remember that this can also be a wireless access point (WAP), a video surveillance system or any other network device compatible with the 802.3af.

The following is a table of the 802.3af power classes:

Power Class	Maximum Power Offered at 48V DC	Notes
0	15.4 W	Default class
1	4.0 W	Optional class
2	7.0 W	Optional class
3	15.4 W	Optional class
4	Up to 50 W	Optional class (802.3at)

Figure 27: 802.3af Power Classes

For Cisco Inline Power (ILP) the switch sends a tone pulse called Fast Link Pulse (FLP) and only a Cisco pre-standard, ILP capable IP phone will be able to loop back the tone. When the switch receives the tone back, it knows an ILP capable IP phone is connected and then applies a very small amount of power (6.3W) to the line. The IP phone powers up and ask for its correct power requirements using Cisco Discovery Protocol (CDP) messages. If CDP is turned off in the switch or the interface, the maximum power (15.4W) is applied to the port.

You must consider PoE when designing your campus network because IP phones will only use the amount of power they need and the remaining will be lost. Most switches cannot provide the maximum amount of power to all their ports. In case the maximum power output is reached, the switch will only power the lower ports until it has no more power. This is called oversubscription.

Using a hypothetical example, if a switch has 24 ports and is capable of providing 100W, and has IP phones connected from ports 0-9, and CDP is turned off in the switch, 15.4W are going to be applied to every port, which means we will see an oversubscription case. Since only 6 ports will be able to be powered with 15.4W, only ports 0-5 will be powered while others will remain without power. Remember oversubscription issues and its possibilities when designing the access layer switches and features. Cisco recommends as a best practice that you manually configure the amount of power supplied through to the port.

PoE Configuration

By default, Cisco Switch ports automatically detect if PoE devices are connected and automatically detect their power requirements. You can configure switch ports to never provide power to connected devices or to provide a fixed amount of power. To do that, use the following interface configuration command:

```
Switch(config-if)# power inline {auto [max milli-watts] | static [max
milli-watts] | never}
```

Theoretically it is possible that a malicious user spoofs an IP phone or any other PoE device and requires the maximum amount of power, taking away unnecessary resources that could lead to depletion and to a form of denial of service attack. This is why it is recommended that you manually configure the amount of power supplied to the port whenever possible. Many specialists use that as a general rule in the Cisco world: never use an "auto" option if there is another feasible option.

Auto is the default option. You can use the **static** keyword to use a fixed or "static" amount of power regardless of what the IP (or any other connected PoE device) tries to obtain. You can also use the **max** keyword to define the maximum amount of power to be provided through the port.

To disable PoE, use the **never** keyword.

To display the power over Ethernet status of a switch use the show power inline EXEC command.

```
Switch# show power inline
Module      Available   Used        Remaining
            (Watts)     (Watts)     (Watts)
--------    ----------  ---------   -----------------
1           369.0       30.8        338.2

Interface   Admin   Oper    Power   Device          Class   Max
                            (Watts)
---------   -----   ----    -------  --------       -----   -------
Fa1/0/1     auto    Off     0       n/a             n/a     15.4
Fa1/0/2     auto    Off     0       n/a             n/a     15.4
Fa1/0/3     auto    Off     0       n/a             n/a     15.4
Fa1/0/4     static  On      15.4    IP Phone 7940   n/a     15.4
Fa1/0/5     static  On      15.4    IP Phone 7940   n/a     15.4
Fa1/0/6     auto    Off     0       n/a             n/a     15.4
[output omitted]
```

Figure 28: Verifying PoE

To display the PoE status of a specific port, use the interface type mod/num optional parameter. The command to display, for instance, the PoE of the Fast Ethernet 0/5 is **show power inline fastethernet 0/5.**

Voice VLANs

When an IP phone is connected to an access layer switch, the data stream coming from a connected PC (from the IP phone Ethernet port) and the voice stream from the phone can be configured to use the same or separate VLANs. The VLAN where the voice stream is assigned is called Voice VLAN. The main function of the voice VLAN is to allow network devices like switches and routers to classify traffic according to certain parameters, offering security and the ability to prioritize voice traffic above data traffic, through the use of Quality of Service traffic engineering.

The security aspect provided with the voice VLAN is given by effectively separating data traffic from voice traffic, which in essence makes it impossible for attackers to intercept and capture voice traffic when they gain access to the data VLAN, be it by accessing an authorized endpoint or by accessing a physically insecure access layer switch port.

If the voice VLAN is not used, both voice and data traffic will be in the native VLAN and Quality of Service features won't be used. This can result in voice quality issues, because all traffic will be competing for both switch and/or router limited resources and also limited bandwidth.

Cisco IP phones usually have one or more ports that you can connect a user PC to, giving access to the upstream switch. They can handle traffic from two VLANs, the voice VLAN and the data VLAN. The connection from the Cisco IP phone to the switch can be configured as an 802.1Q trunk, an access port, and now as something very useful and secure sometimes referenced as a "quasi-access port" or "minitrunk". The quasi-access port or minitrunk offers the added benefit of increased security. The minitrunk configuration is as follows:

```
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# switchport voice vlan 20
```

You configure the switch port as an access port with the **switchport mode access** interface configuration command, and then assign the port to the access VLAN 10, the data VLAN. You then specify and allow the voice VLAN 20 in the port. All used VLANs must be previously created.

This increases security because before a trunk had to be configured in the uplink to the switch, and a user only had to use the cable connecting to the IP phone to gain access to the trunk port, which could result in a VLAN hopping attack. Strangely, Cisco doesn't mention this recent feature in the official material for the SWITCH 642-813 exam, and probably you shouldn't expect the concept or configuration in the exam, but we consider the information way too important to not mention. This is the way you must configure ports connecting to IP phones in production networks, for the optimal security architecture and design.

For the exam, we have four scenarios made possible with the use of the following command:

```
Switch(config-if)# switchport voice vlan {vlan-id | dot1p | untagged | none}
```

1. **Switchport voice vlan vlan-id** (VVID) – Here, both data and voice will use their own VLAN. Data traffic will use the native VLAN, untagged frames. Traffic will be separated and QoS features (Class of Sevice CoS bits in the 802.1p encapsulation) can be implemented. The voice VLAN vvid will be tagged. The special-trunk case is created in this scenario if the switch port is configured as an access port. Only two VLANs will be allowed to cross this special trunk, the vvid and native VLAN frames. A voice VLAN has to be previously created. QoS is possible (tagged with Layer 2 CoS priority value).
2. **Switchport voice vlan dot1p** – In this scenario, the special trunk is also created. The voice VLAN will be the VLAN 0. No previous VLAN must be created because voice frames will be sent in the VLAN 0 and data frames will be sent in the native VLAN. QoS can be used (tagged with Layer 2 CoS priority value).
3. **Switchport voice vlan untagged –** Just like dot1p encapsulation, the special case trunk is created, but no voice VLAN is used. All traffic is sent using the native VLAN. No QoS possible (No layer CoS priority value).
4. **Switchport voice vlan none –** Here, no special trunk is created. Traffic is undifferentiated between voice and data. Both use the native VLAN and no QoS Layer 2 CoS tagging or any other tagging is possible.

Cisco switches instruct Cisco IP phones of the operation and encapsulation mode through CDP messages. This means that in order to properly connect and use an IP phone you must make sure that CDP is enabled globally and that the specific interface does not have CDP disabled. By default Cisco switches run CDP in all interfaces. There is a mention of the security threats and most common mitigation methods available in Cisco switches in the Security Domain of this guide.

To verify the voice VLAN operation, use the **show interface switchport** EXEC command:

```
Switch# show interfaces fastEthernet 0/5 switchport
Name: Fa0/5
Switchport:    Enabled
Administrative Mode:    dynamic auto
Operational Mode:    static access
Administrative Trunking Encapsulation:    negotiate
Operational Trunking Encapsulation:    native
Negotiation of Trunking:    On
Access Mode VLAN:    10 (VLAN0010)
Trunking Native Mode VLAN:    1 (default)
Administrative Native VLAN tagging:    enabled
Voice VLAN:    20 (VoIP)
Administrative private-vlan host-association:    none
Administrative private-vlan mapping:    none
Administrative private-vlan trunk native VLAN:    none
Administrative private-vlan trunk Native VLAN tagging:    enabled
Administrative private-vlan trunk encapsulation:    dot1q
Administrative private-vlan trunk normal VLANs:    none
Administrative private-vlan trunk private VLANs:    none
Operational private-vlan:    none
Trunking VLANs Enabled:    ALL
Pruning VLANs Enabled:    2-1001
Capture Mode Disabled
Capture VLANs Allowed:    ALL
Protected:    false
Unknown unicast blocked:    disabled
Unknown multicast blocked:    disabled
Appliance trust:    none
Switch#
```

Figure 29: Verifying Voice VLAN Operation

Quality of Service (QOS)

Quality of Service is the method used to prioritize time-sensitive traffic above less important traffic. When network resources are plenty and bandwidth is huge, there is practically no need to use these methods of traffic engineering, because switches can forward frames as soon as they are received. Essentially, there is no queuing performed so QoS will never be used. Even packets at Layer 3 can be forwarded at wire speed with MLS, providing an outstanding level of performance.

The problems start when network resources, like bandwidth and switch capabilities start to suffer because of network congestion. The packets are too many for the switch to forward and some of them need to enter a "waiting list" or queue before being forwarded. Quality of Service forces network designers and administrators to decide what type of traffic to prioritize based on some established criteria. Different applications have different requirements. Voice and video traffic require the most prompt delivery possible because any variation in delay or large amounts of packet loss can cause the quality of the communication to suffer to the point that communication can fail. On the other hand, an FTP download can have some delay without the user even noticing it. Therefore, FTP would have a lower QoS priority compared to voice/video when network administrators prioritize their data.

When a packet is being forwarded by a switch, QoS can help with three common packet queuing problems that can appear:

- **Delay** – The amount of time the delivery of a packet takes from the source to the destination within a network. The total amount of time it takes a packet to be transported from the source to the destination is called latency. The causes of delay are the amount of time a router or switch spends in table lookups, processing, and the amount of time it takes for the packet to be transported over physical medium in the form of light (fiber-optic connections) or electricity pulses (Ethernet and copper WAN connections such as a T1 or DS3).
- **Jitter** – The variation of delay of multiple packets from the same source to the same destination. Some types of communications require streams of data, and the quality of the communication is heavily dependent on the order of this stream. If the delay variation, jitter, is too big, the stream cannot be reconstructed in the destination. The IP services that are most susceptible to jitter are video and audio streams.
- **Packet Loss –** Sometimes network congestion can cause packets to be dropped without being forwarded. When the application uses connection-oriented, reliable protocols such as TCP, some loss is acceptable because such protocols use retransmission. But when unreliable, best effort delivery protocols like UDP are used, packet loss results in data loss. Loss is also especially unacceptable in video and audio communications. Since voice and video packet streams are transported in real-time, they cannot use TCP's retransmission function because the resent packet will be out of order and too late to be useful. Therefore, most voice and video communications are sent using UDP.

To mitigate these issues, network administrators have the following types of Quality of Service (QoS):

- **Best-effort delivery –** No priorities are set, making QoS effectively inoperative. Switches and routers in running this QoS type simply make a best effort to deliver packets, without establishing any priorities.
- **Integrated services model (IntServ) –** With this QoS, a path is prearranged for the priority data. The path is from end-to-end, from the source to destination. The Resource Reservation Protocol (RSVP) is the mechanism that schedules and reserves proper path bandwidth for the required application. The source application requests QoS parameters through RSVP. Each network device must check to see if it can meet the minimum requirements, and when the complete path is checked and approved, the source application is signaled with confirmation that it can transmit.
- **Differentiated services model (DiffServ)** – This method was developed to address the limitations of the integrated services model. The limitation was basically scalability. When several applications requiring QoS, using the IntServ model bandwidth was reserved in every network device along the path to the destination. As you can see, it is likely that as the demand for QoS grows, network devices start to leave practically no resources to regular traffic. DiffServ allows each network device to handle packets on an individual basis, as soon as they show. Each network device can be configured to follow specific QoS policies independently. No advance reservations are required, and this provides scalability. While IntServ applies QoS policies in a per flow basis, DiffServ applies QoS decisions in a per hop basis. This means the QoS decisions are made depending on packet header information and is independent in each hop from the source to the destination as long as QoS is configured at each of these hops.

DiffServ QoS offers a **per hop behavior**. Each router or switch that receives the packet inspects its header and identifies certain parameters that will let them know how to proceed about forwarding the packet. The packet cannot change the switch or router behavior in respect to its forwarding decision. It simply presents certain criteria and depending on the switch or router's configuration, a forwarding decision is made. This happens, as we have mentioned, with each network device (router or switch) along the path to the destination of the packet.

Layer 2 QoS Classification

Layer 2 QoS is possible because when a frame exits out a trunk port, a frame tag is added to identify the VLAN the frame belongs to. The encapsulation to provide the tag also includes a Class of Service (CoS) field that can be used at switch boundaries to makes some prioritization and QoS decisions in general. There are two frame tagging encapsulations and they both handle Class of Service differently:

- **IEEE 802.1Q –** With VLAN tagging, each frame is tagged with a 32-bit field situated between the source MAC address and the EtherType fields. The first 16 bits of the field is the Tag Protocol Identifier, which actually identifies the frame as an IEEE 802.1Q frame. The next 3 bits comprise the Priority Code Point field, which indicates 802.1p priority values from 0 to 7; 0 is the lowest priority and 7 the highest. After the PCP, there's a 1 bit CFI field and the 12 bit VLAN ID.
- **Inter-Switch Link (ISL) –** These frames are tagged with a 15 bit VLAN ID and 4 bit USER field. The lowest 3 bits of the user field are used as the CoS value. Although ISL is not standard-based, Cisco switches take the CoS values from one encapsulation and add it to another as they pass the switch. This makes both encapsulation types interoperable from the QoS perspective.

The point of having QoS at both Layer 2 and Layer 3 is that as a packet travels through our network it will move through routers, switches and MLS that could be operating as a Layer 3 or Layer 2 device. The CoS and ToS markings allow these devices to filter traffic and prioritize time-sensitive and mission critical applications, at both Layer 2 and 3. If Layer 2 QoS did not exist on networks that experiences network congestion, the crucial traffic could suffer in those points where a Layer 2 forwarding decision needs to be made.

Layer 3 Quality of Service (QoS)

IP packets have a 1 byte Type of Service (ToS) field in the header that has always been used to mark the packet. The field is divided into a 3-bit IP precedence, the actual bits used for QoS markings, and a 4-bit ToS. To provide a more scalable method, DiffServ reformats the same 1-byte field. Understand that what changed was the way the routers and switches process the packet after reading the markings, the actual fields remain the same. The field is now called Differentiated Services (DS) field. The old and new formats are represented in the following figure:

Tos Byte:	P2	P1	P0	T3	T2	T1	T0	Zero
DS Byte:	DS5	DS4	DS3	DS2	DS1	DS0	ECN1	ECN0
	(Class Selector)			(Drop Precedence)				

Figure 30: Type of Service/Differentiated Services Comparison

Notice that only 6 bits are used for markings. That 6-bit DS value, known as Differentiated Services code point (DSCP) is the value examined for QoS markings in DiffServ network devices.

The Class Selector takes the place of the old IP precedence value, and the Drop Precedence 3-bit value is now added and used in the QoS DiffServ. Bits represented by DS5-DS3 are the class selector and as you should know, its value ranges from 0 to 7. The drop precedence is a bit different, with values ranging from 0-3. Be careful with this because it is confusing at first. What you must understand is that the DS0 value is always 0, not 1, and that is why the value range is 0-3.

As you might have noticed, there is some level of backward compatibility with DSCP and ToS, IP Precedence. When a non-DiffServ receives an IP packet it recognizes the Class Selector bits as IP precedence bits and can classify and apply QoS criteria and traffic engineering when deciding when and how to forward the packet.

Bits DS5 to DS3, the class selector bits, classify packets in 8 categories (3 bits = [0-7]):

- **Class 0** –No QoS is used and only "best-effort delivery" is ensured. This class should be used for standard data that can withstand latency, delay and jitter such as FTP traffic.
- **Class 1 through 4 –Assured forwarding (AF)** service levels, which allows for four priority levels and the higher the level, the higher the priority of the traffic.
- **Class 5 –Expedited forwarding (EF)** could be considered packets given "premium service" and are the least likely to be dropped. Class 5 is useful for time-critical data, such as voice and live video.
- **Classes 6 and 7 –**Internetwork control and network control, respectively and are reserved for control-based traffic, such as inter-router and inter-switch communication, like STP or routing protocol communications.

As we can see in the figure, each Class Selector represented in the DSCP has 3 levels of drop precedence (3 bits, but remember DS0 = 0, that means DS1 = 1 and DS3 = 2, the maximum decimal value is 0+1+2=3). In the case of the drop precedence, high means worse. Again, if two packets arrive at a router and they are both marked as class 5, the tie breaker is the drop precedence and high drop precedence will be dropped before lower drop precedence. Example: A packet marked class 5, drop precedence 1 will be forwarded faster than another class 5, drop precedence 2 or 3, and it's less likely to be dropped.

Remember, a lower drop precedence will receive a "better and faster" service (will have priority) over a higher drop precedence, given that they are both from the same class. Drop precedence can be:

- Low = 1
- Medium = 2
- High = 3

QoS for Voice traffic

The first thing to do when QoS is implemented is classify the level of service the switch should give to the packets it receives. This process of classification uses criteria such as type of traffic based on Layer 4 protocol and port number (application), or according to parameters matched and defined by an access list.

The switch or router must first decide if it trusts the QoS marking when it receives a packet in one of its ports. If the client device (such as an IP phone) can mark packets, you have to decide if it can be trusted or rewritten by the switch. If it decides it trusts the packet, the QoS markings are used to make a decision.

Network designers and administrators must define the points in the network that will receive trusted packets from the QoS perspective. The perimeter formed by network devices that don't trust QoS markings in packets is called ***trust boundary***. Usually trust boundaries extend from access layer switches to the points where the network administrator loses control of the data, the WAN or ISP demarcation points. When the boundary has been identified and established, pretty much everything inside of it should be configured to trust any QoS settings/markings received in packets/frames. Also note that the QoS markings can be modified at any point inside your network or outside your network if you lose administrative control of the destination.

Configuring QoS Trust Boundaries

As the result of the new CCNP curriculum, a lot of the QoS material that used to be part of the CCNP was removed. Quality of Service and traffic engineering in general is a complex subject, worth its own course and even specialization. As a CCNP you must have a basic understanding of the technology and according to the exam blueprint (always check Cisco's Website for the most updated information regarding exams), you must be able to configure a basic implementation, with the use of a set of very useful macro commands, with Auto-QoS. The QOS 642-642 exam certifies mastery of QoS in Cisco devices. The training available for those exams will definitely make you an expert in the subject, and some of the books are recommended readings for those with a CCIE in mind.

The trust boundary should be configured at the edges of our networks. These are the points where data enters and leaves our administrative control. Knowing that imaginary line that defines the trust boundary and that it is a recommended practice to define the switch ports from the IP phones as untrusted, we can configure the trust boundary.

First, let's understand the reason why data coming from ports of the IP phone shouldn't be trusted. Those ports are available for anyone to connect and often times, they are not physically secure. Access ports are commonly the source point of attacks or malicious use. In this case, a malicious use could be a user setting a higher QoS priority for these packets, which could result in the network devices dropping truly sensitive data before dropping packets 'maliciously' marked with higher priority. With this defined, we are almost ready to proceed with the steps to configure the boundary.

A switch communicates initially with the IP phone through CDP messages. That is the same mechanism used by the switch to instruct the IP phone that it should extend the QoS trust to its switch port. The procedure to achieve this is the following:

Step 1: Enable QoS in the switch:

```
Switch(config)# mls qos
```

Step 2: Define the QoS parameters that will be trusted:

```
Switch(config)# interface type mod/num
Switch(config-if)# mls qos trust {cos | ip-precedence | dscp}
```

Step 3: Instruct the IP phone to extend the trust boundary:

```
Switch(config-if)# switchport priority extend {cos value | trust}
```

1. Enable QoS with **mls qos**.
2. Define trusted QoS parameters:
   a. **interface** *mod/num*
   b. **mls qos trust** {**cos | ip-precedence | dscp**}
3. Configure the IP phone to extend the trust boundary: **switchport priority extend** {**cos** *value* | **trust**}

As we mentioned previously, the packets coming from the switch port of the IP phone shouldn't be trusted from a QoS perspective, because a user can easily spoof CoS settings in order to have premium network service, at the expense of really critical applications such as the ones providing voice services. If the incoming packets can't be trusted, the CoS value=0 must be chosen. Notice that this has the same effect of simply leaving the default, which is untrusted, and makes the IP phone set the value to 0 when it receives packets before sending them in the uplink to the switch. There are instances where the port must be trusted. The trust parameter must be used. In other cases we need to define the QoS markings/priority values and in those cases the value parameter must be set accordingly (a higher value than 0).

All other ports inside the trust boundary must be configured as trusted ports. Everything inside the trust boundary is considered a part of a ***trust domain***. The point where QoS are ignored and rewritten is the trust boundary. After it enters the domain, QoS values must be used for priority decisions.
To configure a switchport to be trusted, use the following commands:

```
Switch(config)# interface type mod/num
Switch(config-if)# mls qos trust cos
```

With this configuration the switch will trust all CoS settings received in the port. In the next section we will show how we can configure a basic yet powerful QoS deployment with the use of the Auto-QoS feature.

Simplifying QoS Configuring with Auto-QOS

QoS is a huge topic, worth its own course and exam in the world of Cisco. The Cisco QOS 642-642 exam is part of the CCIP and CCVP certifications. The huge scope of Quality of Service features is the reason why the subject was mostly cut from the CCNP curriculum. More emphasis is now given to actual routing and switching.

Auto-QoS was developed precisely to simplify QoS configurations and deployments. It consists of a series of macro commands that are run in specific points on the networks. Given the nature of the macro commands it is recommended that you only use Auto-QoS in switches that have default QoS settings. If the switch has already been configured with non default settings, the commands run by the macros could alter the previously configured settings and this could result in network instabilities and malfunctions. Auto-QoS is meant to be used mostly in access layer switches, in ports connecting the IP phones to the network, not necessarily the network core.

Auto-QoS automatically configures and sets up the following QoS features:

- Enables QoS.
- CoS to DSCP mappings for QoS markings.
- Ingress and egress queue tuning.
- Strict priority queues for egress voice traffic.
- Establishing an interface QoS trust boundary.

To configure Auto-QoS, identify an interface at the trust boundary and use the **auto qos voip** {**cisco-phone** | **cisco-softphone** | **trust**} interface configuration command. The rest is extremely simple. If a Cisco IP phone is connected to the port, the **cisco-phone** keyword must be used. If a PC running Cisco Communicator IP softphone is connected, use the **cisco-softphone** keyword. If the switch is connected to another switch or router inside the trust domain use the **trust** keyword.

The **auto qos voip** is a macro command that uses several commands in the interface that are not displayed until you look at the running configuration. If you need to see the commands in real-time after you use the **auto qos voip** interface configuration command, you must first use the **debug auto qos** EXEC command. When you are done watching the commands being generated remember to turn off the debugging with the **no debug auto qos privileged** EXEC command.

Verifying VoIP QoS Implementations

You can verify the QoS trust setting of a port with the **show mls qos interface type** *mod/num* EXEC command. The following is an actual switch output:

```
Switch# show mls qos interface fastethernet 0/5
FastEthernet0/5
trust state:   trust cos
trust mode:   trust cos
trust enabled flag:   ena
COS override:   dis
default COS:   0
DSCP Mutation Map:    Default DSCP Mutation Map
Trust device:    none
Switch#
```

Figure 31: Verifying QoS Configuration

There you can see the trust state and trust mode setting. This particular switch port in our example was configured to trust incoming frames and the trust state: trust cos indicates that it is operating as expected.

In this case, an IP phone is connected and frames coming to the port that are tagged with QoS settings are being trusted by the switch and used to properly queue the packets based on priority. Remember this means their CoS markings won't be changed and the frames will be handled accordingly.

Another important command is **show interface type** *mod/num* **switchport**. It is used to display whether or not the IP phone's connected devices are trusted. That is, if the **switchport priority extend** {**cos** *value* | **trust**} interface configuration command was used to extend the trust boundary to the IP phone's connected devices.

```
Switch# show interface fastethernet 0/5 switchport
Name: Fa0/5
Switchport: Enabled
[output deleted...]
Voice VLAN: 20 (VLAN0020)
Appliance trust: none
Switch#
```

Figure 32: Showing Trust Boundaries

In this example, the connected PC is not inside the trust boundary. When it is, you will see the message "Appliance trust: trusted".

When you configure Auto-QoS in an interface, the commands associated with the macro will be executed in the interface. To display the interface configuration commands use the **show running-config interface** type mod/num EXEC command. Here is a sample output:

```
Switch# show running-config interface fastethernet 0/5
Building configuration...
Current configuration :   272 bytes
!
interface FastEthernet0/5
switchport access vlan 10
switchport trunk encapsulation dot1q
switchport mode access
switchport voice vlan 20
mls qos trust device cisco-phone
mls qos trust cos
no mdix auto
end
Switch#
```

Figure 33: Verifying QoS with running-config

As we said, you can verify if the IP phone is connected and trust status with the **show mls qos interface type** *mod/num* EXEC command.

```
Switch# show mls qos interface fastethernet 0/5
FastEthernet0/5
trust state: not trusted
trust mode: trust cos
trust enabled flag: dis
COS override: dis
default COS: 0
DSCP Mutation Map: Default DSCP Mutation Map
Trust device: cisco-phone
Switch#
```

Figure 34: Verifying QoS Trust Status

Take a look at the fourth line of the output, **trust enabled "**flag: dis". This means the trust parameter is not enabled (disabled).

Finally, you can verify Auto-QoS interface configuration with the **show auto qos** *[interface type mod/num]* EXEC command.

Integrating Wireless LANs to the Wired Network

In this section we will introduce the basics of 802.11 wireless networks and how the enterprise and campus switched networks need to be prepared in order to provide security, scalability and reliable wireless access to its resources.

Wireless networks are here to stay. Every day the demand for wireless network connectivity increases. At the beginning, wireless connections were exclusive for laptops, and they themselves were not very common. Now in the United States laptops outsell desktop computers by large numbers. On top of that, smart phones, PDAs, tablets, video game consoles and even IP phones are now being connected to our wired network wirelessly through Access Points.

As a CCNP you should be familiar with wireless technologies and must be ready to implement the mechanisms that allow our wired networks to serve wireless clients reliably while protecting data.

Wireless LANs

Wireless networks provide connectivity to the traditional network and its resources. It extends the physical layer capabilities to provide for connection that has been traditionally achieved through wires, without them.

There are differences between wireless and wired networks. As you might remember from the CCNA studies, the Ethernet mechanism to deal with collisions in a shared media is the Carrier sense collision detection (CSMA/CD) which basically is a method to detect and prevent collisions and provide the means for recovery when they do occur. In the wired network, the physical layer guarantees that only a certain amount of users connect to the media. The collision problems were solved with the implementation of the full duplex mode.

On the other hand, a wireless network's physical layer is provided by a specific wireless frequency range where the signals are transferred and received. These frequencies are set by the IEEE standards board with agreements with various governments where these open wireless standards can be used within a wireless spectrum. Wireless networks for this reason are a shared physical medium, where an unlimited number of users can use the shared media, the open frequencies, at any time. Collisions are simply a fact in 802.11 wireless networks, and something to be dealt with, because every single connection is in half-duplex.

Although it would be practical and implementable to give wireless networks the full duplex capability, the current standard IEEE 802.11 doesn't permit it. Full duplex operation could be provided by using different frequencies to transmit and receive data.

Avoiding Collisions in the WLANs

When two or more clients in the wireless network transmit at the same time, and using the same frequencies, signals become mixed. The receiving end sees the resulting product as garbled data and noise. Stations won't notice it because their receivers must be turned off when they are transmitting.

To prevent data loss because of collisions, acknowledgements must be sent by the receiving station for the frames received. This provides for a collision detection tool but doesn't prevent the collision from actually happening.

The IEEE 802.11 standard defined and uses the Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) method to prevent collisions from happening. CSMA/CA requires that all stations listen before transmitting. When a station needs to transmit, there are two and only two possibilities:

1. No other device is transmitting: The sender can transmit immediately. The receiving station must send acknowledgement of the frames received to confirm that the data was delivered without the effects of a collision.
2. Another device is transmitting: The station must wait until the other transmitting station finishes and then wait a random amount of time before attempting to transmit.

The amount of time the stations must wait before attempting a transmission is dictated by the frame size, which is the determining factor of the size of the DCF interframe space (DIFS).

In addition to the duration timer (DIFS), every wireless end station must implement a random back off time, which will have to run off before the station transmits its frame. The idea is that all stations in the wireless domain wait a period of time before trying to transmit beyond the DIFS, in order to minimize the chances of a collision because the backoff timers ran off at the same time. This process is called the Distributed Coordination Function (DCF). Also note that the more end stations you have on a wireless network, the more likely collisions will occur. If too many devices try to connect to the same 802.11 network and try to send and receive data, the wireless network will be so overrun with collisions and broadcast messages that the wireless network will become useless.

IEEE 802.11 defines any group of wireless clients as a service set. Devices in the same service set must share a service set identifier (SSID), which is nothing more than a text string included in every frame sent. The SSID must match between hosts (sender and receiver) before they can communicate.

A PC or any end-user device must have a compatible wireless network adapter and software that interacts with the wireless protocols or a supplicant.

The IEEE 802.11 standard also allows two or more hosts to directly connect with each other without an AP or any other gateway. This is known as an independent basic service set (IBSS) or ad-hoc wireless network.

An IEEE 802.11 BBS centralizes access and control over a group of wireless devices with a Wireless Access Point (WAP) as the hub of the service set. Wireless clients who wish to connect must agree with the WAP on the following parameters:

- A matching SSID
- Compatible wireless data rate
- Authentication type and credentials

The client starts the association process with an association request message. The AP must respond with an association reply, granting or denying the association. After the client is associated with the AP, all traffic to and from the client must pass through the AP. Clients connected to the same AP cannot directly connect or change data without the intervention of the AP.

WAP Operation

The APs main function is to connect wireless clients to the wired network. It provides access to the wired network and manages the wireless network, giving access to wireless clients just as if they were directly connected to the wired network.

A WAP can also act as a wireless bridge to form a single wireless link between one LAN to another over short or long distances depending on the hardware being used.

Cisco even has a WAP platform that allows the bridging of wireless LAN traffic from AP to AP. This allows for a big wireless WAN without the use of cables: Each AP picks the other's signal and forms a big mesh, providing a big WLAN exclusively through wireless connections.

WAPs act as bridges that take network information from two different media (Layer 1) and merge them in Layer 2. APs are in charge of mapping VLAN information to an SSID. The WAP uses an 802.1Q tag to map the VLAN to the SSID.

When an AP must map more than one VLAN to more than one SSID (the AP has wireless clients connected to different VLANs/subnets) it must be connected to the switch by a trunk and it must allow the required VLANs.

Wireless LAN Cells

The Wireless LAN cell is the imaginary *volume* surrounding the WAP's antenna in which hosts can access the wireless LAN. It is simply the coverage area of the WAP. Generally, the closer you are to a WAP, the stronger the signal. Notice the word *volume*. It is a common mistake to forget that WAP's coverage is three-dimensional and affects floors above and below in a building. The cell is generally represented in a floor plan as a two-dimensional circle in very basic signal strength diagrams.

Careful thought must be given to the AP's placement within a building or external position so that it can provide the coverage area that is needed. Remember the nature of the WLAN will make it operate under constantly changing conditions. The best approach to determine the AP placement is to conduct a wireless site survey.

In a site survey an AP is placed in a desirable place and one or more possible clients move in the expected coverage area taking measurements of the signal strength and quality. The point is to plot the AP range using the actual interfering equipment and also the usual host devices.

To provide a wider coverage area, it is common to overlap cell areas by a small percentage. That provides connectivity to users that might move around in the overlapping cells. If two AP's coverage must overlap in order to provide a bigger WLAN coverage, they must never use the same frequency. If they did, they would only interfere with each other, because they would be using the same medium to communicate, increasing the likelihood of a collision.

Moving from one AP to another is called **wireless roaming**.

Wireless roaming can be at layer 2 or layer 3. Layer 3 roaming occurs when the client changes its association to a new AP and changes its current VLAN and IP subnet to a different one. Layer 2 wireless roaming occurs when the client associates to a new WAP and maintains its IP address.

When designing a WLAN, providing the largest coverage area per WAP might seem like the most viable option, because in a big deployment, an increase coverage area per AP could represent big savings, but there are many considerations to have in mind. A larger cell also opens the possibility of **overcrowding**. Remember the WLAN operates in a shared medium environment, where collisions can occur. If too many associations are provided, clients will constantly be competing for the limited resources thus bandwidth and airtime could be limited.

It is often beneficial to reduce the size of the cell by reducing the transmit power of the WAP signal sent out through the attached antenna. This smaller area guarantees that fewer hosts associate with the WAP and bandwidth will be available at the highest rate. This is an especially sound policy when hosts will be using mission critical applications and bandwidth intensive traffic such as voice or video.

The WLAN Architecture

An autonomous mode WAP is a standalone AP that is centrally positioned to support its clients. These are often referred to as an aAP. It is isolated, configured individually, handles its own use of radio frequency (RF) and enforces its own security policies and so on.

Since all aAPs are "autonomous", managing security and other policies such as quality of service, bandwidth policing and so on, are very difficult, because each aAP must be configured and managed individually.

Managing the RF operation is also a big problem under this architecture because the network administrator must select and configure manually each channel. The power output must be managed too in order to prevent "blind spots" or coverage holes, or on the opposite side, that the signals overlap too much.

Recognizing the issues and shortcomings of the aAP, Cisco developed the Cisco Unified Wireless Network Architecture.

Cisco Unified Wireless Network Architecture

The Cisco Unified Wireless Network Architecture is a collection of equipment that performs a set of functions that are an integral part of a wireless network. As we mentioned earlier, the new architecture solves all the issues generated by the autonomous AP individual management requirements. The new architecture offers centralized management where all WAPs can be managed, monitored and compared from one location. The centralized features in this new model include:

- WLAN management
- WLAN security
- WLAN control
- WLAN deployment

The centralization of the functions of the individual autonomous APs, is achieved by relaying its functions to a central point. This central point is the WLAN controller.

The functions of the AP can be classified as real time processing and management.

The real time processing activities are the following:

- RF transmit and receive
- MAC management
- Encryption

The AP's management activities are the following:

- RF Management
- Association and Roaming management
- Client authentication
- Quality of Service
- Security Management

The real time processes involve sending and receiving 802.11 frames, AP beacons and probe messages and data encryption. These functions must be close to the clients because they are performed at the Layer 2 of the OSI model, the MAC layer. For this reason these functions remain performed at the AP in the Cisco Unified Wireless Network Architecture.

The type of WAP used in a Cisco Unified Wireless Architecture is called **Lightweight AP** (LAP) in the new architecture and performs only the real time functions. The lightweight term is given because the code image and, most importantly, the local intelligence are stripped down. The lighter nature of the new functions when compared to the legacy model is the reason for its name.

The management functions don't involve handling frames over the RF channels. These should be centrally managed by a Wireless LAN Controller (WLC). A WLC manages several LAPs scattered around the switched network. The LAPs becomes totally dependent of the WLC for management functions.

This separation of functions is called split-MAC architecture, and occurs for every LAP in the network, that must register with its WLC at boot up to get operating information such as RF channels to be used, authentication, and security and so on. The LAP remains handling the real time operations.

The binding of the LAP and the WLC occurs when the LAP boots up, and it's a required step before the LAP becomes a functional Access point. They do this by creating a tunnel where the 802.11 related data travels between the devices. The WLC and LAP can be in the same VLAN or IP subnet or in a different one. This is made possible by the **tunnel,** by encapsulating the data between the WCL and the LAP within a new IP packet.

To create and support the **tunneling system**, the WLC and LAP use the **Lightweight Access Point protocol (LWAPP)**, developed by Cisco, or the **Control and Provisioning Wireless Access Points protocol (CAPWAP, defined in RFC 4118)**. Both protocols utilize two different tunnels, one for actual client data and another for control messages.

Control messages are the ones used to control and managed the LAP. They use authentication and encryption to provide security.

Data are the packets to and from the wireless clients. The data is encapsulated in LWAPP or CAPWAP IP packets but no encryption or other method of security is provided for this communication.

LWAPP uses UDP destination ports 12222 and 12223 on the WLC end. Similarly, CAPWAP uses UDP ports 5246 and 5247. Both protocols use digital certificates installed at the moment of the purchase. The certificates are used to authenticate the devices before the tunnels are created.

WLC functions:

- **Dynamic channel assignment –** The WLC chooses and assigns RF channels to each of its LAPs based in the previous assignments to other surrounding LAPs.
- **Transmit power optimization –** The WLC decides the transmit power of each LAP based on the coverage needed. It also runs periodic checks to adjust transmit power and then compares these checks with neighboring LAPs. The WLC then corrects signal strength of the LAPs to get the optimal signal strength overlap.
- **Self healing wireless coverage –** If a LAP dies, the transmit power of surrounding LAPs is increased to cover the coverage hole as best as it possibly can.
- **Flexible roaming –** Clients can roam at Layer 2 or Layer 3 with very fast roaming times.
- **Dynamic client load balancing –** If two or more LAPs are configured to cover the same area, the WLC can associate clients with the least used LAP. This provides for efficient load balancing.
- **RF monitoring –** The WLC instructs the LAP to monitor the RF channels usage. By receiving RF information, the WLC can decide later RF assignments based on the state of the different channels: level of noise, interference, rogue APs, etc.
- **Security Management**.

In large wireless deployments, managing the WLC can be a daunting task. That is why Cisco developed the Wireless Control System (WCS), an optional server platform that can be used to control multiple WLCs deployed in the network with the great easiness provided by a GUI. The WCS can be used to perform most WLAN management tasks.

It is possible to display dynamic representations of the wireless coverage provided by the APs, using building floor plans. The WCS can be used to locate any wireless client by triangulating its position using the client's signal as received by multiple LAPs. This is very useful when trying to find a malicious user or a rogue device.

WCS can also be optionally connected to the Cisco Wireless Location Appliance to track the location of all your connected wireless clients on your network. This tracking is done with the use of the MAC address and it's very useful when tracking corporate assets that are mobile in the WLAN is required.

Lightweight AP operation

The LAP was designed so that no direct configuration was required to operate it. This means that no configuration or management is performed directly to the LAP through the console port or by any other means. Essentially, it is a dumb device with a wireless radio and Ethernet connection. The LAP must connect to the WLC and get all its configuration parameters.

There are several steps that the LAP has to go through before it becomes operational:

1. LAP receives an IP address from the DHCP server.
2. LAP learns available WLC IP addresses.
3. LAP joins the first WLC in its address list, or moves down the list in the event of a connection failure. This process repeats until LAP is connected to a WLC.
4. The WLC compares LAP code images with the LAP. If the WLC possesses newer code, the LAP downloads the image and reboots itself.
5. The WLC and LAP build two secure tunnels between each other, one for management traffic and a second for wireless client data. Wireless client data is not secured on the LWAPP or CAPWAP tunnel.

The LAP can learn the list of addresses to WLCs through DHCP, through the use of the option 43, or broadcasting a join request message. Keep in mind that you must specifically configure your DHCP server to send the IP addresses of the WLC(s) using option 43. For this second option the LAP and WLC must be in the same IP subnet/VLAN.

If the LAP loses connection to the WLC, all wireless clients connected to the LAP lose connectivity. Cisco developed the Hybrid Remote Edge Access Point, to allow a LAP to maintain network connectivity to wireless clients even if the connection to the WLC is lost. This is used in cases where the LAP is connected to the WLC through a slow link WAN. This allows wireless clients to remain communicated within the remote network until the WAN link is restored.

Roaming in a Cisco Unified Wireless Network

In the new architecture, the LAP deals exclusively with the real time operations. However, a wireless client needs to associate with the LAP, but it is the WLC who handles the associations. This provides a centralized, faster and easier management of the associations.

With autonomous APs, roaming is performed at the Layer 2, and Layer 3 roaming requires special equipment. When the client moves, it has to negotiate the roam with every AP along the way. On the other hand, with LAPs, the roaming occurs from WLC to WLC, although from a wireless client perspective the association is being moved from AP to AP. In the new architecture, the client can maintain its IP address even when roaming between controllers.

Intracontroller Roaming: This happens when the wireless client moves from the coverage area of one AP to the coverage of another AP, but both are controlled by the same WLC. The WLC only needs to update its tables to begin using the LWAPP and CAPWAP tunnels in the link connected to the AP associated to the client.

Intercontroller Roaming: This happens when the client moves to an area covered by an AP that is managed by a different WLC than the one it was associated to. There are two possibilities:

1. **Both WLC are in the same IP subnet / VLAN** – The client associates to both the new AP and the new WLC and the change is communicated using mobility message exchange where the information from the client is transferred from one WLC to another. This process is totally transparent to the user.
2. **WLCs are in different VLANs / IP subnets** – When the client moves to an AP-WLC that is in a different VLAN than the one it is coming from, the two WLCs must create an Ether-IP tunnel to connect each other and send the roaming client data to the original WLC. The tunnel provides encapsulation of a Layer 2 frame inside an IP packet, using protocol 97. To move packets to and from the client, one controller encapsulates packets and the other receiving controller de-encapsulates them, where they appear in their original form. Any controller serving the client from a different subnet is called a foreign agent. As the client roams to another LAP-WLC combo, the anchor WLC will follow its track creating Ether-IP tunnels as necessary.

Mobility Groups

Mobility groups are logical groups of WLCs and LAPs. Its characteristic is that a client can roam through each and every one of the members of the group.

A mobility group can contain up to 24 WLCs, and the number of LAPs depends on the capabilities of the WLCs.

A wireless client can move to a LAP that is part of a different mobility group, but its IP address will have to be renewed and all the session information contained in the previous WLC will be dropped.

Configuring Switch Ports for WLAN Use

The actual configuration of the APs and WLCs was removed from the new CCNP SWITCH 642-813 exam. The new CCNP is a routing and switching professional, and he/she should configure the wired LAN to support the WLAN and the wireless specialist should handle the actual WLAN and equipment configuration. In this guide we will learn how to configure switch ports to support the WLCs and APs and the wireless network in general.

Configuring Port Support for Autonomous APs

APs are usually connected to an access layer switch. Every SSID is mapped to a VLAN. In the case the AP offers more than one SSID, several VLANs must be mapped to the switch port. In this case, the port connecting to the AP must be configured as a trunk and the VLANs that will be used must be allowed.

If we assume the AP is connected to the Fast Ethernet 0/6 port of a switch and VLAN 100, 200 and 300 are going to be used for three different SSID, the configuration must be as follows:

```
Switch(config)# interface fastethernet 0/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport trunk allowed vlan 100,200,300
Switch(config-if)# switchport mode trunk
```

Configuring Support for LAPs

As we said, WLCs are designed to be "zero-touch" devices, meaning they are almost always operational right out of the box. The WLC manages almost all functions and operations of the LAP.

The LAP must be connected to an access port, never a trunk. As explained, the VLANs required will travel across the LWAPP or CAPWAP tunnels that will be created between the WLC and the LAP.

The following is a sample configuration of a switch access port connected to a LAP. We will use VLAN 10 as an access VLAN in the port connected to the LAP.

```
Switch(config)# vlan 10
Switch(config-vlan)# name wifi-management
Switch(config-vlan)# exit
Switch(config)# interface fastethernet 0/10
Switch(config-if)# switchport
Switch(config-if)# switchport access vlan 10
Switch(config-if)# switchport mode access
Switch(config-if)# spanning-tree portfast
Switch(config-if)# power inline auto
Switch(config-if)# exit
```

In this configuration we used the VLAN 10. We configured the port as an access port and use the **spanning-tree portfast** interface configuration command to allow the port to be up immediately without extensive and time-consuming spanning tree calculations. The **power inline auto** will provide PoE to the LAP negotiating the amount of power to be provided.

Configuring Switch Port Support for a WLC

WLCs must be in the distribution layer of the campus network, because they aggregate WLAN traffic from the LAPs.

The main thing you must have in mind is that all VLANs that will be tunneled to the LAPs must be accessible by the WLC. This means the links from the switch that are used to connect to the WLC must be trunks. This is one of those rare situations where dragging VLANs between distribution blocks may be necessary.

The following is a sample configuration:

```
Switch(config)# interface range fastethernet 0/10
Switch(config-if)# switchport
Switch(config-if)# switchport encapsulation dot1q
Switch(config-if)# switchport trunk allowed vlan 100,200,300
Switch(config-if)# switchport mode trunk
Switch(config-if)# spanning-tree portfast trunk
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

Domain 5: High Availability

As we have studied, Multilayer switches can act as default gateway for hosts in the same VLAN with the use of SVIs. They can switch packets at Layer 3 (routing). High availability refers to the provision of the redundancy of routing services provided by routers or multilayer switches.

The redundancy can be provided by adding duplicate hardware such as an additional router or multilayer switch configured for that purpose.

We will discuss several approaches and methods of providing router redundancy. The so called first hop redundancy protocols (FHRP).

Hot Standby Router Protocol (HSRP)

Hot Standby Router Protocol (HSRP) is a Cisco Proprietary protocol developed to allow several routers or Layer 3 multilayer switch interfaces to appear with only one IP address. RFC 2281 defines this protocol in detail. The HSRP group can be composed by MLS and routers at the same time.

All the routers that will provide redundancy will be part of an HSRP group. There will be a primary router, also called *active* router, a secondary router is elected as a *standby* router and other participating routers will be in the *listen* state.

Routers in the HSRP group communicate exchanging hello messages using the multicast destination 224.0.0.2 (all routers). The messages are sent at a regular interval in order to let each other know of their existence and which one is the active router. The hello messages are sent using UDP port 1985. Only active and standby routers exchange messages once the active and standby routers have been elected.

A HSRP group can be assigned a unique group number from 0 to 255, but most Cisco Catalyst switches support only 16 unique group numbers. For this reason it is recommended that you use the same group number for every interface in the same VLAN. The HSRP number is only locally significant on an interface.

HSRP Router Election Process

The HSRP router election is based in a priority value that ranges from 0 to 255. It is configured in a per interface basis and the default in Cisco catalyst switches is 100. If the value is left to the default, the tie breaker is the IP address: the router with the highest IP address in the group becomes the active router. The second highest priority becomes the standby router while the other participating routers enter the listen state. You will want to change the priority on the switch that you wish to be the active HSRP gateway.

The standby router is the only that should be constantly monitoring hellos from the active router.
When HSRP is configured in a router, it must pass through several states. First it must exchange hello messages with the other participating routers to determine its function in the HSRP group. All devices participating in HSRP must progress through the following states until they become either Active or Standby:

1. Disabled
2. Init
3. Listen
4. Speak
5. Standby
6. Active

To configure a router to participate in a HSRP group, use the following interface configuration command:

```
Switch(config-if)# standby group priority priority
```

If you plan to make this router the active router for group 1, assuming the other participating routers are left with the default priority, 100, you only need to use a higher number, from 101 to 255, as follows:

```
Switch(config-if)# standby 1 priority 200
```

The hello timer default is 3 seconds, and the holdtime timer is usually three times the hello. The default is 10 seconds. If the standby router doesn't hear a hello for the holdtime timer period, the active router is considered down and the standby can take its functions. A router in the listen state then becomes the standby router.

Remember the holdtimer should be 3 times the hello timer as maximum. Use the following command to configure both timers:

```
Switch(config-if)# standby group timers [msec] hello [msec] holdtime
```

If you use the **msec** keyword, the value can be a range from 15-999 milliseconds.

By default, the active router election is not preemptive. This means that if an active router fails, it will not preempt to once again become the active router when it comes back up. This also means that when routers are powered up or added to a network, the first router to bring up its interface will be the one that becomes the HSRP active router.

You can change this behavior and make the layer 3 interface with the highest priority become the active router as soon as it joins the HSRP group if it has the lowest priority. You can achieve this with the following command:

```
Switch(config-if)# standby group preempt [delay [minimum seconds]
[reload seconds]]
```

After using this command the router can preempt another router with the active router function. The **delay** keyword is used to configured an amount of time specified in a value represented in the word *seconds*(ranging from 0 to 3600 seconds) before ***attempting*** to preempt the current active router it finds after its interface comes up or when the router joins the HSRP group.

The **reload** keyword is used to force the router to wait for *seconds* after it has joined the group or restarted. This is useful because routing protocols usually take some time to populate the routing table and a router should not act as the default gateway-first hop- until it has all the necessary routes the host might need to reach.

HSRP Authentication

Authentication can be used to prevent rogue devices from taking part in the HSRP group. All devices in the group must be configured with the same authentication method and key. You can use MD5 authentication or plain text.

Plain text authentication offers the most basic form of authentication. The HSRP messages are sent with a plain text key string of up to 8 characters. If the key string in the message matches the one configured in the router the message is accepted. Use the following interface configuration command:

```
Switch(config-if)# standby group authentication string
```

Remember that the authentication string is sent in the clear. No encryption is provided. This means that if a packet is intercepted, the attacker can see the key string and could send its own malicious HSRP message and effectively gain access to the network, probably acting as a default gateway. This is why this method is considered basic and mostly insecure. MD5 authentication is recommended.

MD5 Authentication

An MD5 hash is computed on a portion of each HSRP message using a shared secret key known only by legitimate HSRP group peers. Every HSRP message sent by a participating router has the MD5 hash. When a router receives the HSRP message it recalculates the message and compares it to its secret key. If the hash are identical message is validated and accepted, else the message is denied.

To configure MD5 authentication in the HSRP router, use the following interface configuration command:

```
Switch(config-if)# standby group authentication md5 key-string [0 | 7] string
```

The string value can be a chain of up to 64 characters. The default key-string value is 0, which means the key will be communicated plain text. After the key is entered it is shown as an encrypted value in the switch configuration.

You can also configure an MD5 key string as a key on a key chain. The following are the commands required to configure the key and the MD5 authentication for the HSRP group.

```
Switch(config)# key chain chain-name
Switch(config-keychain)# key key-number
Switch(config-keychain-key)# key-string [0 | 7] string
Switch(config)# interface type mod/num
Switch(config-if)# standby group authentication md5 key-chain chain-name
```

This optional configuration is not available in all catalyst switches.

You can configure a HSRP router to decrement its priority if certain links go down. This is useful because a certain router can have several links to the outside world, and as it loses those links, there might be other routers in the HSRP group that could become more desirable to route packets. You can configure a router to decrement its priority value if an interface goes down with the following interface configuration command:

```
Switch(config-if)# standby group track type mod/num [decrementvalue]
```

The default *decrementvalue* value is 10. Remember that it is not the HSRP router interface the one that should affect the priority. This is mostly useful and was developed mainly to evaluate the usefulness of the router as a default gateway. That means you should configure that command in those interfaces that are generally to connect to the outside world (outside of the VLAN of IP subnet). Remember that a router will only become the active router in a HSRP group if it has a higher HSRP priority and if it is using preempt in its HSRP configuration.

HSRP Addressing

Each interface of the HSRP router must be configured with an IP address that is meant to be used by routing protocols and management traffic of the router. It is also configured with the virtual router IP address, the one used as a default gateway and shared by all participating router interfaces in the HSRP group. This address is also called the HSRP address or the standby address. The standby address is the one that must be configured as default gateway for hosts in the subnet. The HSRP will always have one active router providing routing services in the HSRP address.

To configure the HSRP address use the following interface configuration command:

```
Switch(config-if)# standby group ip ip-address [secondary]
```

The *ip-address* must be from the excluded range of addresses for the subnet in the DHCP server.

It is very important to have in mind that both the routers physical interface IP address and the virtual router / HSRP address must be in the same IP subnet.

HSRP defines a special MAC address for the virtual router. This is necessary for hosts communicating with the virtual router in the subnet. The MAC address is 0000.0c07.acxx, where xx is the two digit hex value for the group number that the network administrator chose for this HSRP group. For example, group 15 would have the MAC address 0000.0c07.ac0f. 0f =15.

The following is a sample configuration where we use VLAN 10 as a SVI and then configure it to participate in the HSRP group 1, and assign an IP address in the same subnet to the group. We use preempt to allow this particular interface to become the active router if it is the one with the highest priority in the group.

```
Switch1 (config)# interface vlan 10
Switch1 (config-if)# ip address 192.168.1.5 255.255.255.0
Switch1 (config-if)# standby 1 priority 200
Switch1 (config-if)# standby 1 preempt
Switch1(config-if)# standby 1 ip 192.168.1.1
```

Load Balancing with HSRP

Load balancing with one HSRP group is not possible. But there's a technique that allows you to configure load balancing using HSRP. You must configure two groups and do the following:

1. One group assigns one active router to a switch.
2. One group assigns the other active router to the switch.

Doing this, two different routers can be used as gateways out of the subnet simultaneously. Another very important step is to make each router the standby router for the group it is not the active router. In short, you have two groups; each router is active in one group and standby in the other, thus providing load balancing!!! This is a very useful trick!! Keep in mind however that all the traffic from one subnet will always go to the same switch. So if one subnet is much more heavily used than the other, than load balancing really hasn't been achieved. But it is better than forcing all VLANs to use one switch and leave the other completely in standby. In addition, as a network designer you must try to avoid having a single point of failure in the network whenever possible. This is especially true in points of exits of the subnets. HSRP successfully accomplishes this design requirement.

The following is a sample configuration of two multilayer switches:

Switch 1:

```
Switch1(config)# interface vlan 10
Switch1(config-if)# ip address 192.168.1.5 255.255.255.0
Switch1(config-if)# standby 1 priority 200
Switch1(config-if)# standby 1 preempt
Switch1(config-if)# standby 1 ip 192.168.1.1
Switch1(config-if)# standby 1 authentication PrepLogic
Switch1(config-if)# standby 2 priority 100
Switch1(config-if)# standby 2 ip 192.168.1.2
Switch1(config-if)# standby 2 authentication PrepLogic
```

Switch 2:

```
Switch2(config)# interface vlan 10
Switch2(config-if)# ip address 192.168.1.10 255.255.255.0
Switch2(config-if)# standby 1 priority 100
Switch2(config-if)# standby 1 ip 192.168.1.1
Switch2(config-if)# standby 1 authentication PrepLogic
Switch2(config-if)# standby 2 priority 200
Switch2(config-if)# standby 2 preempt
Switch2(config-if)# standby 2 ip 192.168.1.2
Switch2(config-if)# standby 2 authentication PrepLogic
```

To display information regarding the status of one or more HSRP groups use the following command:

```
Router# show standby [brief] [vlan vlan-id | type mod/num]
```

This command displays the groups the router is part of, the role of the interface (active or standby), the HSRP priority, hello timer and holdtime and if it can preempt the existing active router and the authentication key.

Virtual Router Redundancy Protocol (VRRP)

Virtual Router Redundancy Protocol (VRRP) is a standards-based alternative to HSRP defined in the RFC 2238. VRRP is very similar to HSRP, to the point that only slight differences in operation and terminology must be learned.

VRRP provides one redundant gateway address from a group of routers. The active router is called the master router, whereas all others are in the backup state. The master router is the one with the highest priority in the VRRP group. VRRP groups numbers range from 0 to 255 and router priorities range from 0 to 254. 100 is the default, just like HSRP.

Very similarly to HSRP, the virtual router MAC address is in the form of 000.0e00.11xx, where xx is the two digit hex VRRP group number.

VRRP advertisements are sent at 1 second intervals. Backup routers optionally can learn the advertisement interval from the master router.

The VRRP routers are configured to preempt the master router if they are configured with a higher priority.

VRRP routers don't have any mechanism to track interfaces to allow other more capable routers to become the master router (the **track** keyword present in HSRP is not available in VRRP).

To assign a VRRP router priority use the following interface configuration command:

```
Switch(config-if)# vrrp group priority level
```

You can alter the advertisement timer (the HSRP equivalent of the hello timer) with the following command:

```
Switch(config-if)# vrrp group timers advertise [msec] interval
```

You can also configure the VRRP router to learn the advertisement timer from other VRRP peers:

```
Switch(config-if)# vrrp group timers learn
```

VRRP routers preempt the master router if they have a higher priority by default. To disable the preempting capability, use the no keyword with the following interface configuration command:

```
Switch(config-if)# [no] vrrp group preempt [delay seconds]
```

Use the delay keyword followed by a value in seconds to change the preempt delay. The default is 0 seconds. That means the preemption will take place immediately after a VRRP router with a higher priority joins the group.

Authentication is also very similar to the one provided for HSRP, and very simple to configure. Use the following interface configuration command (remember the string value must be the same in all VRRP routers in the group):

```
Switch(config-if)# vrrp group authentication string
```

Just like in HSRP, you can configure quasi-load balancing in VRRP, using the same "trick" we used in the HSRP configuration.

Switch 1:

```
Switch1(config)# interface vlan 10
Switch1(config-if)# ip address 192.168.1.5 255.255.255.0
Switch1(config-if)# vrrp 1 priority 200
Switch1(config-if)# no vrrp 1 preempt
Switch1(config-if)# vrrp 1 ip 192.168.1.1
Switch1(config-if)# vrrp 1 authentication PrepLogic
Switch1(config-if)# vrrp 2 priority 100
Switch1(config-if)# vrrp 2 ip 192.168.1.2
Switch1(config-if)# vrrp 2 authentication PrepLogic
```

Switch 2:

```
Switch2(config)# interface vlan 10
Switch2(config-if)# ip address 192.168.1.10 255.255.255.0
Switch2(config-if)# vrrp 1 priority 100
Switch2(config-if)# vrrp 1 ip 192.168.1.1
Switch2(config-if)# vrrp 1 authentication PrepLogic
Switch2(config-if)# vrrp 2 priority 200
Switch2(config-if)# no vrrp 2 preempt
Switch2(config-if)# vrrp 2 ip 192.168.1.2
Switch2(config-if)# vrrp 2 authentication PrepLogic
```

As we mentioned, VRRP is an alternative, standards-based high availability protocol to HSRP. This means that if your network pairs use a mix of Cisco and a different vendor's route/switch gear, you must use the open standard VRRP.

You can display information about the VRRP status with the following show command:

```
Switch(config-if)# show vrrp [brief]
```

The following is the actual output of the two switches configured with the previous VRRP configuration:

```
Switch1# show vrrp brief
Interface   Grp Pri Time    Own Pre     State       Master addr     Group addr
Vlan10      1   200 118     Y           Master      192.168.1.5     192.168.1.1
Vlan10      2   100 232                 Backup      192.168.1.10    192.168.1.2
Switch1#
```

Figure 35: VRRP Status on SWITCH1

```
Switch2# show vrrp brief
Interface   Grp Pri Time    Own Pre     State       Master addr     Group addr
Vlan10      1   100 235     Y           Backup      192.168.1.10    192.168.1.2
Vlan10      2   200 329                 Master      192.168.1.5     192.168.1.1
Switch2#
```

Figure 36: VRRP Status on SWITCH2

Gateway Load Balancing Protocol (GLBP)

Configuring Load Balancing using HSRP or VRRP can be considered labor intensive. Gateway Load Balancing Protocol is a Cisco Proprietary protocol developed to expand on the limitations of previous High Availability methods, HSRP and VRRP.

Some of the concepts in GLBP are the same a HSRP or VRRP, defined with different terminology.

GLBP is much more powerful and a big step forward from previous technologies.

Just like with previous technologies, GLBP assigns several routers to a common group and uses a virtual router to provide gateway/routing services to the VLAN / IP subnet. Differently than HSRP or VRRP, all routers in the group can actively participate in packet forwarding at the same time, providing true load balancing by forwarding a portion of the total traffic. This helps to better distribute actual traffic loads across two paths.

This is achieved by the way GLBP operates and assigns the virtual router's MAC address. Every time a host sends an ARP request, GLBP replies with the MAC address of a selected router in the group. This allows GLBP to use the same IP address to forward packets, but actually using several routers in the group.

GLBP is Cisco proprietary and is not supported in older routers and MLS that cannot support multiple MAC addresses on the physical interfaces. Always remember that the only high availability protocol available for networks with multivendor routers and MLS is VRRP.

Active Virtual Gateway

The Active Virtual Gateway (AVG) is the router in the GLBP group with the highest priority or, if the priority hasn't been configured, the one with the highest IP address. The AVG assigns virtual MAC addresses to up to four routers participating in the group. It also replies ARP requests to clients, and sends the corresponding virtual MAC address, depending on which algorithm is in use. Each of routers with an assigned virtual MAC address is called an Active Virtual Forwarder (AVF). AVFs are the ones that will actually forward packets out of the subnet. The rest of the routers in the group are secondary virtual forwarders and act as backups in case an AVF fails.

GLBP configuration is similar to HSRP and VRRP. To assign a router priority in a group, use the following command:

```
Switch(config-if)# glbp group priority level
```

The group value can range from 0 to 1023 and the priority from 1 to 255. The default priority is 100.

Routers or MLS in a GLBP group monitor the group's activity through the exchange of hello messages. If an MLS or router doesn't receive the hello message from a peer they assume the peer is not available and take its role. The peer only assumes the other peer is down if it doesn't receive the hello message within the *holdtime*. To configure the timers in the GLBP group use the following command:

```
Switch(config-if)# glbp group timers [msec] hellotime [msec] holdtime
```

Just like in HSRP the active role cannot be preempted by default, the active virtual forwarder cannot be preempted in GLBP by default. To allow the AVG to be preempted and to set a time interval before preempting begins, use the following command:

```
Switch(config-if)# glbp group preempt [delay minimum seconds]
```

You don't need to configure the timers in all GLBP peers. Configure them in the AVG and it will advertise the values to all peers in its group.

Active Virtual Forwarders (AVFs)

Any router participating in the GLBP group can become an AVF if the Active Virtual Gateway assigns it that role and assigns it a virtual MAC address.

The virtual MAC address always uses the 0007.b4xx.xxyy MAC address. The xxxxyy value is six zero bits followed by the 10 bit GLBP group number. The 8 bit yy value is the Active Virtual Forwarder number.

The AVG expects to hear from the AVFs hello messages. When the hellos are not received in the interval specified in the *holdtime*, the AVG assigns the AVF role to another router participating in the GLBP group.

The AVG can assign two different virtual MAC addresses to one AVF, and have it effectively assume the role of two AVFs. This might be counterproductive for obvious reasons. The AVG uses two timers to solve this issue:

1. The redirect timer determines how long the AVG will respond to ARP requests with the old virtual MAC address. The AVF that was originally assigned the virtual MAC continues to act as the gateway for any clients that try to use it.
2. The timeout timer determines when the old virtual MAC address and the AVF will be flushed from all GLBP peers. When the timeout timer expires the AVG assumes the AVF won't return to service. At that point clients using that AVF as a gateway will have to flush their ARP table entries and send a new ARP request to learn a new MAC to be used with the virtual router's IP address.

Use the following interface configuration command to configure both timers:

```
Switch(config-if)# glbp group timers redirect redirect timeout
```

GLBP can also use a weight function to determine which AVF will take a virtual MAC address in a group. Each router begins with a maximum weight value, and as interfaces go down, the weight is decreased by a certain configured amount.

GLBP uses a configurable threshold to define when a router can or cannot become an AVF, based on the weight values. If certain interfaces go down and the weight of the AVF goes down a certain configured value, the router must give up its AVF role. If interfaces come up and the router weight values are increased above the minimum threshold, the router can resume its AVF function.

A router cannot preempt another AVF is it has a lower weight value.

By default, GLBP router weight value is 100. You can configure the router to track certain interfaces to decrease the weight value if the line protocol or IP routing features go down. Use the following interface configuration command:

```
Switch(config)# track object-number interface type mod/num {line-
protocol | ip routing}
```

The **object-number** parameter takes an arbitrary value from 1 to 500 that is used for weight adjustment. The command evaluates if either the line protocol is down or the IP routing (IP routing is enabled, the interface has an IP address, and the interface is up).

To define the threshold that establishes when a router can or cannot be an AVF, use the following command:

```
Switch(config-if)# glbp group weighting maximum [lower lower] [upper upper]
```

The final step is to configure GLBP to know what objects to track and the amount it must decreased the weight value when the object (interface) goes down. Use the following command:

```
Switch(config-if)# glbp group weighting track object-number [decrement value]
```

As you can see, the decrement value is optional. If no value is specified, the default value taken is 10. The range is 1 to 254.

GLBP Load Balancing

There are three load balancing methods that can be used with GLBP to distribute traffic between redundant routers/switches:

- **Round Robin**: Each participating router interface is used sequentially. This is achieved by replying to the ARP request with one participating interface's MAC address. Round Robin is the default load balancing method in GLBP.
- **Weighted**: The interface weight is used to determine the proportion of ARP replies the AVF will handle. When interface tracking is not configured the maximum weighting value configured is used to set the relative proportions among AVFs.
- **Host Dependent**: Each host is assigned certain AVF and keeps it all the time. The method is useful when hosts are required to use the same point of exit out of networks.

To configure load balancing in GLBP use the following interface configuration command:

```
Switch(config-if)# glbp group load-balancing [round-robin |
weighted | hostdependent]
```

Remember the round robin method is used by default.

Enabling GLBP

The first step is to assign an IP address to the GLBP group. To do that use the following interface configuration command:

```
Switch(config-if)# glbp group ip [ip-address [secondary]]
```

The IP address only needs to be configured in the AVG. You can configure it in all other participating routers, but it is not required, because they will learn it from the AVG.

The following table shows several gateway redundancy verification commands:

Task	*Command Syntax*
HSRP and VRRP	
Display HSRP status	**show standby brief**
Display HSRP on an interface	**show standby type mod/num**
Display VRRP status	**show vrrp brief all**
Display VRRP on an interface	**show vrrp interface type mod/num**
GLBP Display status of a GLBP group	**show glbp [group] [brief]**

Figure 37: Gateway Redundancy Verification Commands

Supervisor and Route Processor Redundancy

Router redundancy protocols can provide fault tolerance for default gateway addresses, but they cannot address the connectivity problems that arise for the directly connected devices on the failed router. If the routing or switching engine fail, it is likely that packets won't be switched. Cisco multilayer switches have the capability to provide redundancy in this situation by having redundant hardware, ready to take the place of a failed engine.

Redundant Switch Supervisors

Certain switch platforms support the use of two or more supervisor modules installed in a single chassis. The first module to boot up becomes the active supervisor for the chassis, while the other stays idle in the standby role, waiting for the active supervisor to fail.

There are three router redundancy modes on Catalyst switches:

- **Route Processor Redundancy** (**RPR**): Only a few features of the redundant router processor are booted and initialized and if the active module fails, this redundant, standby module and all other switch modules are reloaded and all router processor functions initialize.
- **Route Processor Redundancy plus** (**RPR+**): Very similar to RPR but here the main difference is that the redundant Route Processor and Supervisor are fully initialized, but layer 2 and layer 3 functions are not available. The enhancement over RPR is that all switch modules are not required to reload/restart, and this allow the MLS/router to maintain the ports' states. Ports remain operational with practically zero downtime.
- **Stateful Switchover** (**SSO**): Both supervisors are fully booted and initialized. All layer 2 functions are maintained in both standby and active supervisors, and so is the running configuration. This allows the MLS to continue layer 2 switching without disruption during a failure in the active supervisor.

Configuring the Route Processor Redundancy mode:

To configure the redundancy mode use the following commands:

```
Router(config)# redundancy
Router(config-red)# mode {rpr | rpr-plus | sso}
```

When configuring the route processor redundancy mode for the first time, use the previous commands on both supervisor modules. After redundancy is enabled, changes are only required in the active supervisor. The active supervisor will keep the running configuration synchronized between both supervisors.

To verify the router processor redundancy mode and state of the supervisor modules use the following command:

```
Router# show redundancy states
```

Configuring Supervisor Synchronization

The active supervisor synchronizes its startup configuration and configuration register values with the standby supervisor. Additional information can be synchronized. Use the following command to do that:

```
Router(config)# redundancy
Router(config-red)# main-cpu
Router(config-r-mc)# auto-sync {startup-config | config-register | bootvar}
```

The first two commands in the first two lines are used to enter main-cpu configuration command. You can repeat the command in the third line if synchronization is required for more than one parameter.

Nonstop Forwarding (NSF)

Nonstop Forwarding (NSF) is a redundancy feature that can be used with stateful switchover to provide very quick rebuilding of the Routing Information Base (RIB) table after a supervisor switchover. The RIB is used to generate the FIB table for CEF which is downloaded to any switch modules or hardware that can perform CEF.

NSF requires the addition of certain Cisco proprietary parameters to both the router giving the assistance and the router receiving the assistance. Both routers must be "NSF-aware" in order to perform the operation, and they must be running compatible routing protocols.

NSF is compatible with many routing protocols including: Enhanced Interior Gateway Routing Protocol (EIGRP), Border Gateway Protocol (BGP), Open Shortest Path First (OSPF) and Intermediate system to Intermediate System (IS-IS).

Use the following commands with each of the compatible Routing protocols to enable them as "NSF-aware":

Routing Protocol	Configuration Commands
BGP	Router(config)# **router bgp** as-number Router(config-router)# **bgp graceful-restart**
EIGRP	Router(config)# **router eigrp** as-number Router(config-router)# **nsf**
OSPF	Router(config)# **router ospf** process-id Router(config-router)# **nsf**
IS-IS	Router(config)# **router isis** [tag] Router(config-router)# **nsf** [**cisco** | **ietf**] Router(config-router)# **nsf interval** [minutes] Router(config-router)# **nsf t3 {manual** [seconds] |**adjacency}** Router(config-router)# **nsf interface wait** seconds

Figure 38: NSF-Aware Routing Protocols

Enterprise Campus Network Design

Hierarchical Network Design

A campus network is an enterprise network if a portion of the computing infrastructure that provides access to network communication services is spread over a single geographic location.

The goal of the network design should be on providing a network that is optimized to known, studied or predicted traffic flows. A proper design must consider scalability for future growth and adaptability to new technologies.

One vital definition of the network design is the network segmentation.

A switch provides Layer 2 segmentation, while a router or a multilayer switch provides Layer 3 segmentation. This means switches break collision domains while routers break both collision domains and broadcast domains.

Cisco has defined a hierarchical network model that organizes the network into distinct layers of devices. This design provides a network that is easily managed, scalable and efficient.

The three layers are:

1) **The Access Layer** – This is where the switches that are closer to the users are located. These switches are the ones responsible for providing connectivity to end users. Devices in this layer should have the following capabilities:

    I. Low cost per switch port
    II. High port density
    III. Scalable uplinks to higher layers
    IV. User access functions such as Vlan memberships, traffic and protocol filtering and QoS.
    V. Resilience through redundant and loop-free uplinks to the distribution layer.

This layer is the point of entry to the network, where end users and the devices connecting them are located.

2) **The Distribution Layer –** Aggregates the access layer. Provides connectivity between buildings in the campus network. Devices in this layer are sometimes called building aggregation switches. They should have the following capabilities:

    I. Aggregation of multiple access layer switches
    II. High Layer 3 throughput for packet handling
    III. Security and policy-based connectivity functions through access lists or packet filters.
    IV. QoS features
    V. Scalable and resilient high-speed links to the core and access layers.
    VI. The distribution layer switches must be capable of processing all traffic from all connected devices.
    VII. VLANs and broadcast domains are aggregated at the distribution layer. Here routing, policing, classification and security must be provided. Switches at this layer must be able to perform multilayer switching with high throughput .

3) **Core Layer –** Connects and aggregates all distribution layer switches and devices. The core is also called the backbone, and must be able to switch packets as efficiently as possible. Devices in the core must be capable of the following:

    I. Very high throughput at Layer 3
    II. No costly (from a resource perspective, CPU or ASIC) or unnecessary packet manipulations (access lists, packet filtering).
    III. Redundancy and fault tolerance for high availability
    IV. Advanced QoS functions

Devices in the Core layer must be designed with efficiency in mind. They should be optimized for high performance packet switching. At the core, things must be kept simple and extremely efficient because the core must be capable of transporting the most data. Often times this is not the case but the capability must be there in order to handle a full network load.

There are situations when the campus network doesn't really need the functionality of each of the three layers. When the distribution and core layer switches are combined, the resulting network is known as a **collapsed core network**. This network design is often found in smaller network environments.

To maintain the organized, predictable and scalable character of the network, you can design it using a modular approach, where each layer of the hierarchical model can be broken into two basic functional blocks:

1. **The Switch Block –**Here you find switching devices from access and distribution layers. The switch block connects to the core block to provide end to end connectivity across the campus.

   The switch block contains a mix of Layer 2 and Layer 3 functionality, just as you would find in the access and distribution layer. In general, broadcast and all kinds of VLAN traffic should never pass the distribution layer. In this case, it should be contained in the switch block, never going into the core block. The size of the switch block depends on several factors. The more important are:

   - Traffic types and patterns.
   - Number of users connected to access layer switches.
   - Amount of Layer 3 switching capacity at the distribution layer.
   - Geographic boundaries of subnets or VLANs.
   - Size of spanning-tree domains.

   You can determine if the switch block is too big if you find traffic bottlenecks at the distribution layer. The congestion could be caused because of the volume of inter-VLAN traffic, intensive CPU processing, or switching times required by policy or security functions. Another easy way to determine if the switch block is too big is to verify that broadcast or multicast traffic is not creating congestion in the switch block.

   Switch block redundancy – The switch block should have at least two distribution layer switches to aggregate all access layer switches. Each access layer switch should have at least two uplinks, one to each distribution layer switch. A best practice is to keep access layer connected with Layer 2 links, while using only Layer 3 links to connect the distribution layer switches.

2. **The Core Block –** The core block connects two or more switch blocks in the campus network. This portion of the network must be as efficient and resilient as possible. All considerations must be to allow the maximum packet forwarding efficiency possible. The core block represents the network foundation and carries more traffic than any other network block or segment.

   There are two basic core blocks designs:

   - Collapsed core
   - Dual core

A **collapsed core** network is one in which the core layer is collapsed into the distribution layer. The functions usually found in the core and distributions layers are provided in this collapsed core layer by the same group of switches. This type of network is suited for small networks where the performance required is not excessive and the costs of implementing two different layers are not justified.

In the collapsed core network each access layer switch has two or more uplinks to the distribution/core layer switches. All Layer 3 subnets (VLANs) end in the distribution/core layer switch Layer 3 interfaces. It is recommended that distribution and core switches connect to each other by more than one link to provide a redundancy failover mechanism.

A **dual core** block is a core that connects two or more switch blocks providing redundancy. The collapse core topology can also connect two or more switch blocks, but it has scalability limitations.

In the past, dual core blocks were built with layer 2 links to provide the most efficient communication. Now, with the layer 3 multilayer switches' cost effectiveness and great performance, using them to build the dual core block is a recommended practice. With MLSs, the dual core block is equipped to break and stop bridging loops because the links can be Layer 3 links.

In the dual core block architecture, distribution switches have equal cost links to the core, providing redundancy and load balancing, with the ability to use both links simultaneously.

VLAN traffic ends in the distribution layer.

Core Size in the Network

The dual core consists of redundant switches and is segmented by Layer 3 devices. Path selection through redundant connections is determined by routing protocols.

A very important point to have in mind when designing the core is that core switches need to be able to handle each of the incoming distribution links at 100% capacity. If one of the redundant links were to fail, you want the other to be able to handle the full load of the server switch block. This is a case where link aggregation can come in handy.

Questions

Chapter 1

1. What IOS switch command was run to produce the following output? Mac Entries for Vlan.

1: _____ Dynamic Address Count :0Static Address Count :0Total Mac Addresses :0Mac Entries for Vlan

2: _____ Dynamic Address Count : 89Static Address Count : 0Total Mac Addresses : 89Mac Entries for Vlan 580:

3: _____ Dynamic Address Count : 600Static Address Count : 0Total Mac Addresses 600Total Mac Address Space Available: 4810

Choose the best answer.

- ❍ A. clear mac-address-table dynamic
- ❍ B. show mac-address-table vlans
- ❍ C. show mac-address-table count
- ❍ D. show mac-address-table entries

2. What does the IEEE 802.1d standard define? Choose the best answer.

- ❍ A. Ethernet
- ❍ B. Gigabit Ethernet
- ❍ C. SONET
- ❍ D. STP

3. Which of the following is NOT a way to reset the VTP configuration revision number on a switch? Choose the best answer.

- ❍ A. Change the mode to VTP transparent, and then back to either client or server.
- ❍ B. Change the VTP domain name to something different. Then change it back to what it should be.
- ❍ C. Reboot the switch.
- ❍ D. Delete the vlan.dat file and then reboot the switch.

4. Which of the following is NOT an EtherChannel configuration setting that must match all links in a bundle? Choose the best answer.

- ❍ A. Belong to the same native VLAN
- ❍ B. Identical speed and duplex settings
- ❍ C. Identical Spanning-Tree settings
- ❍ D. Channelized ports must be sequential. For example, you can aggregate ports fa0/1, fa0/2, fa0/3 and fa0/4 but you cannot aggregate fa0/1, fa0/2, fa0/3 and fa0/48.
- ❍ E. Identical trunking mode
- ❍ F. Pass the same VLANs

5. A network engineer is configuring LACP using 2 links. The priority for both links looks like this.lacp system-priority 100Which interface becomes the LACP decision maker? Choose the best answer.

- A. If the priority is the same on all links, they both can make decisions.
- B. The link with the lowest interface number becomes the LACP decision maker.
- C. The link with the highest MAC address becomes the LACP decision maker.
- D. The link with the lowest MAC address becomes the LACP decision maker.
- E. The link with the lowest interface number becomes the LACP decision maker.

6. You are troubleshooting an EtherChannel problem and want to see when changes occurred on the aggregated link. What show command should you use? Choose the best answer.

- A. Show etherchannel summary
- B. show etherchannel port
- C. show etherchannel port-channel
- D. show etherchannel detail

7. How does a switch identify BPDU's? Choose the best answer.

- A. All switches send BPDU's with a well-known destination MAC address.
- B. The BPDU is sent with a well-known multicast address.
- C. The BPDU is sent with a special VLAN tag.
- D. The BPDU is encapsulated with a special tag identifier.

8. Which of the following are the two different types of BPDU? Choose two.

- A. Used to detect layer 2 misconfigurations.
- B. Used to detect layer 1 and 2 errors on the network.
- C. Used for initial STP calculations.
- D. Used to announce changes on the network.

9. The STP bridge priority is what by default? Choose the best answer.

- A. 1
- B. 0
- C. 32,768
- D. 8,192

10. What STP port state cannot send or receive data frames but it can send and receive BPDUs? Choose the best answer.

- A. Blocking
- B. Listening
- C. Learning
- D. Disabled

11. When designing a layer 2 network that uses STP, why is it important for the network engineer to carefully choose which switch is the root bridge? Choose two.

- A. The root bridge should be an access-layer switch.
- B. A slow or unreliable switch could be elected as the root bridge.
- C. You should choose a root bridge switch that has a well defined backup switch.
- D. You should choose a switch that has both layer 2 and layer 3 capabilities.

12. What does the following command do? Spanning-tree vlan 128 root primary. Choose the best answer.

- A. Modifies the MAC address of the switch so it is more likely to become the root bridge.
- B. Modifies a switches bridge priority value to become less than the bridge priority of the current root bridge.
- C. Sets the priority of all VLANs configured on the switch to 128.
- D. Modifies the switch for VLAN 128 so that it never becomes the root bridge.

13. A network engineer wants to configure a FastEthernet link to make it more desirable to be chosen as the STP forwarding path for VLAN 3. Which command below accomplishes this goal? Choose the best answer.

- A. Spanning-tree vlan 3 cost 50
- B. Spanning-tree vlan 3 cost 10
- C. Spanning-tree vlan 3 primary
- D. spanning-tree vlan 3 root primary

14. Which two configurations can be used to prepare a switchport for a user PC? Choose two.

- A. Switch(config-if)# switchport host
- B. Switch(config-if)# switchport mode access
  Switch(config-if)# spanning-tree portfast
- C. Switch(config-if)# switchport access
- D. Switch(config-if)# switchport mode access
  Switch(config-if)# spanning-tree uplinkfast

15. What is the purpose of the STP root guard command? Choose the best answer.

- A. It controls the BPDU message information propagated from root bridges.
- B. It's a security feature to disable the uplink when a downstream switch malfunctions causing a layer 2 loop.
- C. It controls where candidate root bridges can be connected and found on a network.
- D. A feature where switch ports are immediately placed into a forwarding state as soon as the link comes up.

Chapter 2

1. A CEF-capable switch consists of two functional blocks. What are they? Choose two.

- ❍ A. Layer 3 engine
- ❍ B. TCAM
- ❍ C. Layer 2 forwarding engine
- ❍ D. Layer 3 forwarding engine

2. A switch is configured with 3 VLANs that are trunked to a single FastEthernet connection on a router. What type of InterVLAN routing setup is this? Choose the best answer.

- ❍ A. Multi-layer switch
- ❍ B. Route switch processor
- ❍ C. Router-on-a-stick
- ❍ D. Cisco Express Forwarding

Chapter 3

1. What layer of the three-tiered model is usually discouraged from any kind of packet filtering? Choose the best answer.

- ❍ A. Internet Edge
- ❍ B. Core
- ❍ C. Access
- ❍ D. Distribution

2. How is the Active Virtual Gateway (AVG) router chosen when configuring GLBP? Choose the best answer.

- ❍ A. The routers compare and choose the highest priority value first and if they are identi cal, it chooses the router with the highest IP address in the group.
- ❍ B. The routers compare and choose the highest priority value first and if they are ident- cal, it chooses the router with the highest MAC address in the group.
- ❍ C. The routers compare and choose the lowest priority value first and if they are identical, it chooses the router with the highest IP address in the group.
- ❍ D. The routers compare and choose the lowest priority value first and if they are identical, it chooses the router with the highest MAC address in the group.

3. GLBP reacts to router failures by doing what? Choose the best answer.

- ❍ A. The peer router uses it's own MAC address to respond to clients of the failed gateway.
- ❍ B. The peer router discards it's own MAC address and instead uses the downed peer's MAC address to respond to clients of the failed gateway.
- ❍ C. The peer router uses a new virtual MAC address and requests that clients clear the ARP table so the new MAC will be propagated to all devices using that gateway IP address.
- ❍ D. The peer router will accept requests destined for both its own MAC address as well as the downed peer address.

4. Which of the following correctly shows how to configure RPR+ on a compatible router with dual supervisors? Choose the best answer.

- ❍ A. Router(config)# redundancy mode rpr-plus
- ❍ B. Router(config)# redundancy
  Router(config-red)# mode rpr-plus
- ❍ C. Router(config)# redundancy
  Router(config-red)# type rpr-plus
- ❍ D. Router(config)# redundancy type rpr-plus

Chapter 4

1. Which of the following is NOT a voice VLAN tagging method? Choose the best answer.

- ❍ A. 802.1p
- ❍ B. 802.1q
- ❍ C. 802.1w
- ❍ D. Untagged

2. The DSCP value is divided into a two parts. What are they? Choose two.

- ❍ A. 3-bit class selector
- ❍ B. 3-bit checksum value
- ❍ C. 3-bit TOS value
- ❍ D. 3-bit Drop Precedence value
- ❍ E. 3-bit QoS tag value

Chapter 5

1. How does DHCP spoofing work? Choose the best answer.

- ❍ A. A rogue DHCP server sends requests to DHCP clients with the default gateway of the rogue device.
- ❍ B. A rogue DHCP client sends requests to DHCP servers with the default gateway of the rogue device.
- ❍ C. A rogue DHCP server sends requests to routers with the default gateway of the rogue device.
- ❍ D. A rogue DHCP server sends requests to switches with the default gateway of the rogue device.

2. For a VLAN hopping exploit to work on a switch, all of the following conditions must be met EXCEPT what? Choose the best answer.

- ❍ A. The attacker is connected to an access port on the switch.
- ❍ B. The VLAN that the attacker is connected to must be the native VLAN on the switch trunk.
- ❍ C. The attacker must be on a PVLAN in promiscuous mode.
- ❍ D. The switch must be configured with a dot1q trunk.

Explanations

Chapter 1

1. **Answer: C**

Explanation A. Incorrect. The command given is not valid on a Cisco switch.

Explanation B. Incorrect. The command given is not valid on a Cisco switch.

Explanation C. Correct. This command shows the number of static and dynamic addresses for each VLAN in the CAM table.

Explanation D. Incorrect. The command given is not valid on a Cisco switch.

2. **Answer: D**

Explanation A. Incorrect. Ethernet is a physical layer standard.

Explanation B. Incorrect. Gigabit Ethernet is a physical layer standard.

Explanation C. Incorrect. SONET is a physical layer standard.

Explanation D. Correct. This standard specifies spanning-tree protocol rules.

3. **Answer: C**

Explanation A. Incorrect. This is a valid method to reset the revision number. The revision number would show 0 is this action was performed on a switch.

Explanation B. Incorrect. This is a valid method to reset the revision number. The revision number would show 0 is this action was performed on a switch.

Explanation C. Correct. Rebooting the switch will not reset the revision number because this is not stored in volatile ram but either in NVRAM or flash depending on the switch type.

Explanation D. Incorrect. This is a valid method to reset the revision number. Implement VLAN based solution, given a network design and a set of requirements 20.

4. **Answer: D**

Explanation A. Incorrect. The 802.1q native VLAN must match on all links in the EtherChannel bundle otherwise frames will not be properly forwarded on the native VLANs. The EtherChannel will not function if the native VLANs do not match.

Explanation B. Incorrect. The speed and duplex settings must match on all links in the EtherChannel bundle.

Explanation C. Incorrect. The STP settings must match on all links in the EtherChannel bundle. The EtherChannel will not function if the STP is different.

Explanation D. Correct. The ports used on the switch do not need to be sequential.

Explanation E. Incorrect. The trunking mode must match on all links in the EtherChannel bundle. The EtherChannel will not function if the trunking modes are not identical.

Explanation F. Incorrect. The VLANs passed across the trunk must match on all links in the EtherChannel bundle. If the VLANs were not the same and a link in the bundle goes down, a previously allowed VLAN would stop being forwarded across the EtherChannel.

5. **Answer: D**

Explanation A. Incorrect. Two links never make decisions. One is always chosen over another.

Explanation B. Incorrect. Interface number does not factor into the decision.

Explanation C. Incorrect. The higher MAC would not be chosen.

Explanation D. Correct. The tie-breaker is lowest MAC address.

Explanation E. Incorrect. Interface number does not factor into the decision.

6. **Answer: C**

Explanation A. Incorrect. The summary command Displays a one-line summary per channel group and contains no information on changes that may have been made.

Explanation B. Incorrect. The port command shows EtherChannel port information but it does not contain information regarding changes that may have been made.

Explanation C. Correct. This command show you when EtherChannel changes occurred including timestamps of the events.

Explanation D. Incorrect. The detail command shows a large amount of information about the current state of the EtherChannel links but does not show information about changes being made.

7. **Answer: A**

Explanation A. Correct. The destination MAC is always 0180.c200.0000 which is designated on Ethernet networks to be used for BPDU transmissions.

Explanation B. Incorrect. BPDUs are not sent using multicast.

Explanation C. Incorrect. BPDUs do not have VLAN tags.

Explanation D. Incorrect. BPDUs are not tagged as the answer suggests.

8. **Answers: C, D**

Explanation A. Incorrect. This is not a BPDU type. BPDUs are used for STP and not for configuration checking.

Explanation B. Incorrect. This is not a BPDU type. BPDUs are used for STP and not for error checking.

Explanation C. Correct. BPDU's are used to compute the STP structure.

Explanation D. Correct. BPDU's are used to announce any topology changes on the L2 network.

9. **Answer: C**

Explanation A. Incorrect. The Default is not 1.

Explanation B. Incorrect. The Default is not 0.

Explanation C. Correct. The Default is 32768.

Explanation D. Incorrect. The Default is not 8,192.

10. **Answer: B**

Explanation A. Incorrect. Blocking would not send/receive BPDUs.

Explanation B. Correct. The correct STP port state is Listening.

Explanation C. Incorrect. The learning state does not perform the actions described in the question.

Explanation D. Incorrect. When a port is disabled, nothing can be sent or received.

11. **Answers: B, C**

Explanation A. Incorrect. It is likely that you will want your root bridge to be as close to the core as possible. This is because typically, these switches have more redundancy than access layer switches.

Explanation B. Correct. You want your root switch to be able to handle its responsibilities reliably.

Explanation C. Correct. You need to design your STP so that a well defined redundant switch is waiting to take over the root bridge responsibilities in case of a failure.

Explanation D. Incorrect. Because STP is a layer 2 protocol, there is no need to choose a switch that has layer 3 capabilities as this does nothing to improve STP. Implement VLAN based solution, given a network design and a set of requirements 23.

12. **Answer: B**

Explanation A. Incorrect. The command does not modify the MAC address.

Explanation B. Correct. This command is essentially a macro that modifies the priority for VLAN 128 which will let it be less than the default.

Explanation C. Incorrect. The command does not set the VLAN priority to 128.

Explanation D. Incorrect. The command does not set the VLAN priority to 128.

13. **Answer: B**

Explanation A. Incorrect. Setting the cost to 50 will not get you the desired results.

Explanation B. Correct. The default cost for a FastEthernet link is 19. You would want to set the cost to be lower than 19 to make it more desirable.

Explanation C. Incorrect. This is not a valid way to manipulate path metrics.

Explanation D. Incorrect. This command sets the switch for VLAN 3 as the root but it does not change path cost.

14. **Answers: A, B**

Explanation A. Correct. This command essentially is a macro that configures the following two commands:

```
Switch(config-if)# switchport mode access
Switch(config-if)# spanning-tree portfast
```

Explanation B. Correct. This command sets the port to not run STP and to allow the PC to connect to the network before the STP timer expires. Implement VLAN based solution, given a network design and a set of requirements 24.

Explanation C. Incorrect. This is not a valid command.

Explanation D. Incorrect. Uplink fast should not be used on switchports used by end devices.

15. **Answer: C**

Explanation A. Incorrect. The command does not control BPDU message information.

Explanation B. Incorrect. The command is not a security feature.

Explanation C. Correct. The command controls where candidate root bridges can be connected and found on a network.

Explanation D. Incorrect. This describes BPDU guard.

Chapter 2

1. **Answers: A, D**

Explanation A. Correct. This is one of the components of a CEF capable switch.

Explanation B. Incorrect. This is not one of the components of a CEF capable switch.

Explanation C. Incorrect. This is not one of the components of a CEF capable switch.

Explanation D. Correct. This is one of the components of a CEF capable switch.

2. **Answer: C**

Explanation A. Incorrect. The setup does not use a multi-layer switch.

Explanation B. Incorrect. The setup does not use a route switch processor.

Explanation C. Correct. The topology described is a router-on-a-stick setup.

Explanation D. Incorrect. CEF has nothing to do with the question.

Chapter 3

1. **Answer: B**

Explanation A. Incorrect. This is not part of the standard 3 tiered model.

Explanation B. Correct. The goal of the core is very fast and efficient packet switching. Because of this, packet filtering is done at the lower tiers.

Explanation C. Incorrect. Packet filtering is often done at this level of the three-tiered model.

Explanation D. Incorrect. Packet filtering is often done at this level of the three-tiered model.

2. **Answer: A**

Explanation A. Correct. Here is the method of choosing the AVG:

1. Highest priority
2. Highest IP address

Explanation B. Incorrect. GLBP does not use the highest MAC address when determining the AVG.

Explanation C. Incorrect. GLBP does not use the lowest priority value when determining the AVG.

Explanation D. Incorrect. GLBP does not use the highest MAC address when determining the AVG.

3. **Answer: D**

Explanation A. Incorrect. The GLBP router does not use its own MAC address to respond to clients of the failed gateway.

Explanation B. Incorrect. The peer router will not discard its own MAC address and instead uses the downed peer's MAC address to respond to clients of the failed gateway.

Explanation C. Incorrect. The peer router will not use a new virtual MAC address and requests that clients clear the ARP table so the new MAC will be propagated to all devices using that gateway IP address.

Explanation D. Correct. The peer router will respond using both it's MAC address for clients associated to it as well as the failed peer MAC address to service clients associated to the downed router.

4. **Answer: B**

Explanation A. Incorrect. This is not the correct syntax for configuring RPR+.

Explanation B. Correct. First you must enter config-red mode and then define the redundancy method between rpr, rpr-plus and sso.

Explanation C. Incorrect. This is not the correct syntax for configuring RPR+.

Explanation D. Incorrect. This is not the correct syntax for configuring RPR+.

Chapter 4

1. **Answer: C**

Explanation A. Incorrect. This is a valid tagging method.

Explanation B. Incorrect. This is a valid tagging method.

Explanation C. Correct. This is not a valid tagging method.

Explanation D. Incorrect. This is a valid tagging method.

2. **Answers: A, D**

Explanation A. Correct. The first portion of the DSCP value is a 3-bit class selector.

Explanation B. Incorrect. The DSCP value is not divided into a 3-bit checksum value.

Explanation C. Incorrect. The DSCP value is not divided into a 3-bit TOS value.

Explanation D. Correct. The second portion of the DSCP value is a 3-bit Drop Precedence value.

Explanation E. Incorrect. The DSCP value is not divided into a 3-bit QoS tag value.

Chapter 5

1. **Answer: A**

Explanation A. Correct. Once the client has a default gateway of the offending DHCP server, all traffic can be seen, recorded and manipulated.

Explanation B. Incorrect. A DHCP Spoofing attack does not originate from a client device.

Explanation C. Incorrect. A DHCP Spoofing attack goes after end devices such as PC's. Routers are not commonly configured as DHCP clients.

Explanation D. Incorrect. A DHCP Spoofing attack goes after end devices such as PC's. Switches are not commonly configured as DHCP clients.

2. **Answer: C**

Explanation A. Incorrect. This is one of the conditions that must be met.

Explanation B. Incorrect. This is one of the conditions that must be met.

Explanation C. Correct. This is not one of the conditions that must be met for VLAN hopping to work.

Explanation D. Incorrect. This is one of the conditions that must be met.