Table of Contents

[MODULE 2: SWITCHING CONCEPTS 3](#_Toc206771381)

[Frame Forwarding 3](#_Toc206771382)

[Switching in Networking 3](#_Toc206771383)

[The Switch MAC Address Table 4](#_Toc206771384)

[The Switch Learn and Forward Method 5](#_Toc206771385)

[Video - MAC Address Tables on Connected Switches 5](#_Toc206771386)

[Switching Forwarding Methods 7](#_Toc206771387)

[Store-and-Forward Switching 7](#_Toc206771388)

[Cut-Through Switching 8](#_Toc206771389)

[Activity - Switch It! 9](#_Toc206771390)

[Collision and Broadcast Domains 14](#_Toc206771391)

[Collision Domains 14](#_Toc206771392)

[Broadcast Domains 15](#_Toc206771393)

[Alleviate Network Congestion 16](#_Toc206771394)

[Check Your Understanding - Switching Domains 17](#_Toc206771395)

[MODULE 3: VLANS 18](#_Toc206771396)

[Introduction 18](#_Toc206771397)

[Overview of VLANs 18](#_Toc206771398)

[VLAN Definitions 18](#_Toc206771399)

[Benefits of a VLAN Design 20](#_Toc206771400)

[Types of VLANs 21](#_Toc206771401)

[Check Your Understanding - Overview of VLANs 23](#_Toc206771402)

[VLANs in a Multi-Switched Environment 24](#_Toc206771403)

[Defining VLAN Trunks 24](#_Toc206771404)

[Network without VLANs 25](#_Toc206771405)

[Network with VLANs 26](#_Toc206771406)

[VLAN Identification with a Tag 27](#_Toc206771407)

[Native VLANs and 802.1Q Tagging 28](#_Toc206771408)

[Voice VLAN Tagging 29](#_Toc206771409)

[Voice VLAN Verification Example 30](#_Toc206771410)

[Check Your Understanding - VLANs in a Multi-Switch Environment 31](#_Toc206771411)

[VLAN Configuration 34](#_Toc206771412)

[VLAN Ranges on Catalyst Switches 34](#_Toc206771413)

[VLAN Creation Commands 36](#_Toc206771414)

[VLAN Creation Example 36](#_Toc206771415)

[VLAN Port Assignment Commands 37](#_Toc206771416)

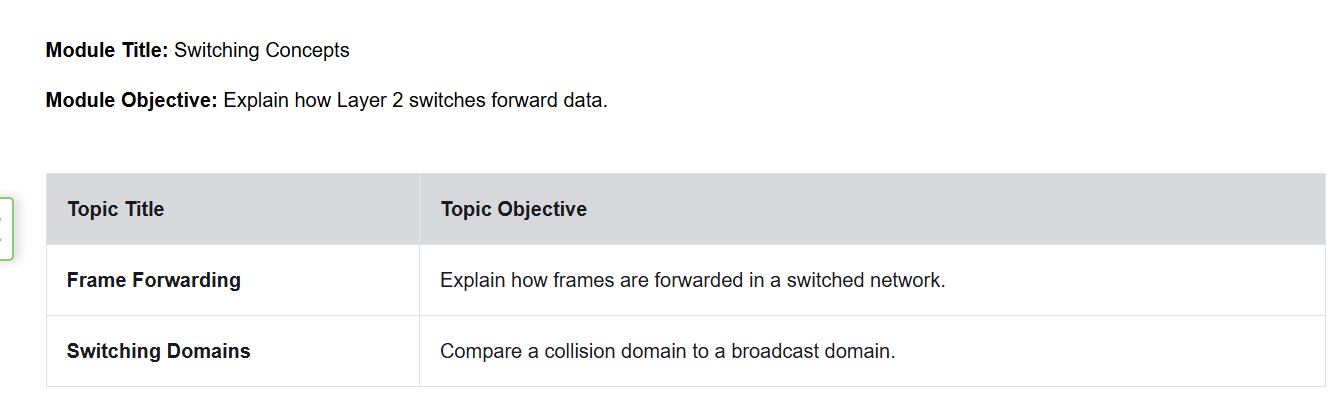
[VLAN Port Assignment Example 37](#_Toc206771417)

[Data and Voice VLANs 38](#_Toc206771418)

[Data and Voice VLAN Example 39](#_Toc206771419)

[Verify VLAN Information 40](#_Toc206771420)

# MODULE 2: SWITCHING CONCEPTS



## Frame Forwarding

### Switching in Networking

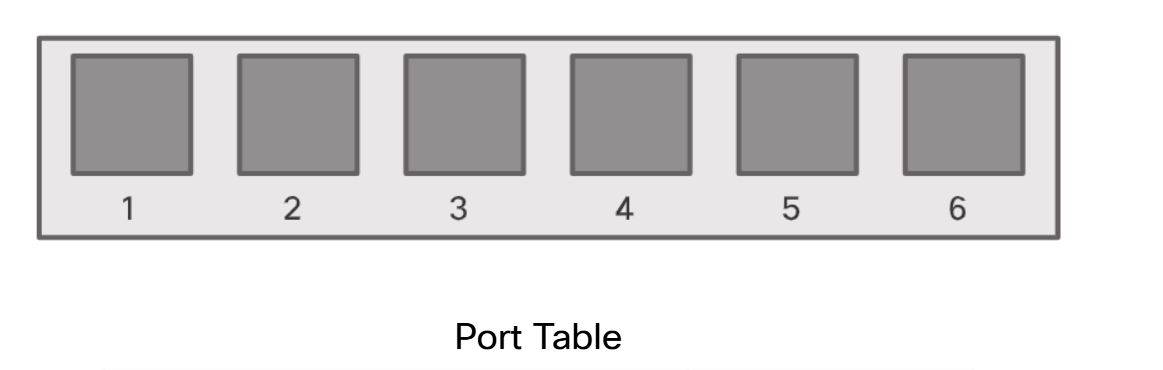
The concept of switching and forwarding frames is universal in networking and telecommunications. Various types of switches are used in LANs, WANs, and in the public switched telephone network (PSTN).

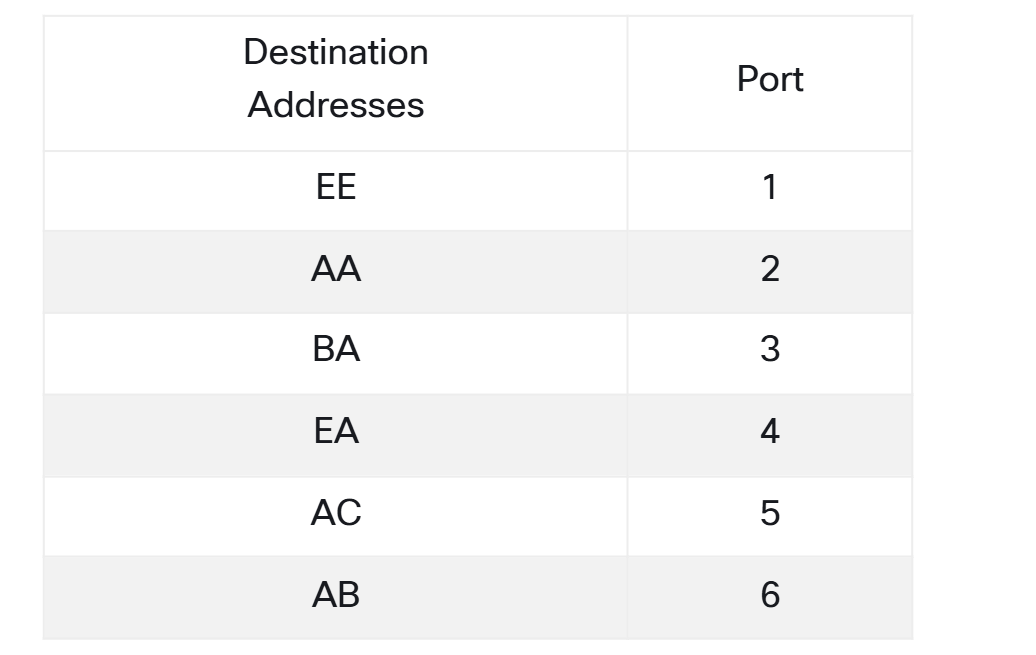
The decision on how a switch forwards traffic is made based on the flow of that traffic. There are two terms associated with frames entering and leaving an interface:

* **Ingress** - This is used to describe the port where a frame enters the device.
* **Egress** - This is used to describe the port that frames will use when leaving the device.

A LAN switch maintains a table that is referenced when forwarding traffic through the switch. The only intelligence of a LAN switch is its ability to use its table to forward traffic. A LAN switch forwards traffic based on the ingress port and the destination MAC address of an Ethernet frame. With a LAN switch, there is only one master switching table that describes a strict association between MAC addresses and ports; therefore, an Ethernet frame with a given destination address always exits the same egress port, regardless of the ingress port it enters.

**Note:** An Ethernet frame will never be forwarded out the same port it was on which it was received.

****

****

### The Switch MAC Address Table

A switch is made up of integrated circuits and the accompanying software that controls the data paths through the switch. Switches use destination MAC addresses to direct network communications through the switch, out the appropriate port, toward the destination.

For a switch to know which port to use to transmit a frame, it must first learn which devices exist on each port. As the switch learns the relationship of ports to devices, it builds a table called a MAC address table. This table is stored in content addressable memory (CAM) which is a special type of memory used in high-speed searching applications. For this reason, the MAC address table is sometimes also called the CAM table.

LAN switches determine how to handle incoming data frames by maintaining the MAC address table. A switch populates its MAC address table by recording the source MAC address of each device connected to each of its ports. The switch references the information in the MAC address table to send frames destined for a specific device out of the port which has been assigned to that device.

### The Switch Learn and Forward Method

The following two-step process is performed on every Ethernet frame that enters a switch.

**Step 1. Learn - Examining the Source MAC Address**

Every frame that enters a switch is checked for new information to learn. It does this by examining the source MAC address of the frame and port number where the frame entered the switch:

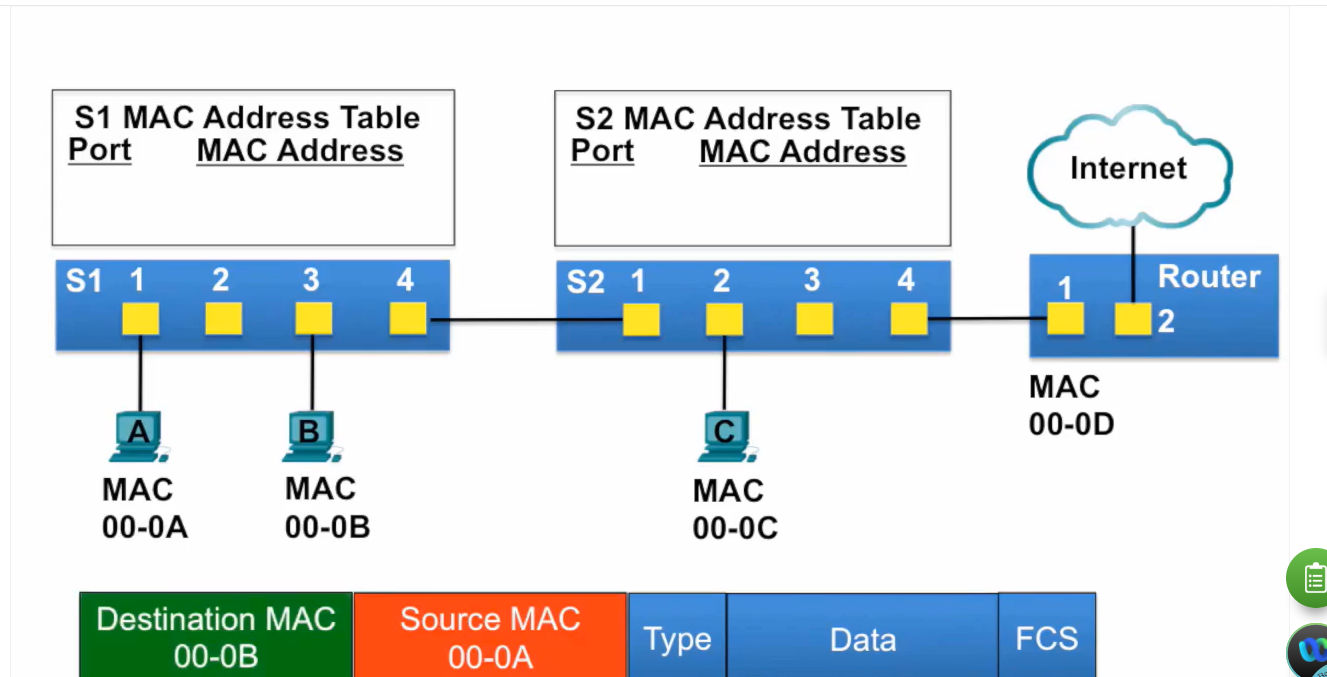
* If the source MAC address does not exist in the MAC address table, the MAC address and incoming port number are added to the table.
* If the source MAC address does exist, the switch updates the refresh timer for that entry. By default, most Ethernet switches keep an entry in the table for five minutes. If the source MAC address does exist in the table but on a different port, the switch treats this as a new entry. The entry is replaced using the same MAC address, but with the more current port number.

**Step 2. Forward - Examining the Destination MAC Address**

If the destination MAC address is a unicast address, the switch will look for a match between the destination MAC address of the frame and an entry in its MAC address table:

* If the destination MAC address is in the table, it will forward the frame out of the specified port.
* If the destination MAC address is not in the table, the switch will forward the frame out all ports except the incoming port. This is called an unknown unicast. If the destination MAC address is a broadcast or a multicast, the frame is also flooded out all ports except the incoming port.

### Video - MAC Address Tables on Connected Switches



* 00:02- In this video, PC-A is going to send
* 00:04an ethernet frame to PC-B, and
* 00:07PC-B is going to send an ethernet frame to PC-A.
* 00:10We're going to examine how switches S1 and S2
* 00:14build their MAC address tables,
* 00:16and also how they forward frames
* 00:18based on the information in those MAC address tables.
* 00:23PC-A has an ethernet frame to send to PC-B.
* 00:27The source MAC address of the frame is 00-0A
* 00:30and the destination MAC address is 00-0B.
* 00:34The ethernet frame is sent to switch S1.
* 00:38S1 receives the ethernet frame,
* 00:40examines the source MAC address
* 00:42and notices that this MAC address
* 00:43is not in its MAC address table,
* 00:46so it adds the MAC address and the incoming port number.
* 00:51Next, switch S1 examines the destination MAC address
* 00:55and notices that this MAC address is not in its table,
* 00:58so it floods it out all ports.
* 01:03PC-B receives the ethernet frame,
* 01:06examines the destination MAC address
* 01:08against its own MAC address and notices that
* 01:11that is a match and receives the rest of the frame.
* 01:16The ethernet frame continues to be
* 01:18forwarded to switch S2.
* 01:20Switch S2 examines the source MAC address
* 01:23to the frame and notices it is not in
* 01:25its MAC address table, so it adds the MAC address
* 01:29and the incoming port to its MAC address table.
* 01:33Next, switch S2 examines the destination MAC address,
* 01:36notices that is not in its MAC address table
* 01:39so it floods it out all ports.
* 01:43PC-C receives the ethernet frame
* 01:46and its MAC address does not match
* 01:49the destination MAC address of the ethernet frame,
* 01:52so it does not accept the rest of the frame.
* 01:55The router receives the ethernet frame,
* 01:58examines the destination MAC address
* 02:00against its own MAC address
* 02:02and notices it is not a match
* 02:04so it does not receive the rest of the frame.
* 02:08Now, let's have PC-B sending a frame back to PC-A.
* 02:16The source MAC address of the frame is 00-0B
* 02:20and the destination MAC address is 00-0A.
* 02:24PC-B sends it to switch S1.
* 02:27S1 notices that the source MAC address
* 02:31is not in its MAC address table
* 02:32so it adds the MAC address
* 02:34and the incoming port number.
* 02:37Next, switch S1 examines the destination MAC address
* 02:41and notices that MAC address is in its MAC address table.
* 02:46So it sends it out just port 1.
* 02:51PC-A receives the ethernet frame,
* 02:54examines the destination MAC address
* 02:56against its own MAC address and notices
* 02:59it is a match so it receives the rest of the frame.

### Switching Forwarding Methods

Switches make Layer 2 forwarding decisions very quickly. This is because of software on application-specific-integrated circuits (ASICs). ASICs reduce the frame-handling time within the device and allow the device to manage an increased number of frames without degrading performance.

Layer 2 switches use one of two methods to switch frames:

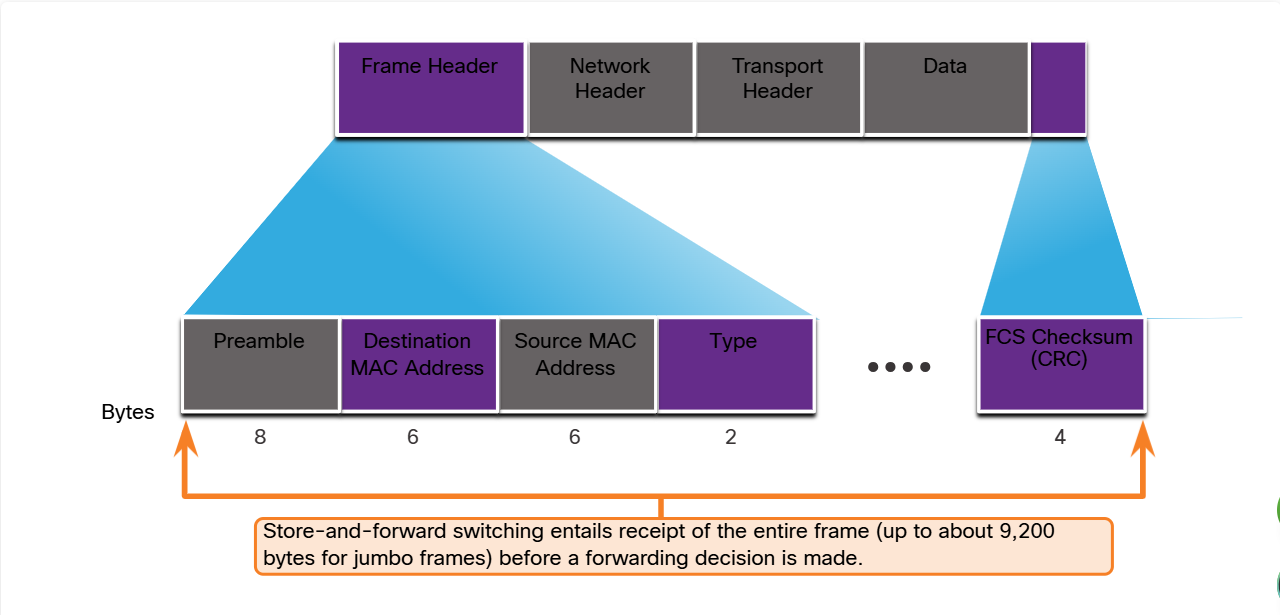
* **Store-and-forward switching** - This method makes a forwarding decision on a frame after it has received the entire frame and checked the frame for errors using a mathematical error-checking mechanism known as a cyclic redundancy check (CRC). Store-and-forward switching is Cisco’s primary LAN switching method.
* **Cut-through switching** - This method begins the forwarding process after the destination MAC address of an incoming frame and the egress port have been determined.

### Store-and-Forward Switching

Store-and-forward switching, as distinguished from cut-through switching, has the following two primary characteristics:

* **Error checking** - After receiving the entire frame on the ingress port, the switch compares the frame check sequence (FCS) value in the last field of the datagram against its own FCS calculations. The FCS is an error checking process that helps to ensure that the frame is free of physical and data-link errors. If the frame is error-free, the switch forwards the frame. Otherwise, the frame is dropped.
* **Automatic buffering** - The ingress port buffering process used by store-and-forward switches provides the flexibility to support any mix of Ethernet speeds. For example, handling an incoming frame traveling into a 100 Mbps Ethernet port that must be sent out a 1 Gbps interface would require using the store-and-forward method. With any mismatch in speeds between the ingress and egress ports, the switch stores the entire frame in a buffer, computes the FCS check, forwards it to the egress port buffer and then sends it.

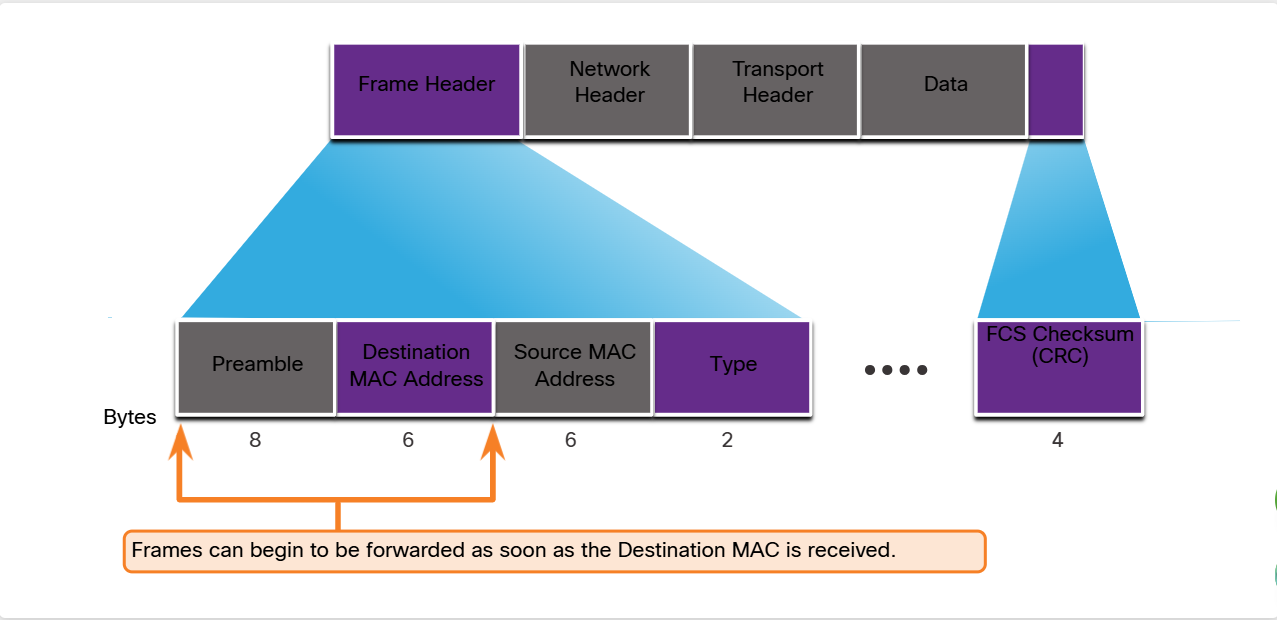
The figure illustrates how store-and-forward makes a decision based on the Ethernet frame.



### Cut-Through Switching

The store-and-forward switching method drops frames that do not pass the FCS check. Therefore, it does not forward invalid frames.

By contrast, the cut-through switching method may forward invalid frames because no FCS check is performed. However, cut-through switching has the ability to perform rapid frame switching. This means the switch can make a forwarding decision as soon as it has looked up the destination MAC address of the frame in its MAC address table, as shown in the figure.



The switch does not have to wait for the rest of the frame to enter the ingress port before making its forwarding decision.

Fragment free switching is a modified form of cut-through switching in which the switch only starts forwarding the frame. Fragment free switching provides better error checking than cut-through, with practically no increase in latency.

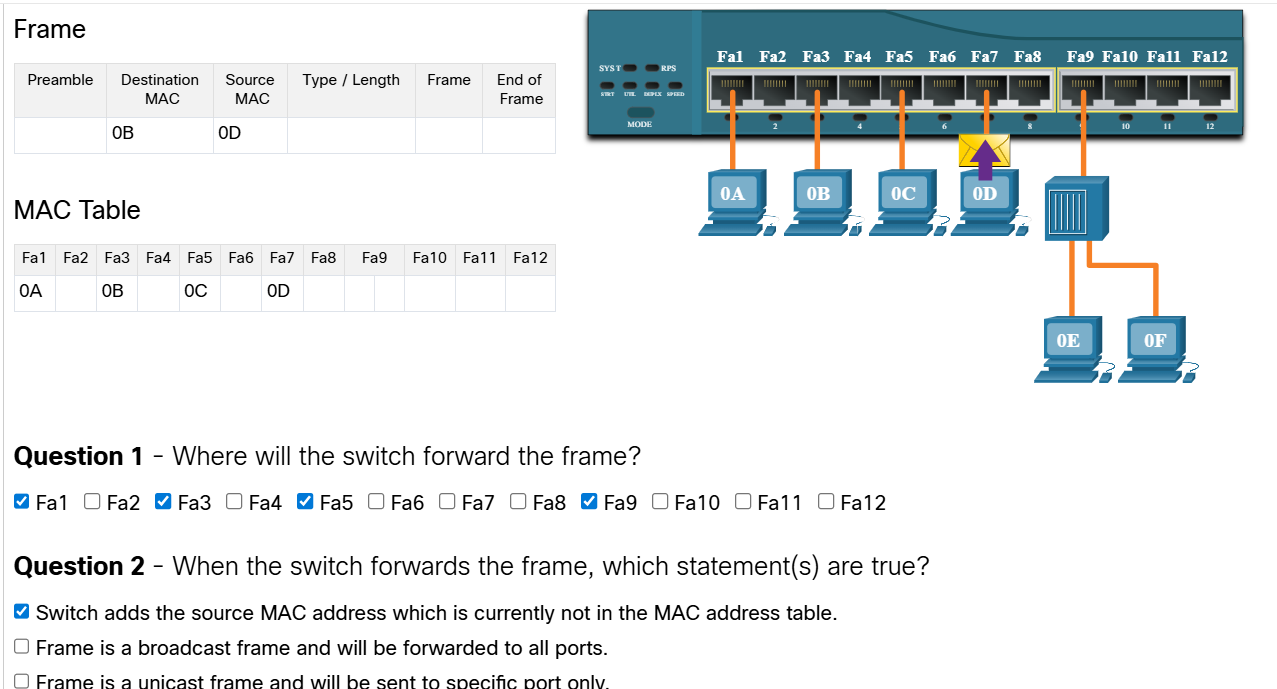
The lower latency speed of cut-through switching makes it more appropriate for extremely demanding, high-performance computing (HPC) applications that require process-to-process latencies of 10 microseconds or less.

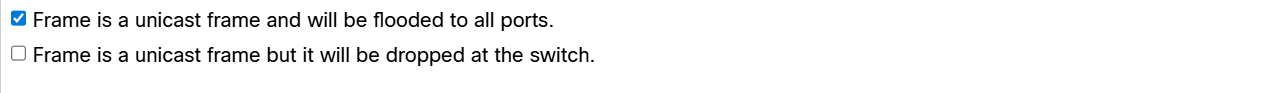
The cut-through switching method can forward frames with errors. If there is a high error rate (invalid frames) in the network, cut-through switching can have a negative impact on bandwidth, thereby clogging up bandwidth with damaged and invalid frames.

### Activity - Switch It!

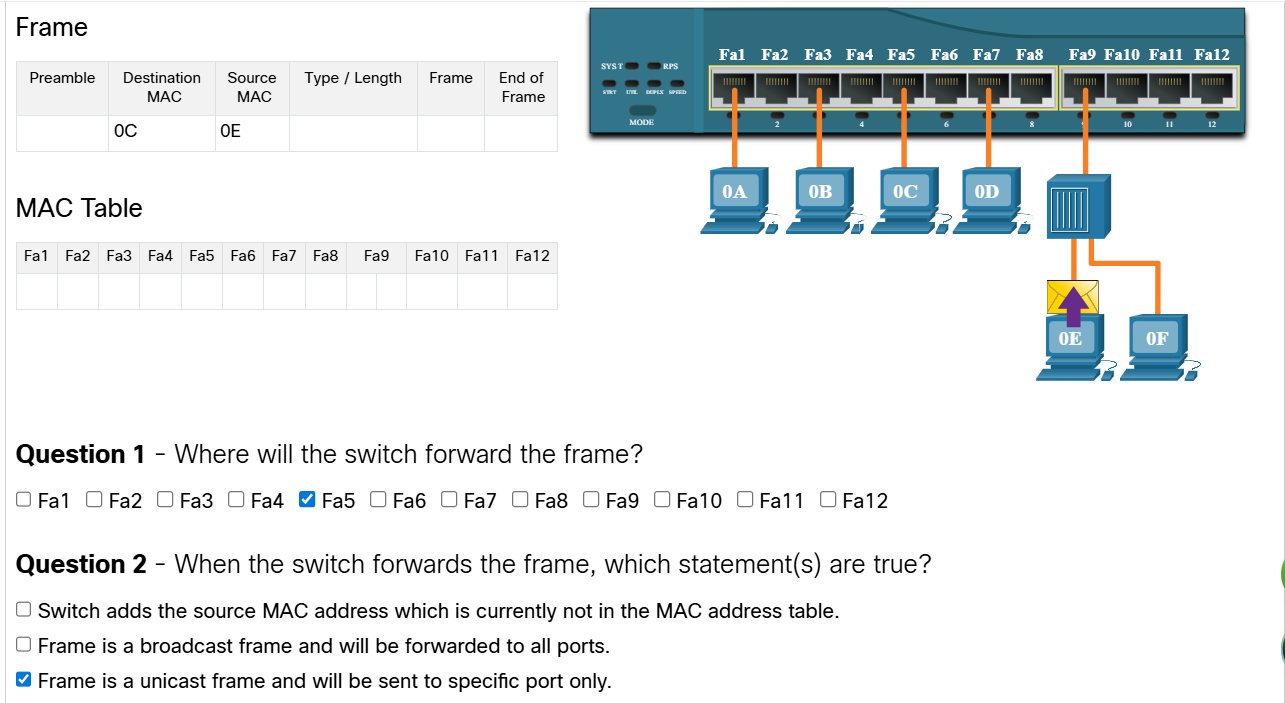
Use this activity to check your understanding of how a switch learns and forwards frames.

#### Challenge 1

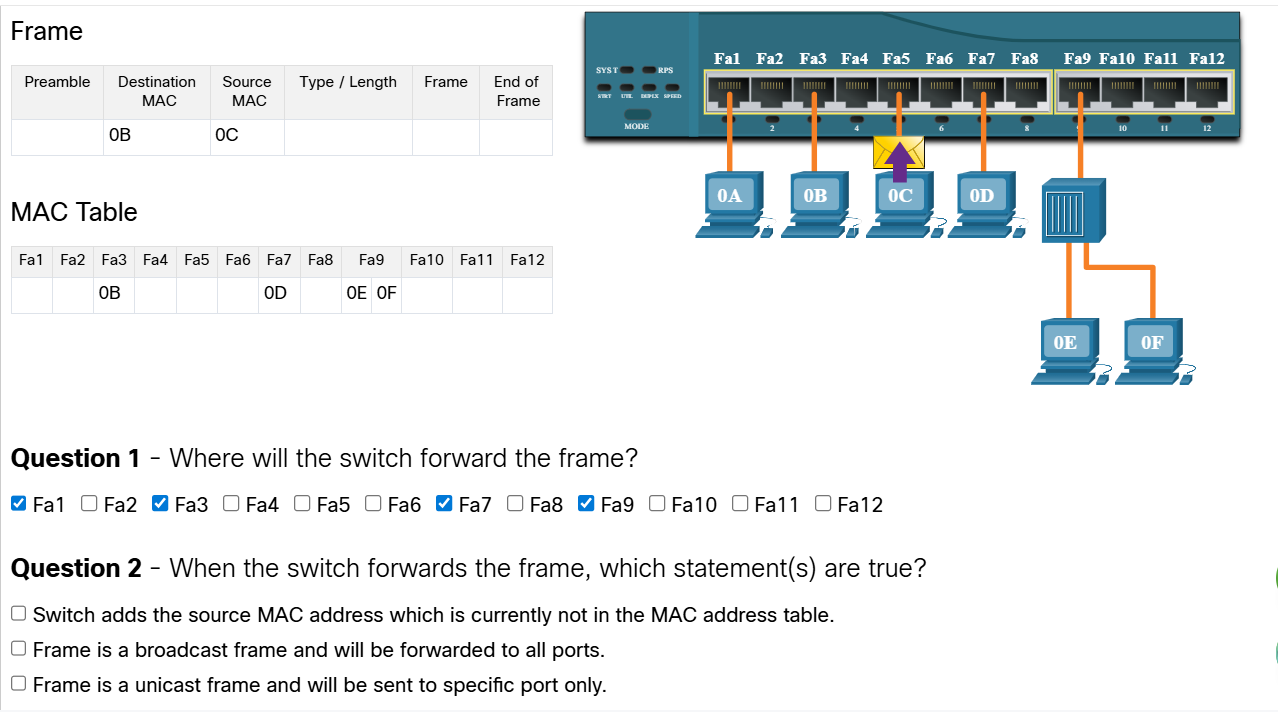




#### Challenge 2

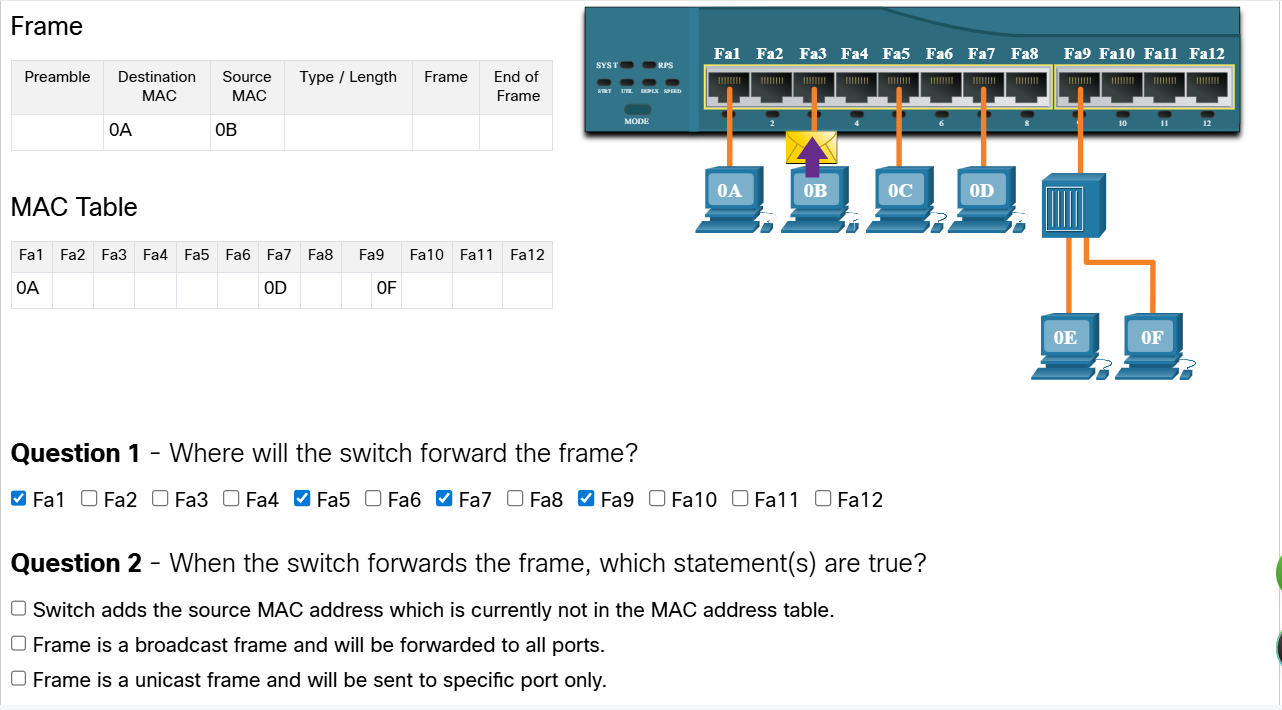


#### Challenge 3



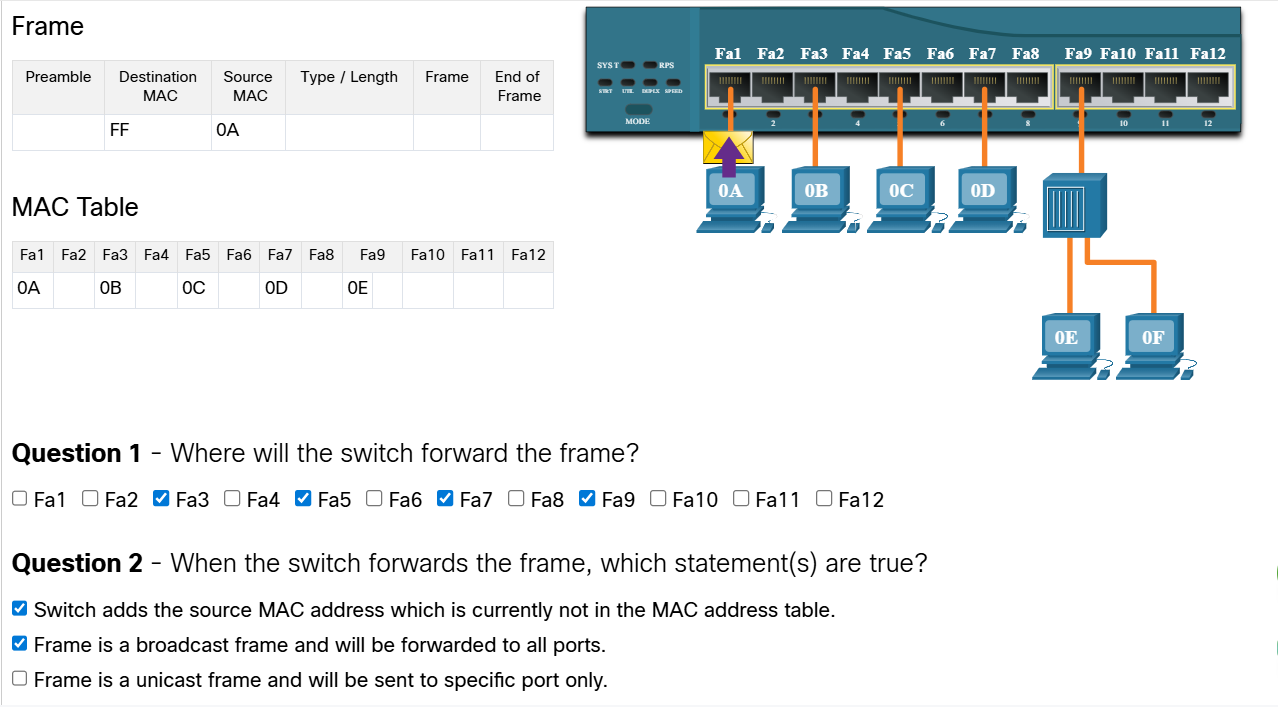


#### Challenge 4

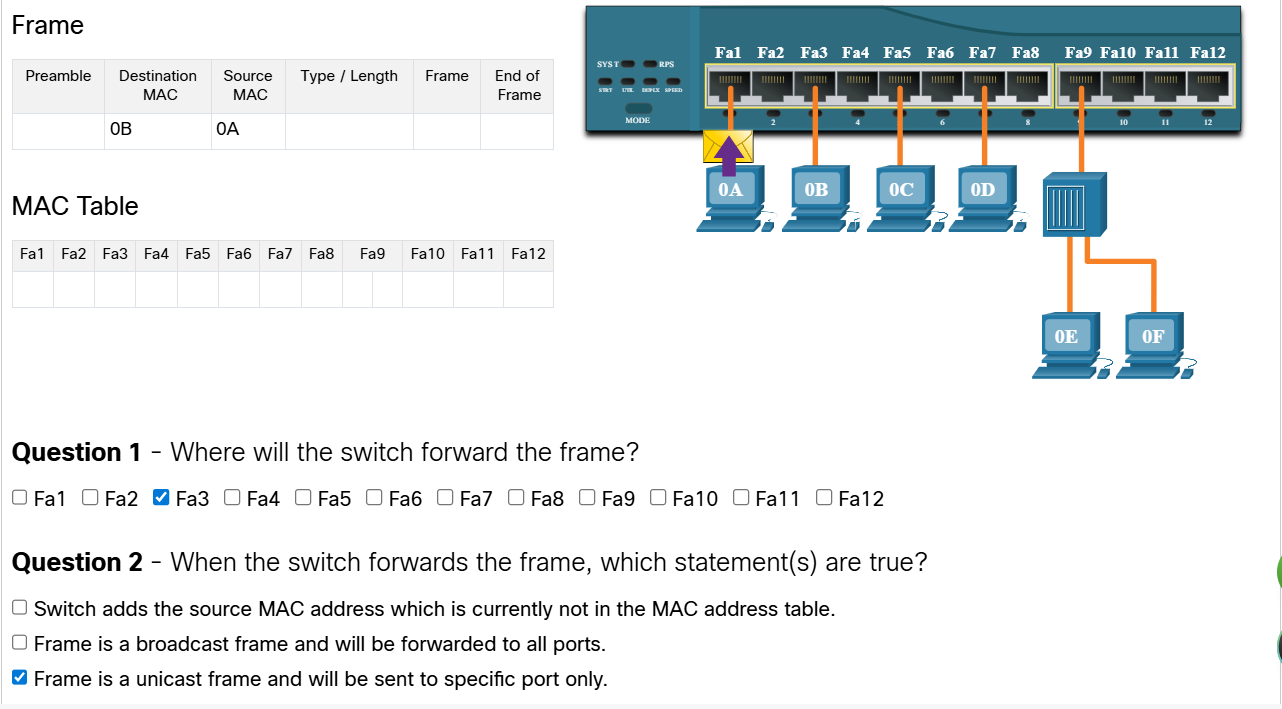




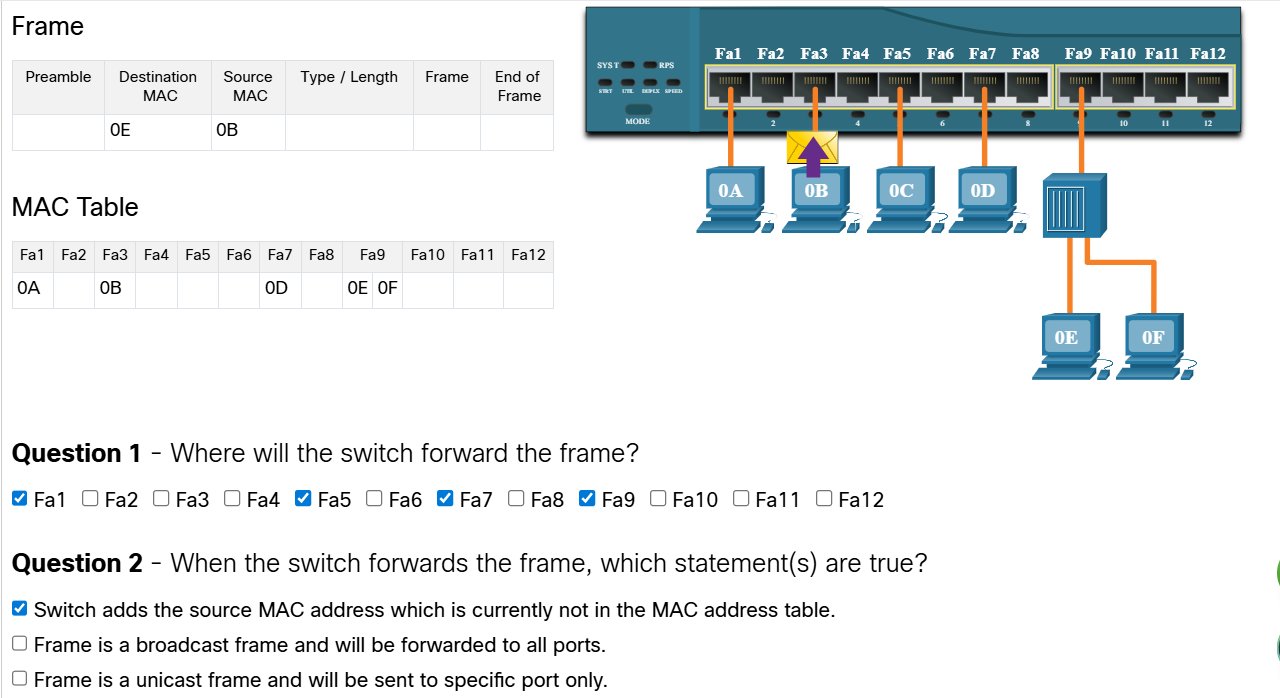
#### Challenge 5



#### Challenge 6

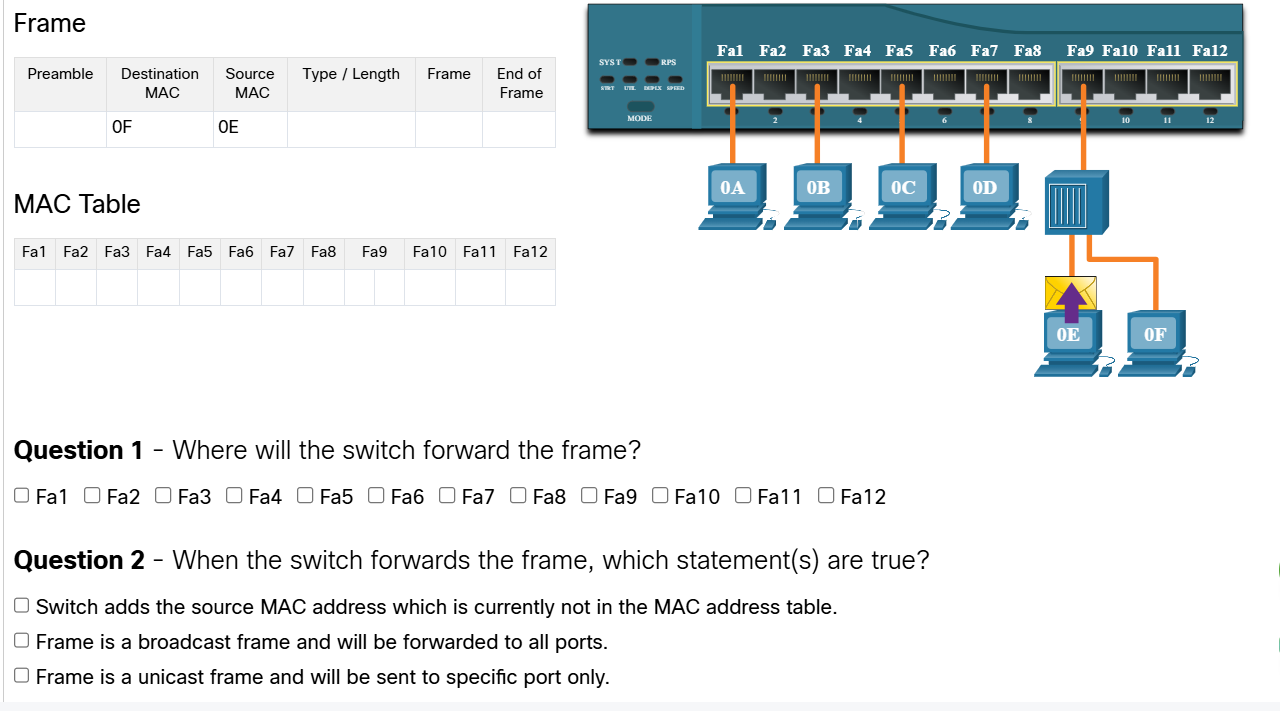


#### Challenge 7



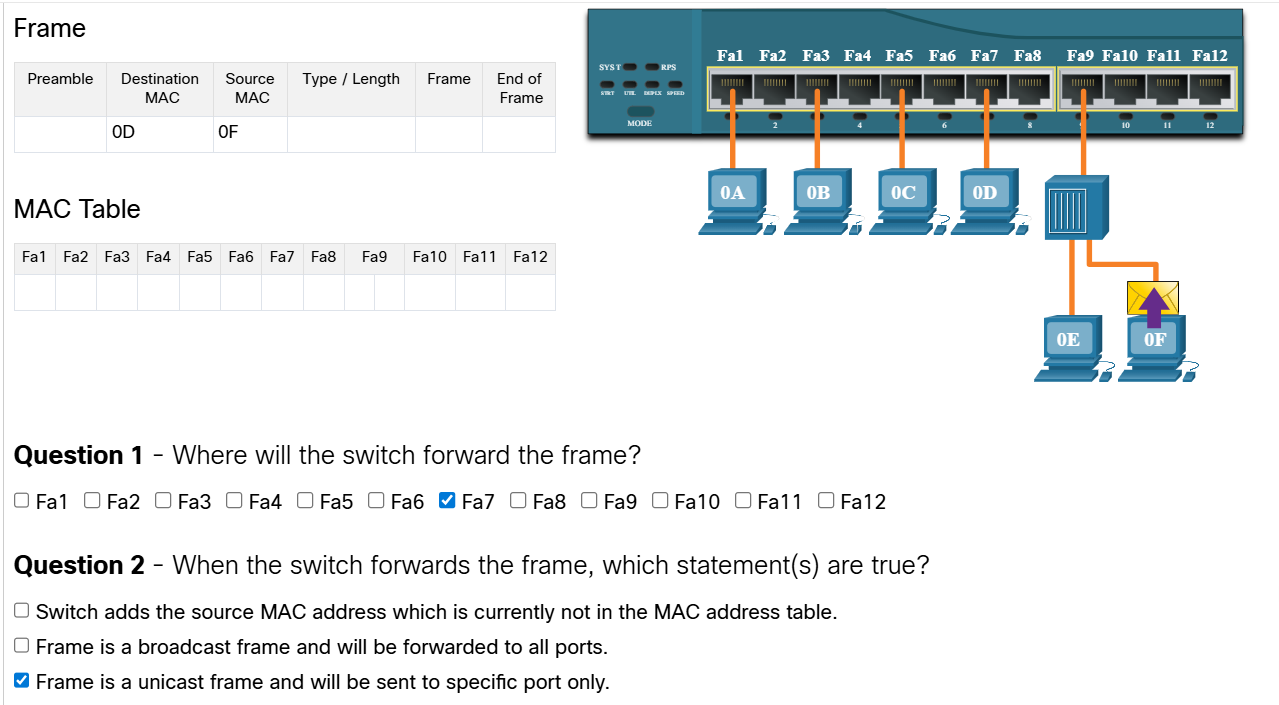


#### Challenge 8

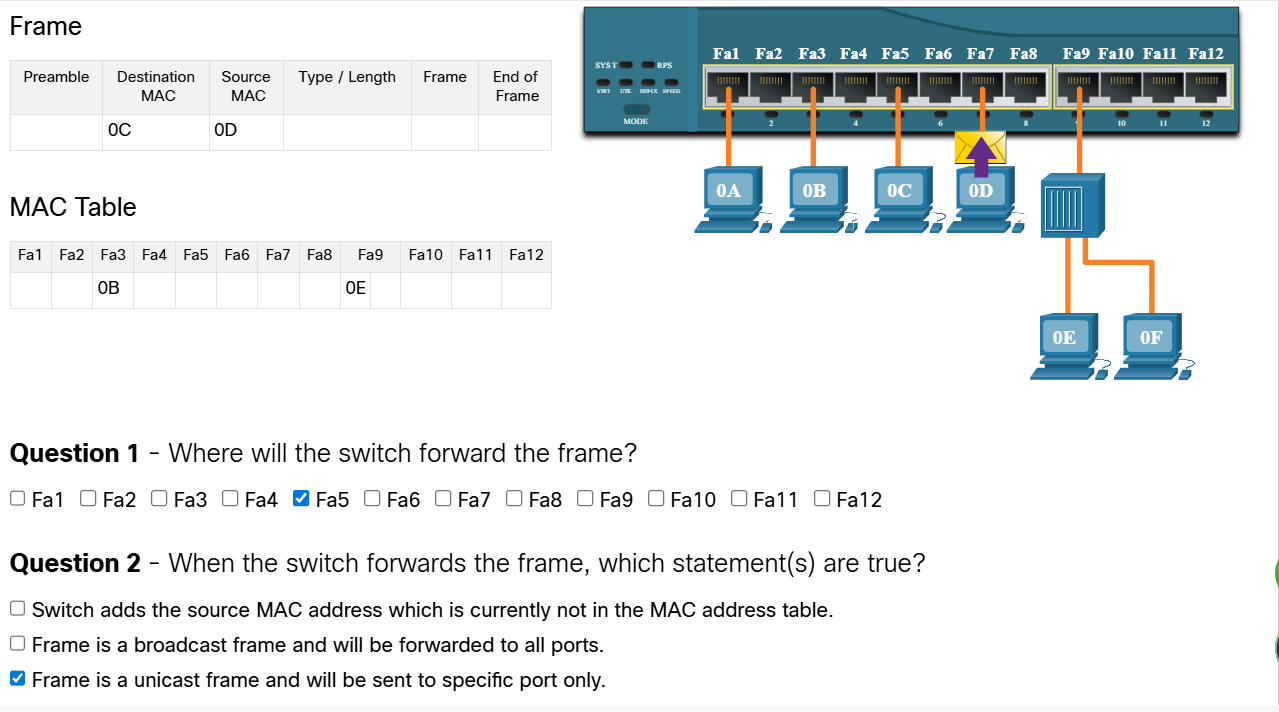




#### Challenge 9



#### Challenge 10



## Collision and Broadcast Domains

### Collision Domains

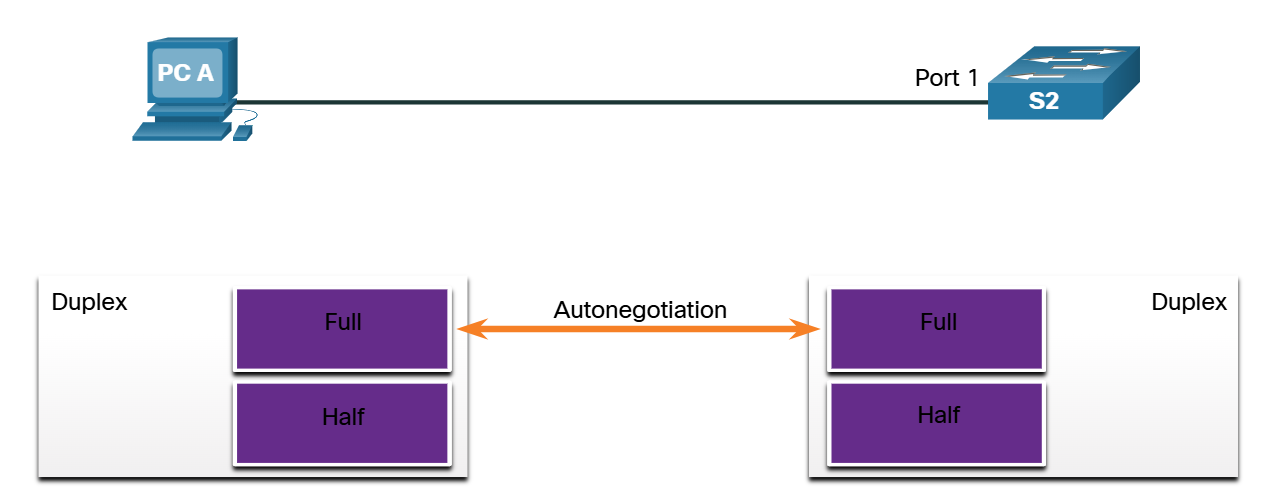
In the previous topic, you gained a better understanding of what a switch is and how it operates. This topic discusses how switches work with each other and with other devices to eliminate collisions and reduce network congestion. The terms collisions and congestion are used here in the same way that you use them in street traffic.

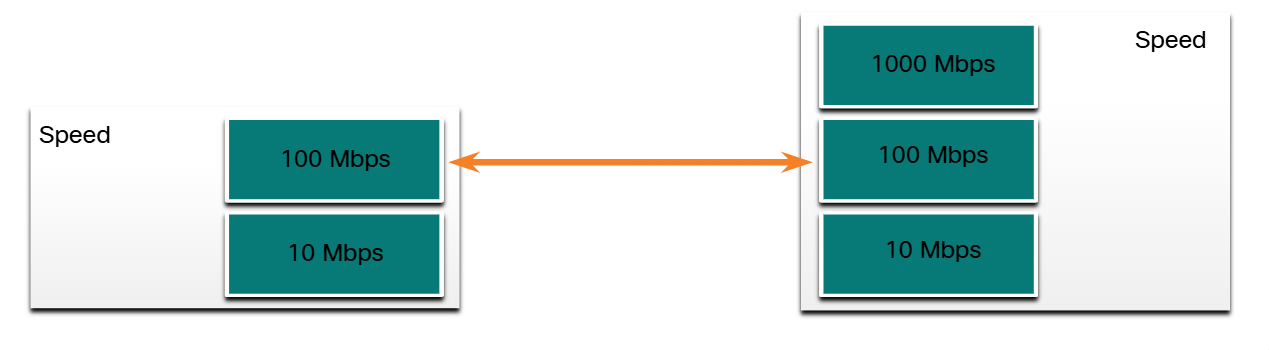
In legacy hub-based Ethernet segments, network devices competed for the shared medium. The network segments that share the same bandwidth between devices are known as collision domains. When two or more devices within the same collision domain try to communicate at the same time, a collision will occur.

If an Ethernet switch port is operating in half-duplex, each segment is in its own collision domain. There are no collisions when switch ports are operating in full-duplex. However, there could be a collision domain if a switch port is operating in half-duplex.

By default, Ethernet switch ports will autonegotiate full-duplex when the adjacent device can also operate in full-duplex. If the switch port is connected to a device operating in half-duplex, such as a legacy hub, then the switch port will operate in half-duplex. In the case of half-duplex, the switch port will be part of a collision domain.

As shown in the figure, full-duplex is chosen if both devices have the capability along with their highest common bandwidth.



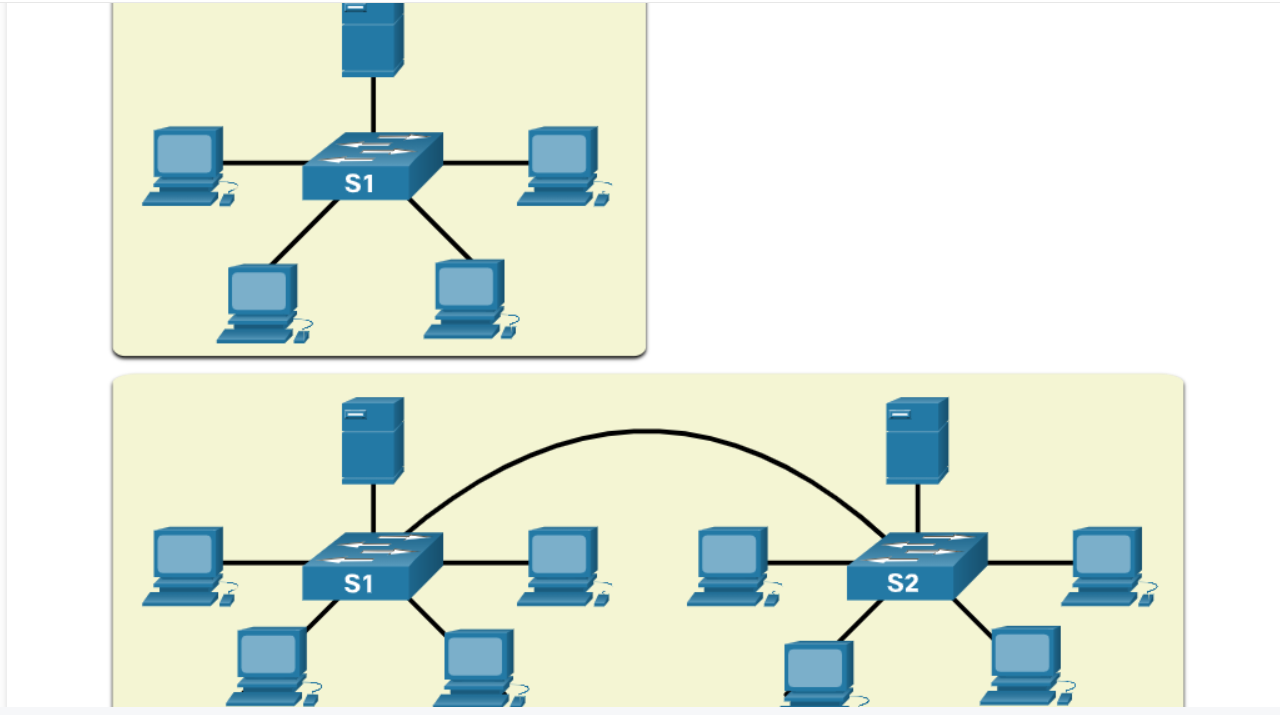


### Broadcast Domains

A collection of interconnected switches forms a single broadcast domain. Only a network layer device, such as a router, can divide a Layer 2 broadcast domain. Routers are used to segment broadcast domains, but will also segment a collision domain.

When a device sends a Layer 2 broadcast, the destination MAC address in the frame is set to all binary ones.

The Layer 2 broadcast domain is referred to as the MAC broadcast domain. The MAC broadcast domain consists of all devices on the LAN that receive broadcast frames from a host.



When a switch receives a broadcast frame, it forwards the frame out each of its ports, except the ingress port where the broadcast frame was received. Each device connected to the switch receives a copy of the broadcast frame and processes it.

Broadcasts are sometimes necessary for initially locating other devices and network services, but they also reduce network efficiency. Network bandwidth is used to propagate the broadcast traffic. Too many broadcasts and a heavy traffic load on a network can result in congestion, which slows down network performance.

When two switches are connected together, the broadcast domain is increased, as seen in the second half of the animation. In this case, a broadcast frame is forwarded to all connected ports on switch S1. Switch S1 is connected to switch S2. The frame is then also propagated to all devices connected to switch S2.

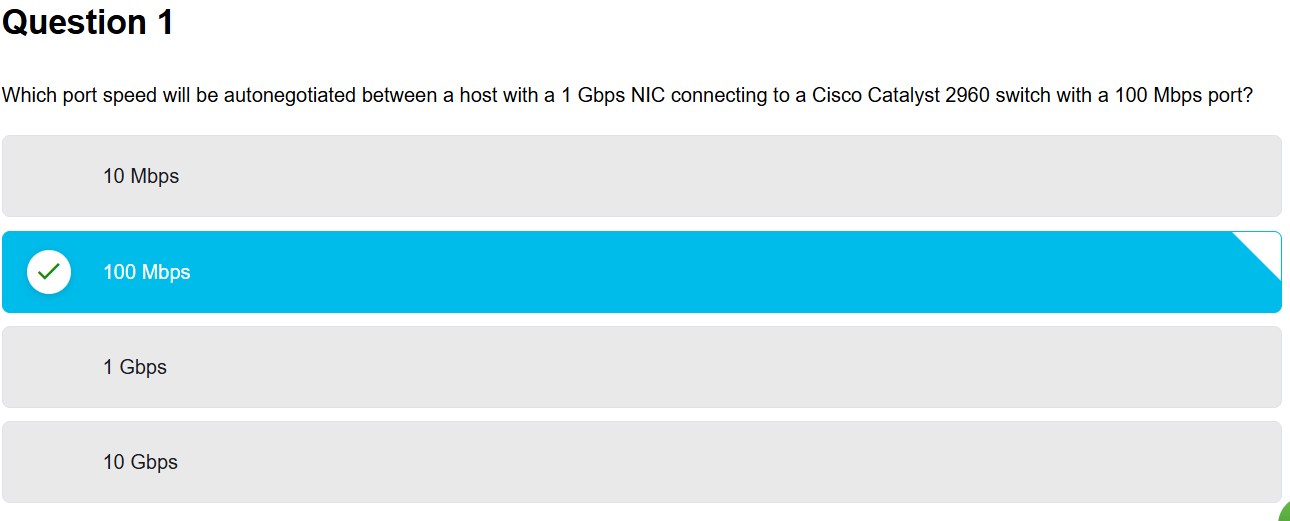
### Alleviate Network Congestion

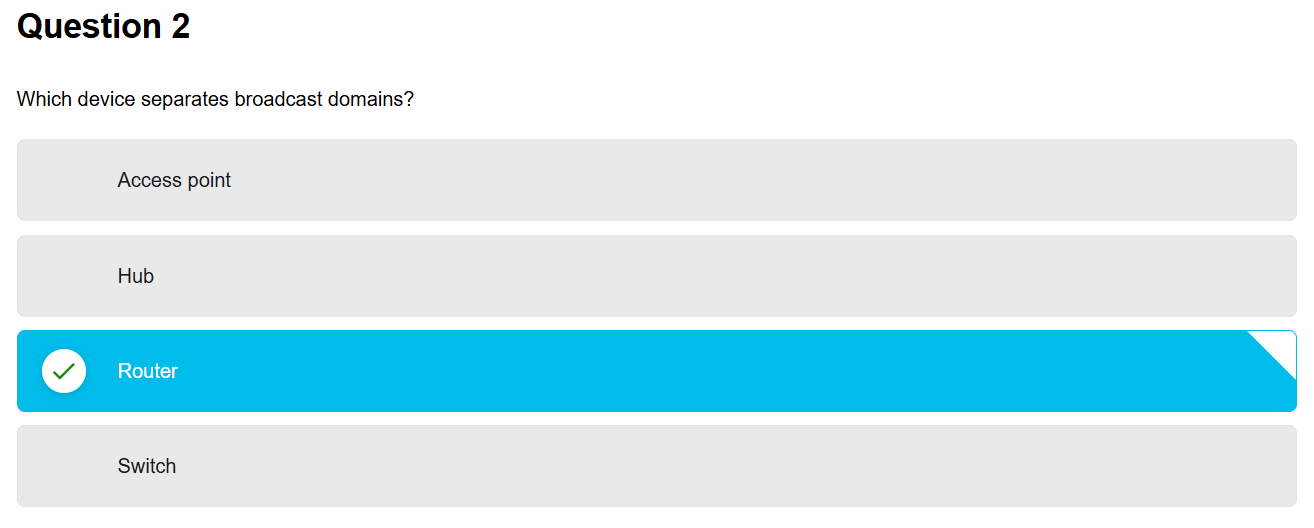
LAN switches have special characteristics that help them alleviate network congestion. By default, interconnected switch ports attempt to establish a link in full-duplex, therefore eliminating collision domains. Each full-duplex port of the switch provides the full bandwidth to the device or devices that are connected to that port. Full-duplex connections have dramatically increased LAN network performance, and are required for 1 Gbps Ethernet speeds and higher.

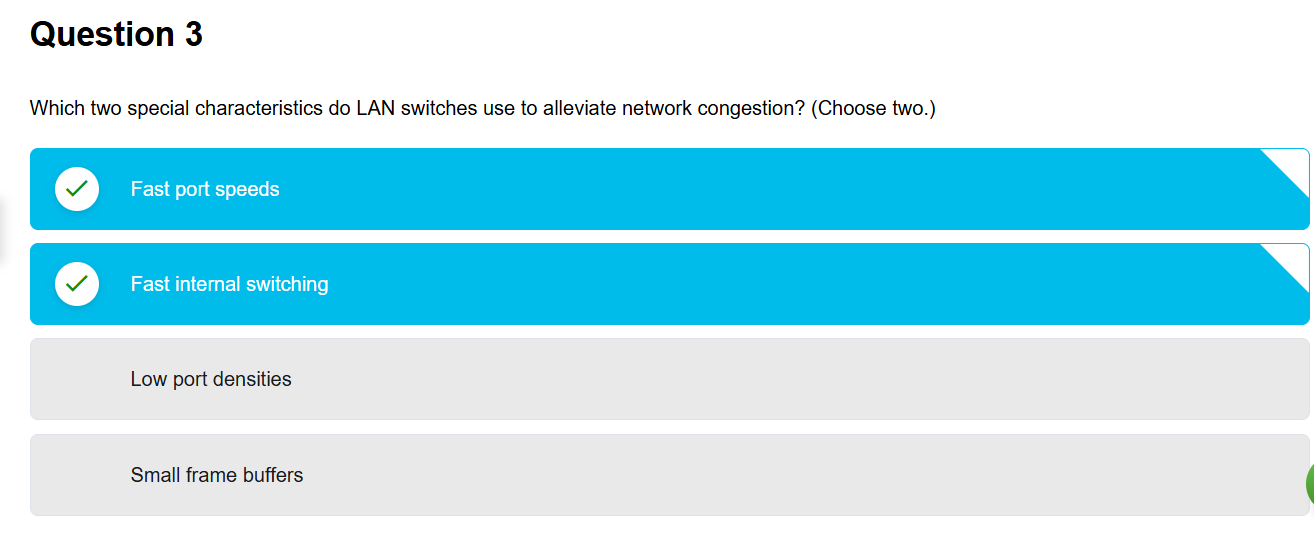
Switches interconnect LAN segments, use a MAC address table to determine egress ports, and can lessen or eliminate collisions entirely. Characteristics of switches that alleviate network congestion include the following:

* **Fast port speeds** - Ethernet switch port speeds vary by model and purpose. For instance, most access layer switches support 100 Mbps and 1 Gbps port speeds. Distribution layer switches support 100 Mbps, 1 Gbps, and 10 Gbps port speeds and core layer and data center switches may support 100 Gbps, 40 Gbps, and 10 Gbps port speeds. Switches with faster port speeds cost more but can reduce congestion.
* **Fast internal switching** - Switches use a fast internal bus or shared memory to provide high performance.
* **Large frame buffers** - Switches use large memory buffers to temporarily store more received frames before having to start dropping them. This enables ingress traffic from a faster port (e.g., 1 Gbps) to be forwarded to a slower (e.g., 100 Mbps) egress port without losing frames.
* **High port density** - A high port density switch lowers overall costs because it reduces the number of switches required. For instance, if 96 access ports were required, it would be less expensive to buy two 48-port switches instead of four 24-port switches. High port density switches also help keep traffic local, which helps alleviate congestion.

### Check Your Understanding - Switching Domains



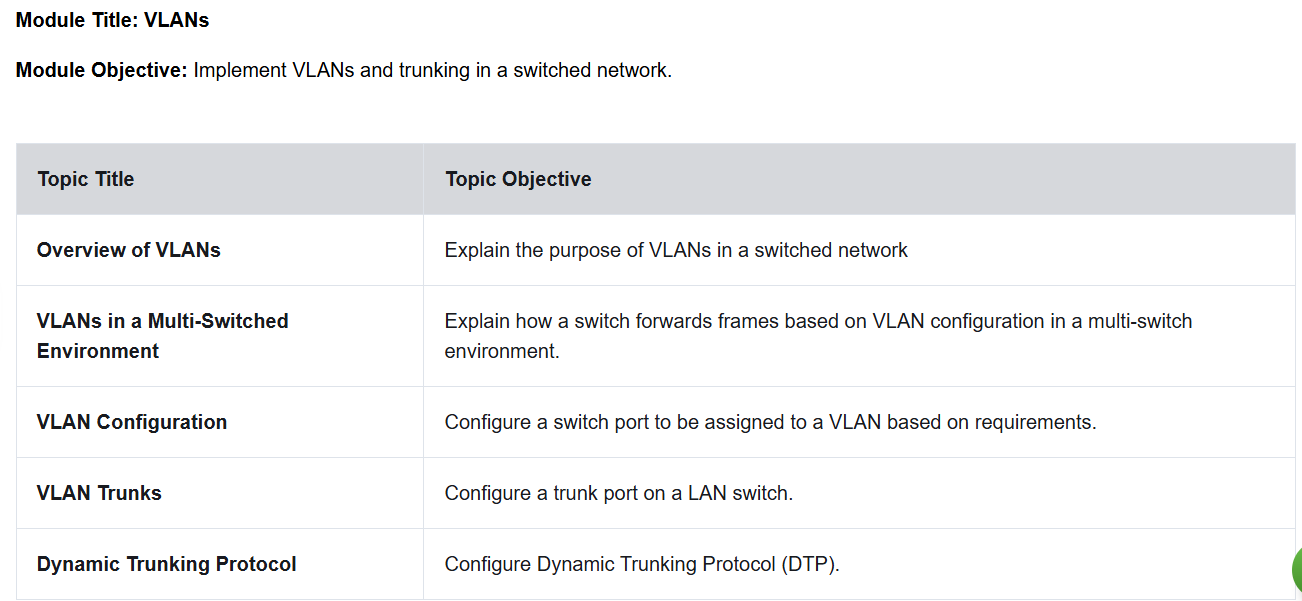




# MODULE 3: VLANS

## Introduction

VLANs are created at Layer 2 to reducing or eliminate broadcast traffic. **VLANs** are how you break up your network into smaller networks, so that the devices and people within a single VLAN are communicating with each other and not having to manage traffic from other networks. The network administrator can organize VLANs by location, who is using them, the type of device, or whatever category is needed.

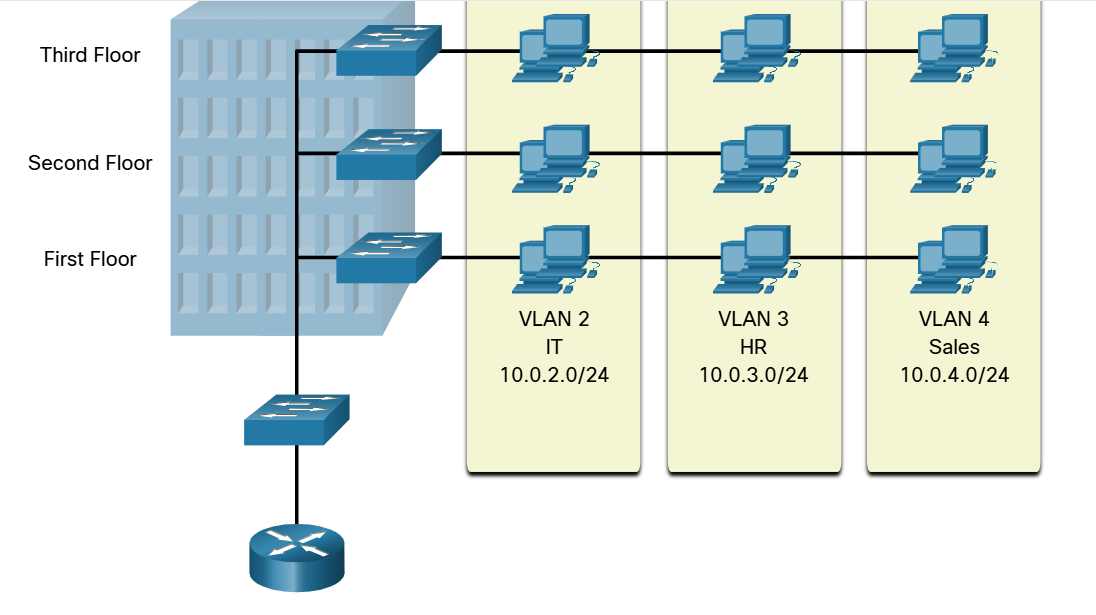


## Overview of VLANs

### VLAN Definitions

Virtual LANs (VLANs) provide segmentation and organizational flexibility in a switched network. A group of devices within a VLAN communicate as if each device was attached to the same cable. VLANs are based on logical connections, instead of physical connections.

As shown in the figure, VLANs in a switched network enable users in various departments (i.e., IT, HR, and Sales) to connect to the same network regardless of the physical switch being used or location in a campus LAN.



VLANs allow an administrator to segment networks based on factors such as function, team, or application, without regard for the physical location of the users or devices. Each VLAN is considered a separate logical network. Devices within a VLAN act as if they are in their own independent network, even if they share a common infrastructure with other VLANs. Any switch port can belong to a VLAN.

Unicast, broadcast, and multicast packets are forwarded and flooded only to end devices within the VLAN where the packets are sourced. Packets destined for devices that do not belong to the VLAN must be forwarded through a device that supports routing.

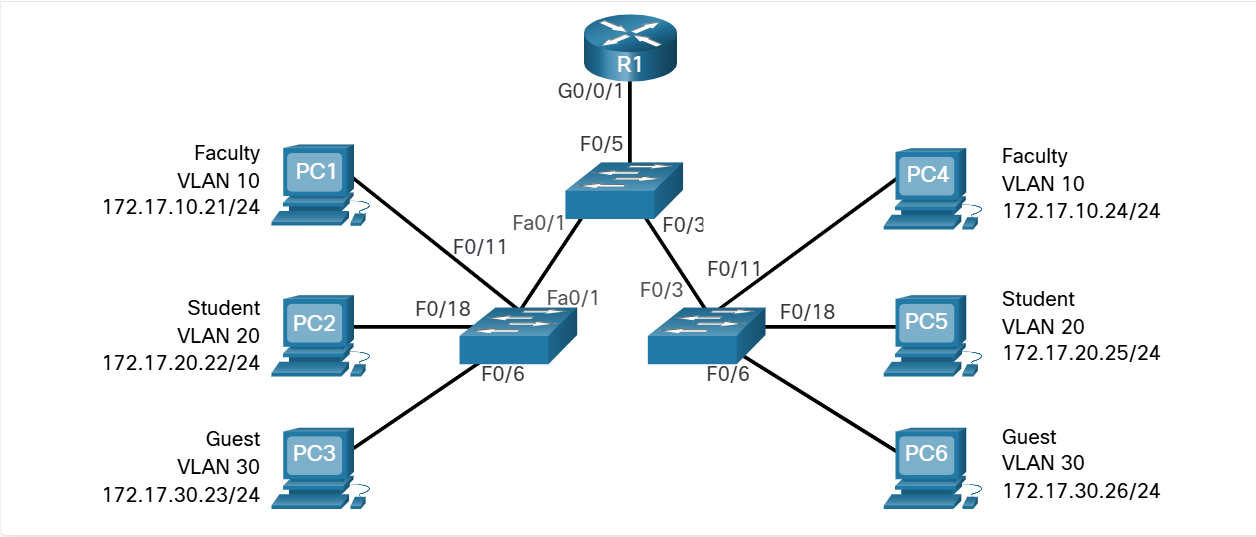
Multiple IP subnets can exist on a switched network, without the use of multiple VLANs. However, the devices will be in the same Layer 2 broadcast domain. This means that any Layer 2 broadcasts, such as an ARP request, will be received by all devices on the switched network, even by those not intended to receive the broadcast.

A VLAN creates a logical broadcast domain that can span multiple physical LAN segments. VLANs improve network performance by separating large broadcast domains into smaller ones. If a device in one VLAN sends a broadcast Ethernet frame, all devices in the VLAN receive the frame, but devices in other VLANs do not.

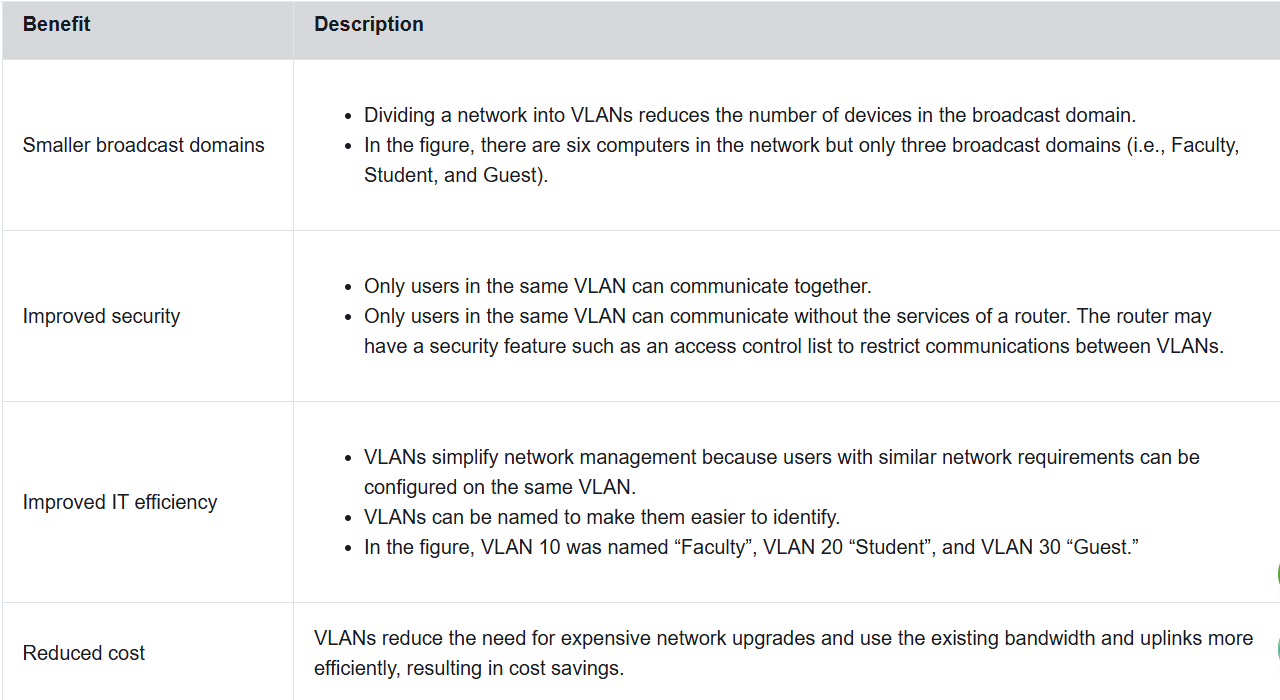
Using VLANs, network administrators can implement access and security policies according to specific groupings of users. Each switch port can be assigned to only one VLAN (except for a port connected to an IP phone or to another switch).

### Benefits of a VLAN Design

Each VLAN in a switched network corresponds to an IP network. Therefore, VLAN design must take into consideration the implementation of a hierarchical network-addressing scheme. Hierarchical network addressing means that IP network numbers are applied to network segments or VLANs in a way that takes the network as a whole into consideration. Blocks of contiguous network addresses are reserved for and configured on devices in a specific area of the network, as shown in the figure.



The table lists the benefits of designing a network with VLANs





### Types of VLANs

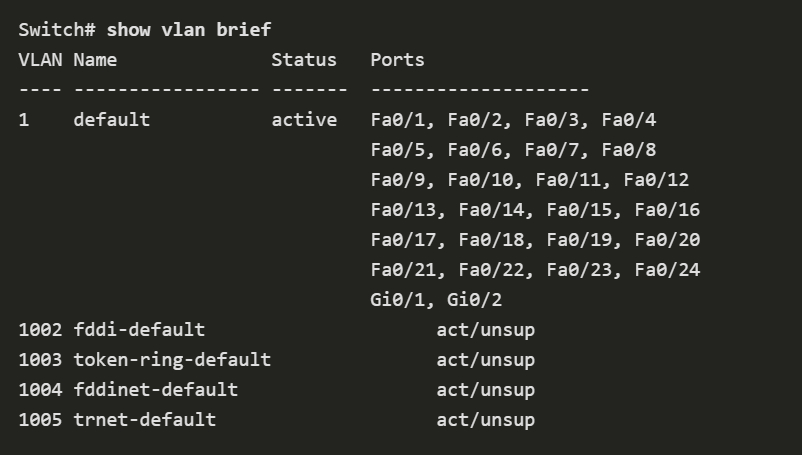
#### Default VLAN

The default VLAN on a Cisco switch is VLAN 1. Therefore, all switch ports are on VLAN 1 unless it is explicitly configured to be on another VLAN. By default, all Layer 2 control traffic is associated with VLAN 1.

Important facts to remember about VLAN 1 include the following:

* All ports are assigned to VLAN 1 by default.
* The native VLAN is VLAN 1 by default.
* The management VLAN is VLAN 1 by default.
* VLAN 1 cannot be renamed or deleted.

For instance, in the **show vlan brief** output, all ports are currently assigned to the default VLAN 1. No native VLAN is explicitly assigned and no other VLANs are active; therefore, the network is designed with the native VLAN the same as the management VLAN. This is considered a security risk.



#### Data VLAN

Data VLANs are VLANs configured to separate user-generated traffic. They are referred to as user VLANs because they separate the network into groups of users or devices. A modern network would have many data VLANs depending on organizational requirements. Note that voice and network management traffic should not be permitted on data VLANs.

#### Native VLAN

User traffic from a VLAN must be tagged with its VLAN ID when it is sent to another switch. Trunk ports are used between switches to support the transmission of tagged traffic. Specifically, an 802.1Q trunk port inserts a 4-byte tag in the Ethernet frame header to identify the VLAN to which the frame belongs.

A switch may also have to send untagged traffic across a trunk link. Untagged traffic is generated by a switch and may also come from legacy devices. The 802.1Q trunk port places untagged traffic on the native VLAN. The native VLAN on a Cisco switch is VLAN 1 (i.e., default VLAN).

It is a best practice to configure the native VLAN as an unused VLAN, distinct from VLAN 1 and other VLANs. In fact, it is not unusual to dedicate a fixed VLAN to serve the role of the native VLAN for all trunk ports in the switched domain.

#### Management VLAN

A management VLAN is a data VLAN configured specifically for network management traffic including SSH, Telnet, HTTPS, HTTP, and SNMP. By default, VLAN 1 is configured as the management VLAN on a Layer 2 switch.

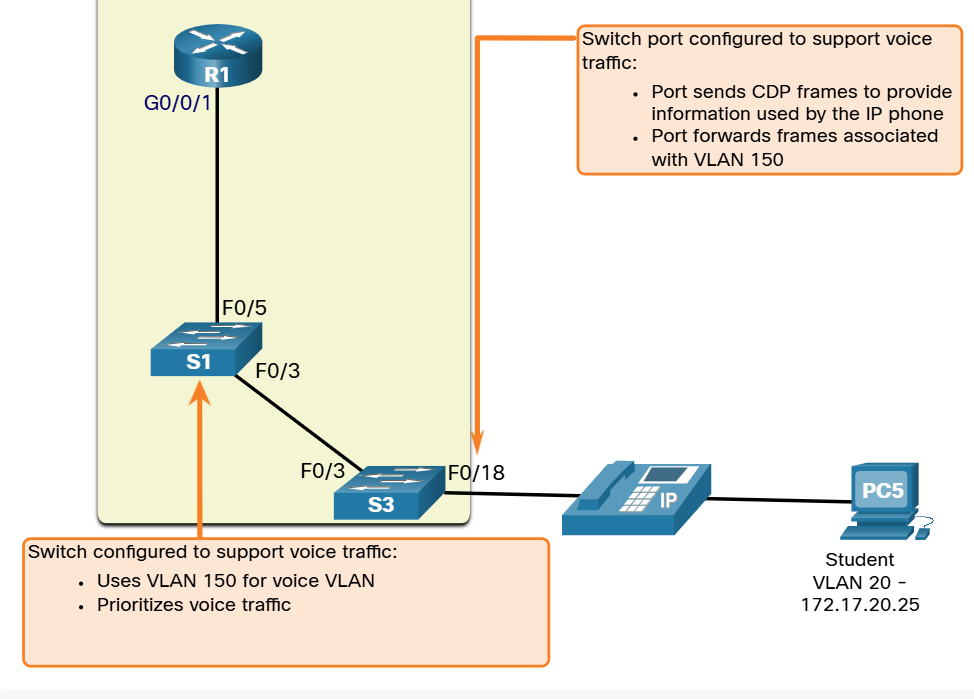
#### Voice VLAN

A separate VLAN is needed to support Voice over IP (VoIP). VoIP traffic requires the following:

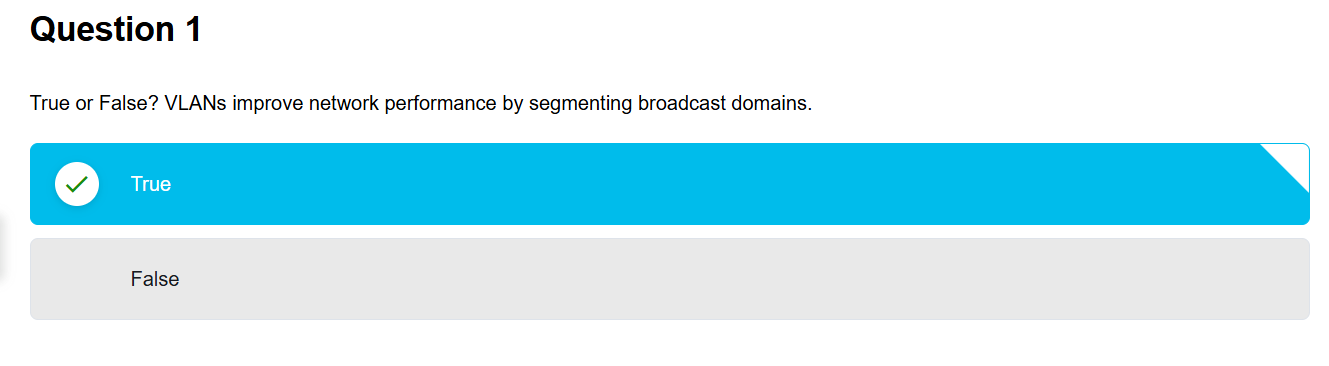
* Assured bandwidth to ensure voice quality
* Transmission priority over other types of network traffic
* Ability to be routed around congested areas on the network
* Delay of less than 150 ms across the network

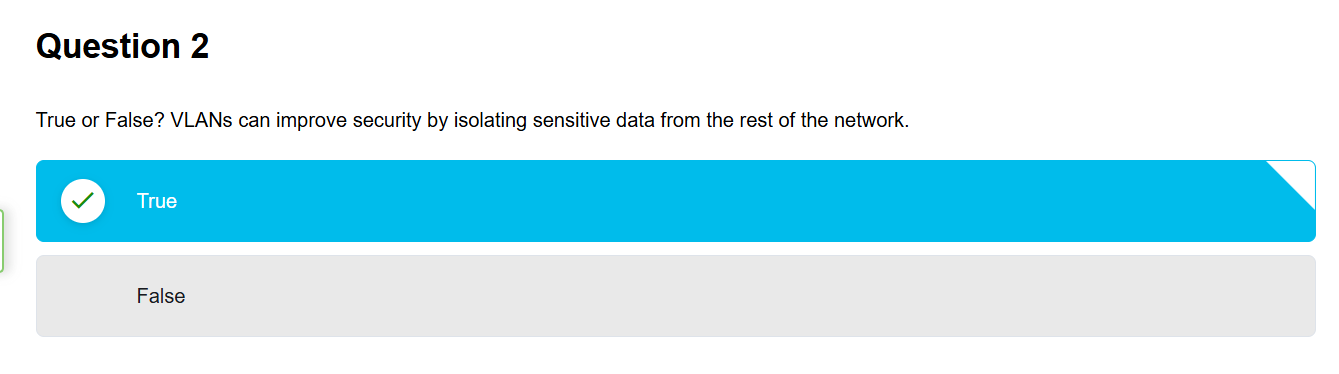
To meet these requirements, the entire network has to be designed to support VoIP.

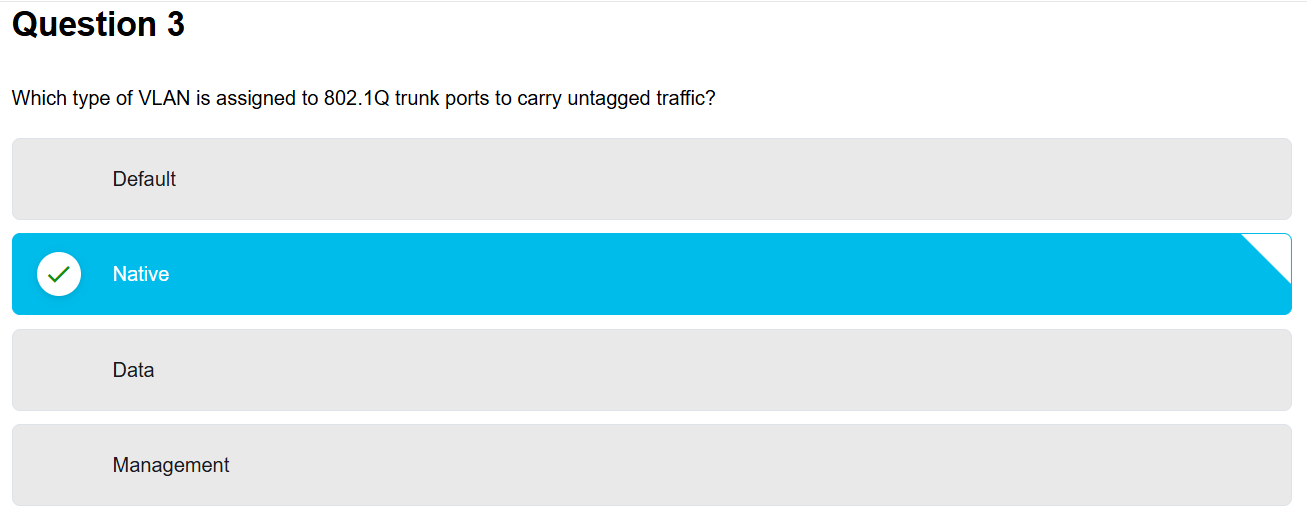
In the figure, VLAN 150 is designed to carry voice traffic. The student computer PC5 is attached to the Cisco IP phone, and the phone is attached to switch S3. PC5 is in VLAN 20, which is used for student data.

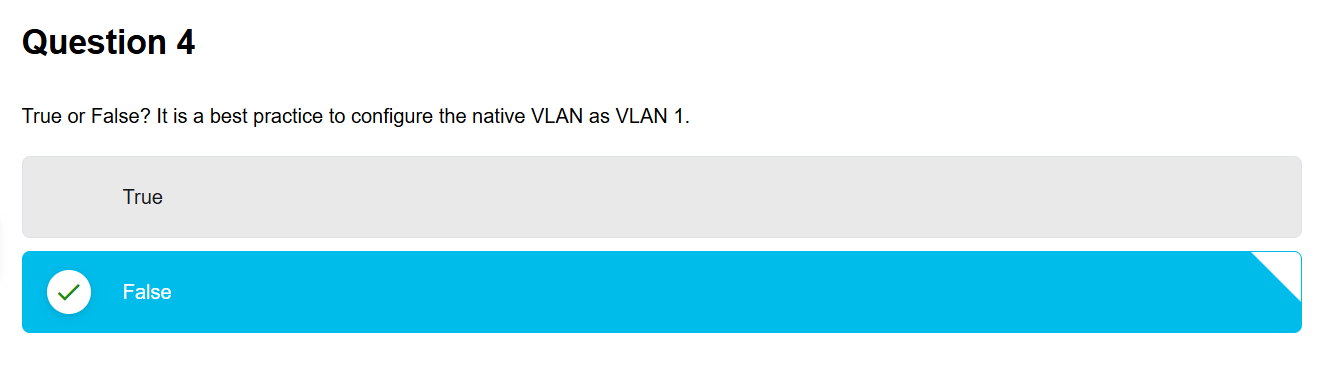


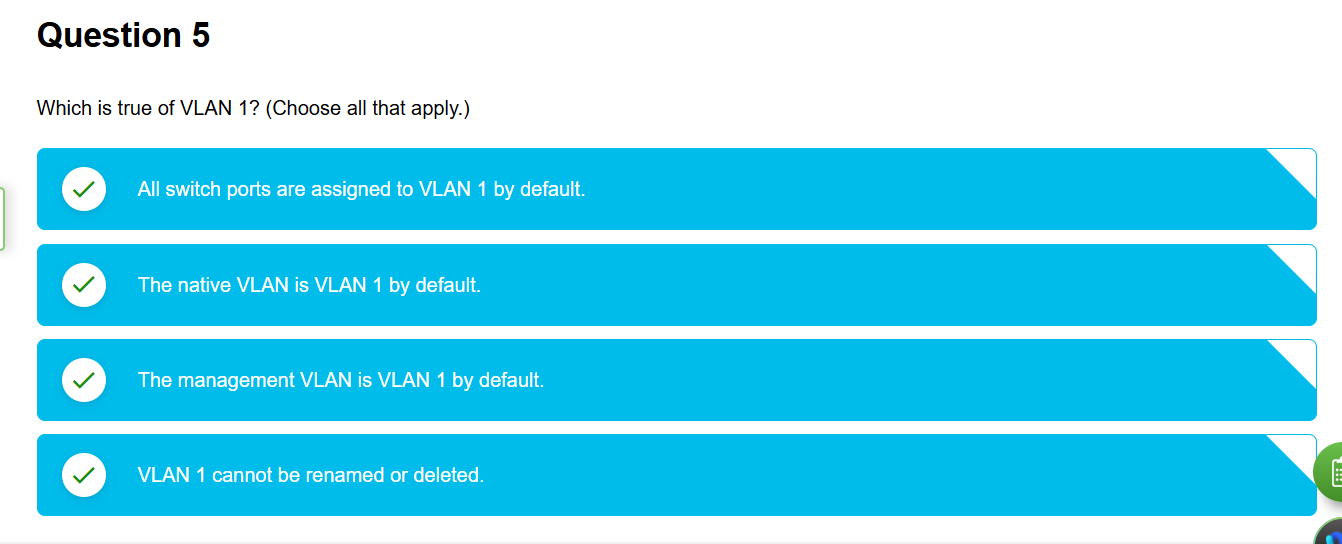
### Check Your Understanding - Overview of VLANs











## VLANs in a Multi-Switched Environment

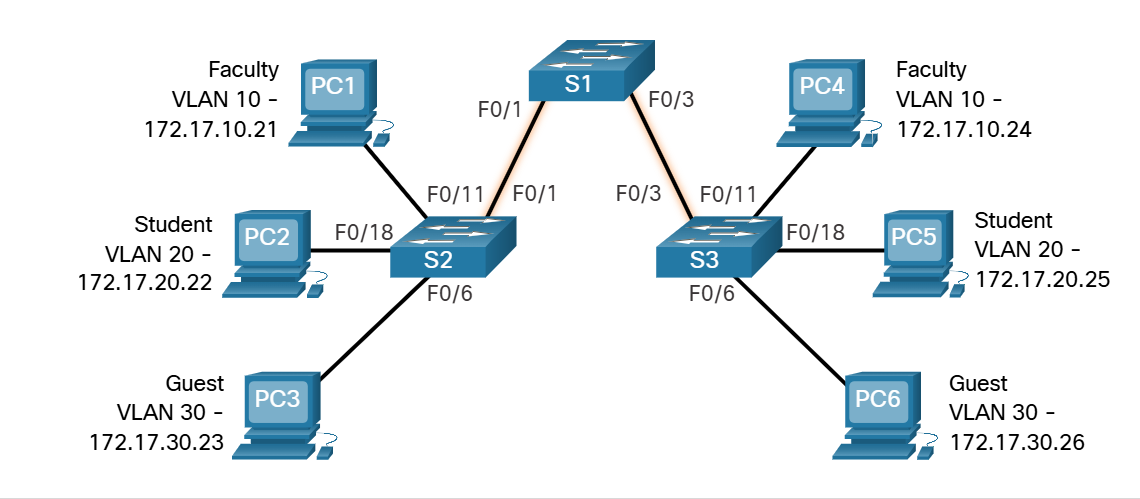
### Defining VLAN Trunks

VLANs would not be very useful without VLAN trunks. VLAN trunks allow all VLAN traffic to propagate between switches. This enables devices connected to different switches but in the same VLAN to communicate without going through a router.

A trunk is a point-to-point link between two network devices that carries more than one VLAN. A VLAN trunk extends VLANs across an entire network. Cisco supports IEEE 802.1Q for coordinating trunks on Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces.

A VLAN trunk does not belong to a specific VLAN. Instead, it is a conduit for multiple VLANs between switches and routers. A trunk could also be used between a network device and server or another device that is equipped with an appropriate 802.1Q-capable NIC. By default, on a Cisco Catalyst switch, all VLANs are supported on a trunk port.

In the figure, the highlighted links between switches S1 and S2, and S1 and S3 are configured to transmit traffic coming from VLANs 10, 20, 30, and 99 (i.e., native VLAN) across the network. This network could not function without VLAN trunks.



### Network without VLANs

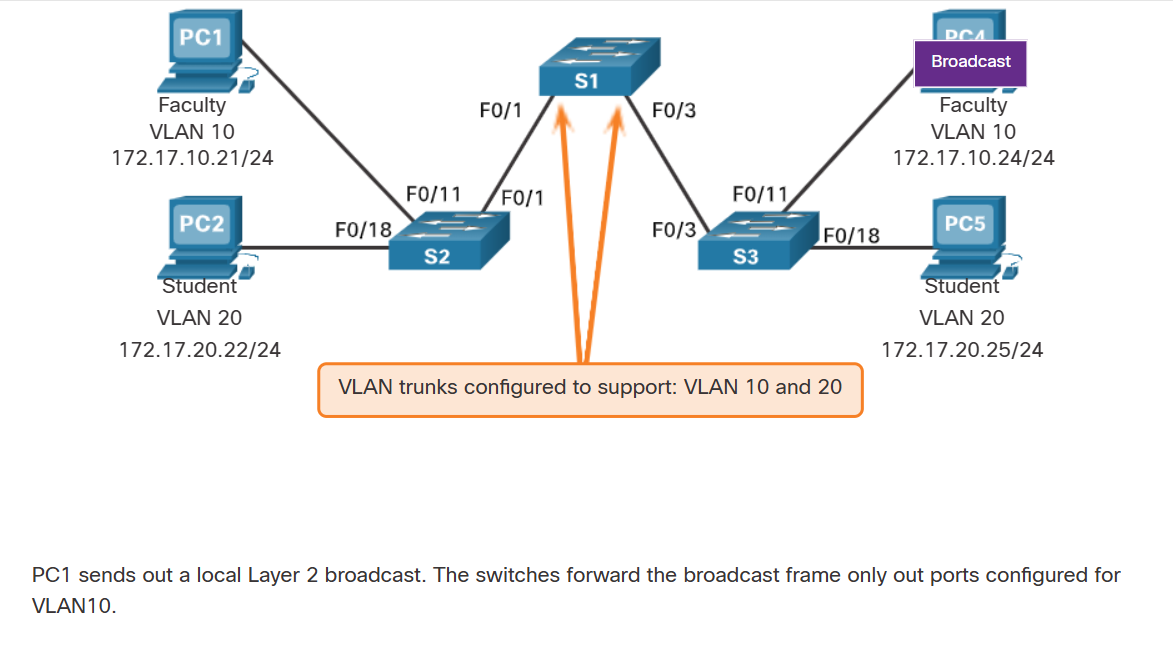
When a switch receives a broadcast frame on one of its ports, it forwards the frame out all other ports except the port where the broadcast was received. In the animation, the entire network is configured in the same subnet (172.17.40.0/24) and no VLANs are configured. As a result, when the faculty computer (PC1) sends out a broadcast frame, switch S2 sends that broadcast frame out all of its ports. Eventually the entire network receives the broadcast because the network is one broadcast domain.



### Network with VLANs

VLANs are associated with and configured on individual switch ports. Devices attached to those ports have no concept of VLANs. However, these devices are configured with IP addressing and are members of a specific IP network. This is where the connection between VLAN and IP network is apparent. A VLAN is the equivalent to an IP network (or subnet). VLANs are configured on the switch, whereas IP addressing is configured on the device.

Click Play in the animation to see that the same network has now been segmented using two VLANs. Faculty devices are assigned to VLAN 10 and student devices are assigned to VLAN 20. When a broadcast frame is sent from the faculty computer, PC1, to switch S2, the switch forwards that broadcast frame only to those switch ports configured to support VLAN 10.



The ports that comprise the connection between switches S2 and S1 (ports F0/1), and between S1 and S3 (ports F0/3) are trunks and have been configured to support all the VLANs in the network.

When S1 receives the broadcast frame on port F0/1, S1 forwards that broadcast frame out of the only other port configured to support VLAN 10, which is port F0/3. When S3 receives the broadcast frame on port F0/3, it forwards that broadcast frame out the only other port configured to support VLAN 10, which is port F0/11. The broadcast frame arrives at the only other computer in the network configured in VLAN 10, which is faculty computer PC4.

When VLANs are implemented on a switch, the transmission of unicast, multicast, and broadcast traffic from a host in a particular VLAN are restricted to the devices that are in that VLAN.

### VLAN Identification with a Tag

The standard Ethernet frame header does not contain information about the VLAN to which the frame belongs. Therefore, when Ethernet frames are placed on a trunk, information about the VLANs to which they belong must be added. This process, called tagging, is accomplished by using the IEEE 802.1Q header, specified in the IEEE 802.1Q standard. The 802.1Q header includes a 4-byte tag inserted within the original Ethernet frame header, specifying the VLAN to which the frame belongs.

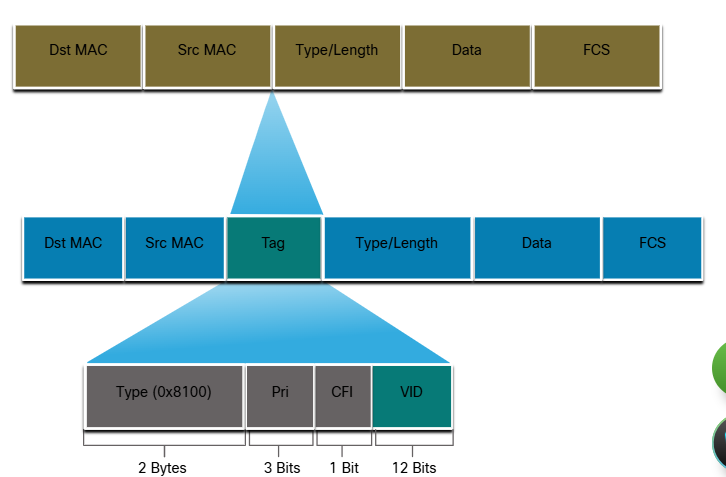
When the switch receives a frame on a port configured in access mode and assigned a VLAN, the switch inserts a VLAN tag in the frame header, recalculates the Frame Check Sequence (FCS), and sends the tagged frame out of a trunk port.

**VLAN Tag Field Details**

As shown in the figure, the VLAN tag control information field consists of a Type field, a Priority field, a Canonical Format Identifier field, and VLAN ID field:

* **Type** - A 2-byte value called the tag protocol ID (TPID) value. For Ethernet, it is set to hexadecimal 0x8100.
* **User priority** - A 3-bit value that supports level or service implementation.
* **Canonical Format Identifier (CFI)** - A 1-bit identifier that enables Token Ring frames to be carried across Ethernet links.
* **VLAN ID (VID)** - A 12-bit VLAN identification number that supports up to 4096 VLAN IDs.

After the switch inserts the tag control information fields, it recalculates the FCS values and inserts the new FCS into the frame.



### Native VLANs and 802.1Q Tagging

The IEEE 802.1Q standard specifies a native VLAN for trunk links, which defaults to VLAN 1. When an untagged frame arrives on a trunk port it is assigned to the native VLAN. Management frames that are sent between switches is an example of traffic that is typically untagged. If the link between two switches is a trunk, the switch sends the untagged traffic on the native VLAN.

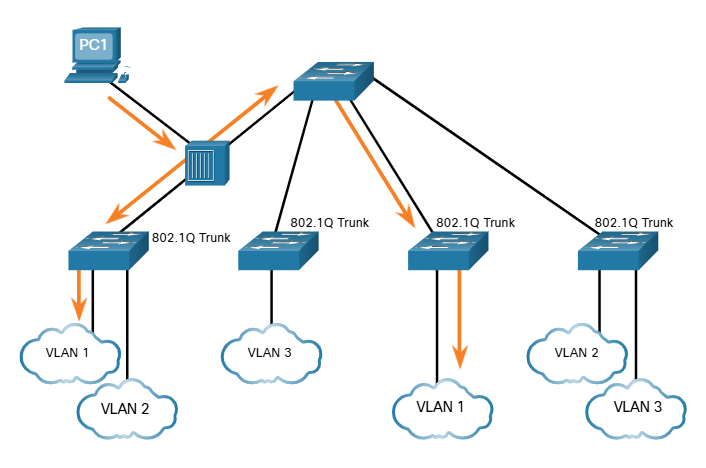
#### Tagged Frames on the Native VLAN

Some devices that support trunking add a VLAN tag to native VLAN traffic. Control traffic sent on the native VLAN should not be tagged. If an 802.1Q trunk port receives a tagged frame with the VLAN ID that is the same as the native VLAN, it drops the frame. Consequently, when configuring a switch port on a Cisco switch, configure devices so that they do not send tagged frames on the native VLAN. Devices from other vendors that support tagged frames on the native VLAN include IP phones, servers, routers, and non-Cisco switches.

#### Untagged Frames on the Native VLAN

When a Cisco switch trunk port receives untagged frames (which are unusual in a well-designed network), it forwards those frames to the native VLAN. If there are no devices associated with the native VLAN (which is not unusual) and there are no other trunk ports (which is not unusual), then the frame is dropped. The default native VLAN is VLAN 1. When configuring an 802.1Q trunk port, a default Port VLAN ID (PVID) is assigned the value of the native VLAN ID. All untagged traffic coming in or out of the 802.1Q port is forwarded based on the PVID value. For example, if VLAN 99 is configured as the native VLAN, the PVID is 99 and all untagged traffic is forwarded to VLAN 99. If the native VLAN has not been reconfigured, the PVID value is set to VLAN 1.

In the figure, PC1 is connected by a hub to an 802.1Q trunk link.



PC1 sends untagged traffic, which the switches associate with the native VLAN configured on the trunk ports, and forward accordingly. Tagged traffic on the trunk received by PC1 is dropped. This scenario reflects poor network design for several reasons: it uses a hub, it has a host connected to a trunk link, and it implies that the switches have access ports assigned to the native VLAN. It also illustrates the motivation for the IEEE 802.1Q specification for native VLANs as a means of handling legacy scenarios.

### Voice VLAN Tagging

A separate voice VLAN is required to support VoIP. This enables quality of service (QoS) and security policies to be applied to voice traffic.

A Cisco IP phone connects directly to a switch port. An IP host can connect to the IP phone to gain network connectivity as well. The access port connected to the Cisco IP phone can be configured to use two separate VLANs. One VLAN is for voice traffic and the other is a data VLAN to support the host traffic. The link between the switch and the IP phone simulates a trunk link to carry both voice VLAN traffic and data VLAN traffic.

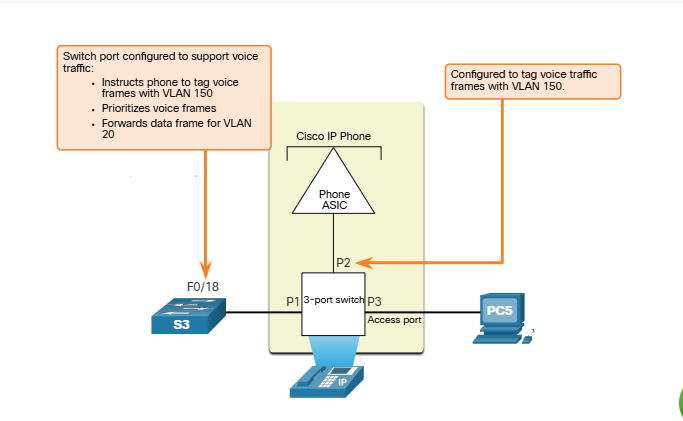
Specifically, the Cisco IP Phone contains an integrated three-port 10/100 switch. The ports provide dedicated connections to the following devices:

* Port 1 connects to the switch or other VoIP device.
* Port 2 is an internal 10/100 interface that carries the IP phone traffic.
* Port 3 (access port) connects to a PC or other device.

The switch access port sends CDP packets instructing the attached IP phone to send voice traffic in one of three ways. The method used varies based on the type of traffic:

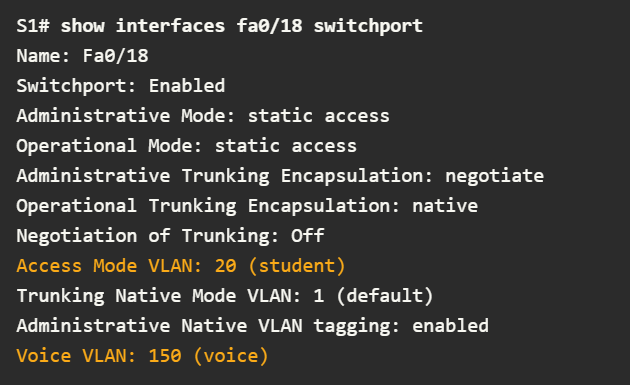
* Voice VLAN traffic must be tagged with an appropriate Layer 2 class of service (CoS) priority value
* Access VLAN traffic can also be tagged with a Layer 2 CoS priority value
* Access VLAN is not tagged (no Layer 2 CoS priority value)

In the figure, the student computer PC5 is attached to a Cisco IP phone, and the phone is attached to switch S3. VLAN 150 is designed to carry voice traffic, while PC5 is in VLAN 20, which is used for student data.

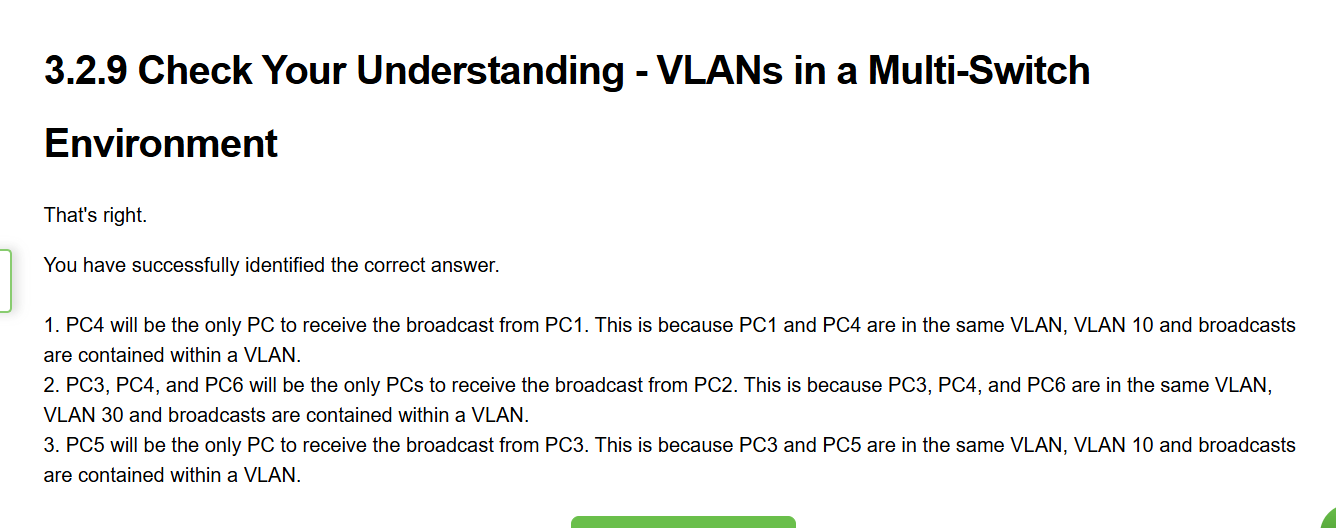


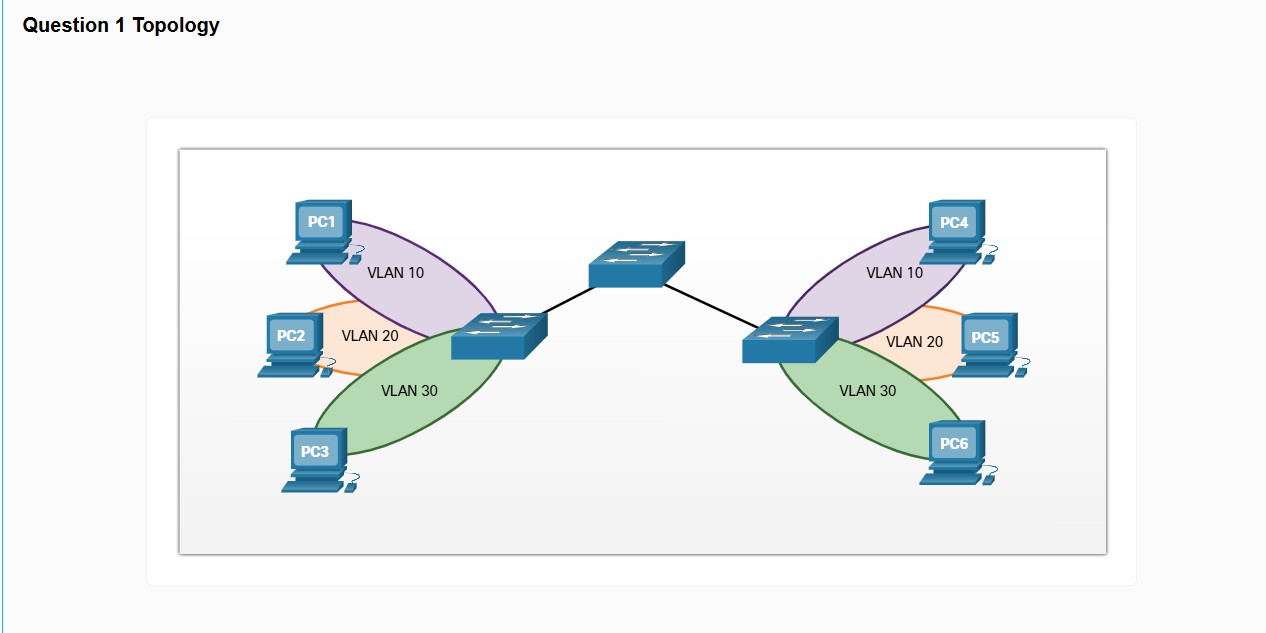
### Voice VLAN Verification Example

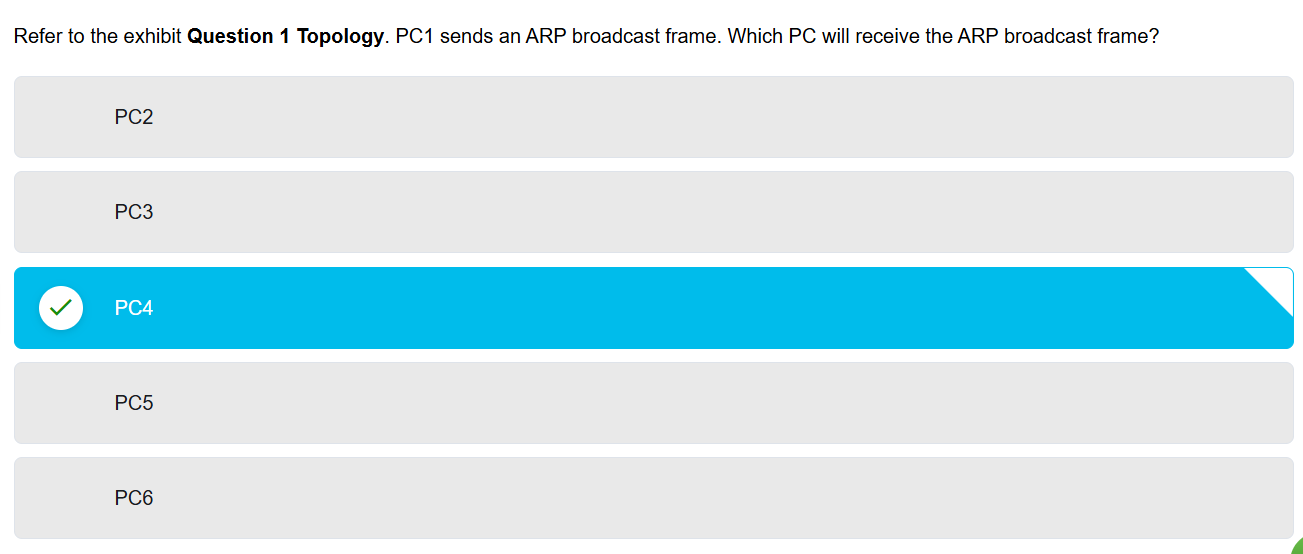
The example output for the **show interface fa0/18 switchport** command is shown. The highlighted areas in the sample output show the F0/18 interface configured with a VLAN that is configured for data (VLAN 20), and a VLAN configured for voice (VLAN 150).

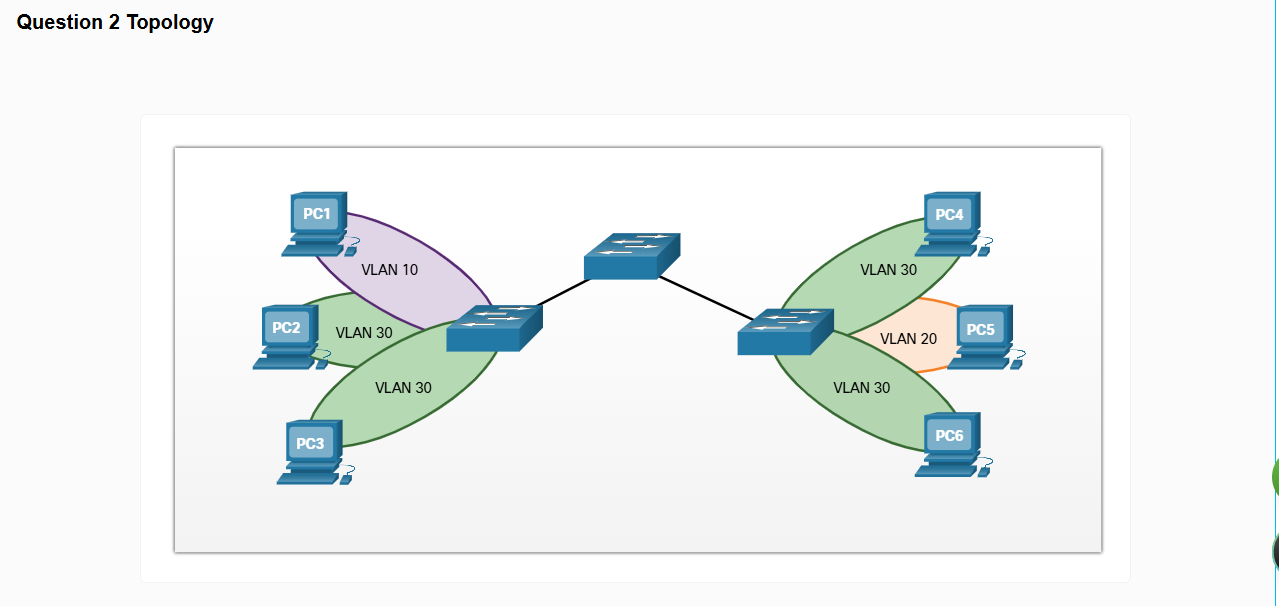


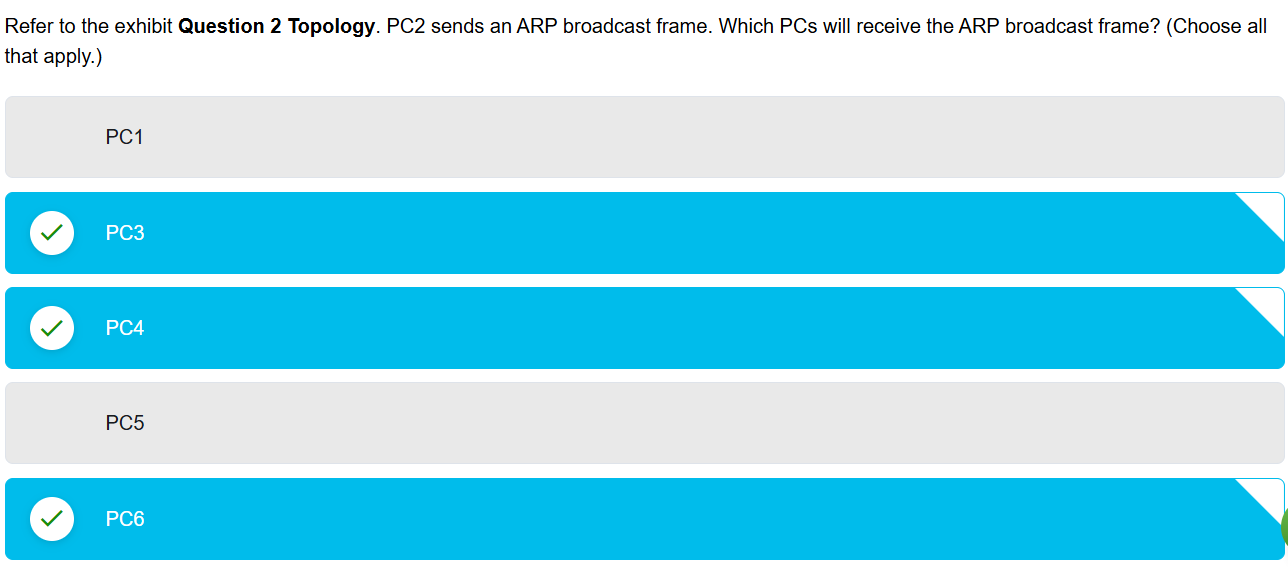
### Check Your Understanding - VLANs in a Multi-Switch Environment

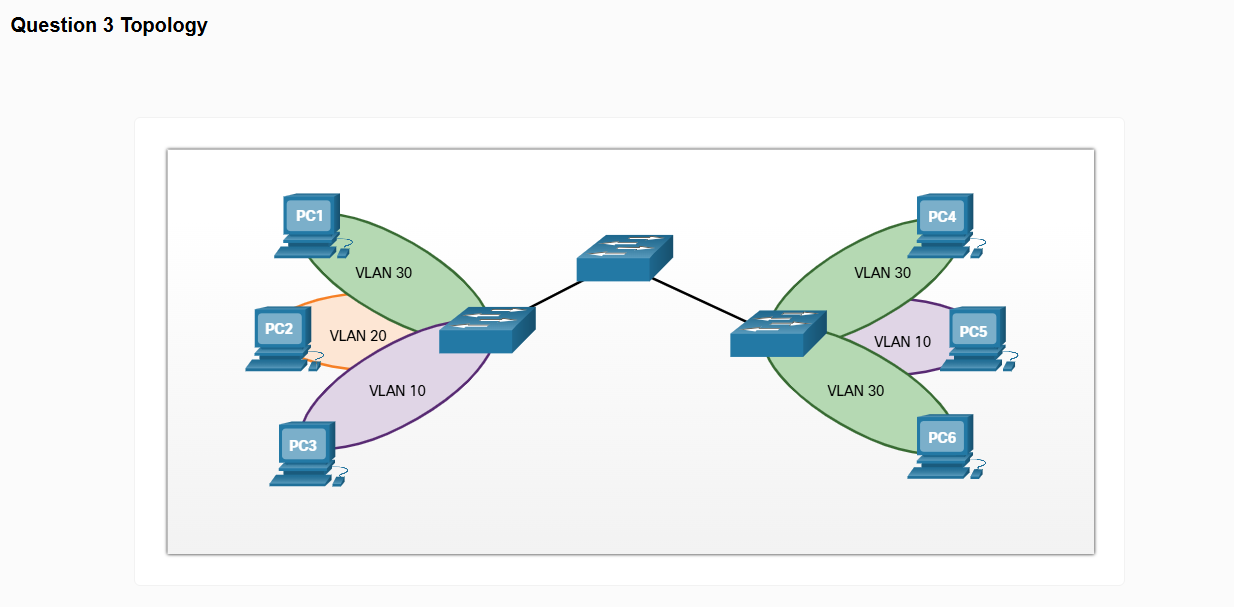


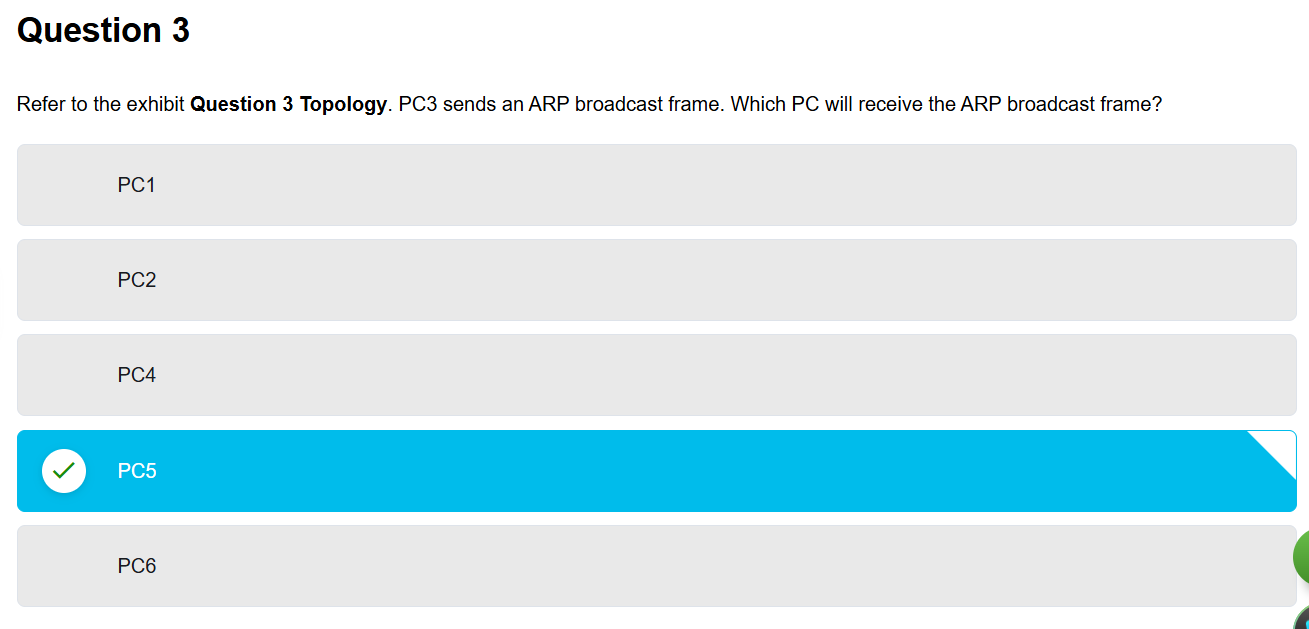










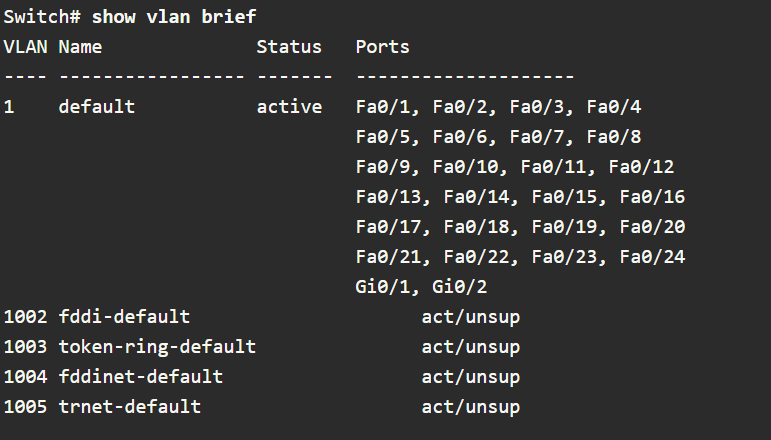


## VLAN Configuration

### VLAN Ranges on Catalyst Switches

Creating VLANs, like most other aspects of networking, is a matter of entering the appropriate commands. This topic details how to configure and verify different types of VLANs.

Different Cisco Catalyst switches support various numbers of VLANs. The number of supported VLANs is large enough to accommodate the needs of most organizations. For example, the Catalyst 2960 and 3650 Series switches support over 4,000 VLANs. Normal range VLANs on these switches are numbered 1 to 1,005 and extended range VLANs are numbered 1,006 to 4,094. The figure illustrates the default VLANs on a Catalyst 2960 switch running Cisco IOS Release 15.x.



#### Normal Range VLANs

The following are characteristics of normal range VLANs:

* They are used in all small- and medium-sized business and enterprise networks.
* They are identified by a VLAN ID between 1 and 1005.
* IDs 1002 through 1005 are reserved for legacy network technologies (i.e., Token Ring and Fiber Distributed Data Interface).
* IDs 1 and 1002 to 1005 are automatically created and cannot be removed.
* Configurations are stored in the switch flash memory in a VLAN database file called vlan.dat.
* When configured, VLAN trunking protocol (VTP), helps synchronize the VLAN database between switches.

#### Extended Range VLANs

The following are characteristics of extended range VLANs:

* They are used by service providers to service multiple customers and by global enterprises large enough to need extended range VLAN IDs.
* They are identified by a VLAN ID between 1006 and 4094.
* Configurations are saved, by default, in the running configuration.
* They support fewer VLAN features than normal range VLANs.
* Requires VTP transparent mode configuration to support extended range VLANs.

Note: 4096 is the upper boundary for the number of VLANs available on Catalyst switches, because there are 12 bits in the VLAN ID field of the IEEE 802.1Q header.

### VLAN Creation Commands

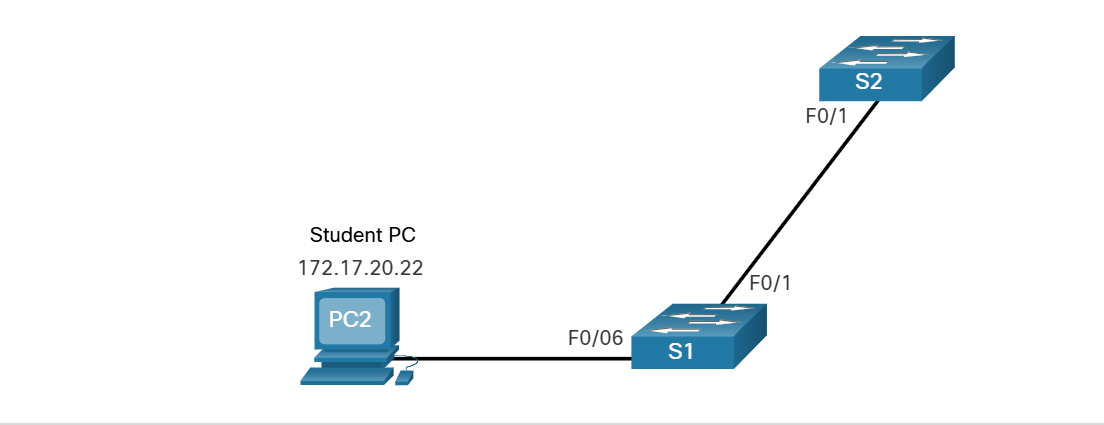
When configuring normal range VLANs, the configuration details are stored in flash memory on the switch in a file called vlan.dat. Flash memory is persistent and does not require the **copy running-config startup-config** command. However, because other details are often configured on a Cisco switch at the same time that VLANs are created, it is good practice to save running configuration changes to the startup configuration.

The table displays the Cisco IOS command syntax used to add a VLAN to a switch and give it a name. Naming each VLAN is considered a best practice in switch configuration.

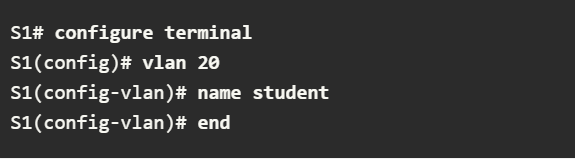


### VLAN Creation Example

In the topology, the student computer (PC2) has not been associated with a VLAN yet, but it does have an IP address of 172.17.20.22, which belongs to VLAN 20.



The example shows how the student VLAN (VLAN 20) is configured on switch S1.

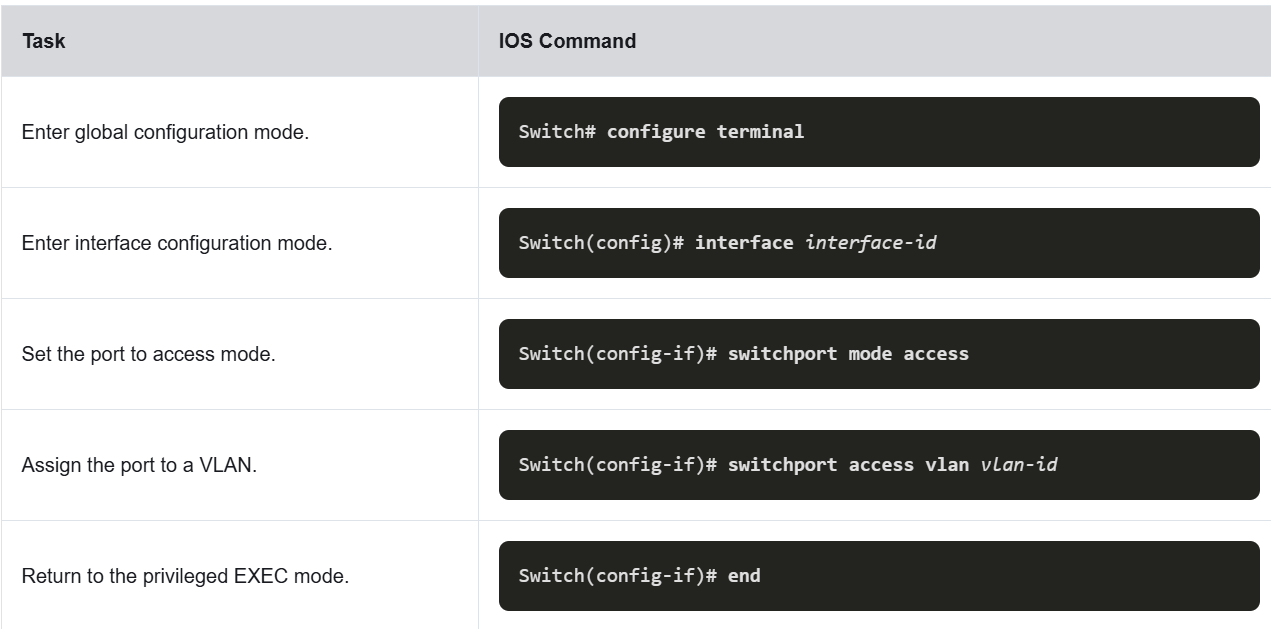


**Note:** In addition to entering a single VLAN ID, a series of VLAN IDs can be entered separated by commas, or a range of VLAN IDs separated by hyphens using the **vlan** *vlan-id* command. For example, entering the **vlan 100,102,105-107** global configuration command would create VLANs 100, 102, 105, 106, and 107.

### VLAN Port Assignment Commands

After creating a VLAN, the next step is to assign ports to the VLAN.

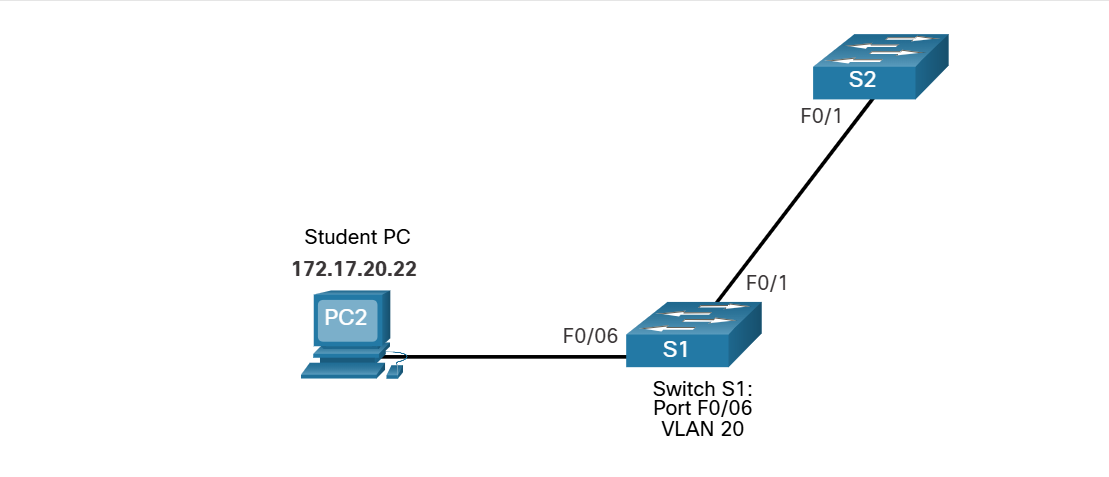
The table displays the syntax for defining a port to be an access port and assigning it to a VLAN. The **switchport mode access** command is optional, but strongly recommended as a security best practice. With this command, the interface changes to strictly access mode. Access mode indicates that the port belongs to a single VLAN and will not negotiate to become a trunk link.



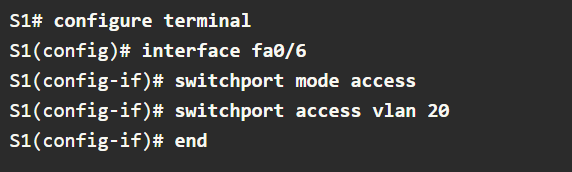
**Note:** Use the **interface range** command to simultaneously configure multiple interfaces.

### VLAN Port Assignment Example

In the figure, port F0/6 on switch S1 is configured as an access port and assigned to VLAN 20. Any device connected to that port will be associated with VLAN 20. Therefore, in our example, PC2 is in VLAN 20.



The example shows the configuration for S1 to assign F0/6 to VLAN 20.

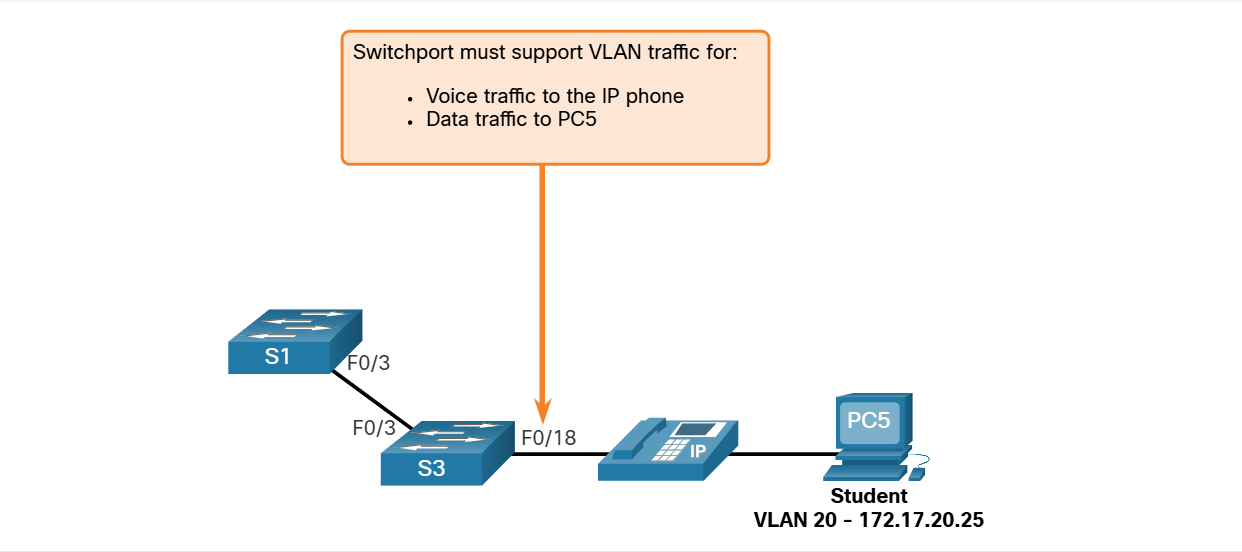


VLANs are configured on the switch port and not on the end device. PC2 is configured with an IPv4 address and subnet mask that is associated with the VLAN, which is configured on the switch port. In this example, it is VLAN 20. When VLAN 20 is configured on other switches, the network administrator must configure the other student computers to be in the same subnet as PC2 (172.17.20.0/24).

### Data and Voice VLANs

An access port can belong to only one data VLAN at a time. However, a port can also be associated to a voice VLAN. For example, a port connected to an IP phone and an end device would be associated with two VLANs: one for voice and one for data.

Consider the topology in the figure. PC5 is connected to the Cisco IP phone, which in turn is connected to the FastEthernet 0/18 interface on S3. To implement this configuration, a data VLAN and a voice VLAN are created.

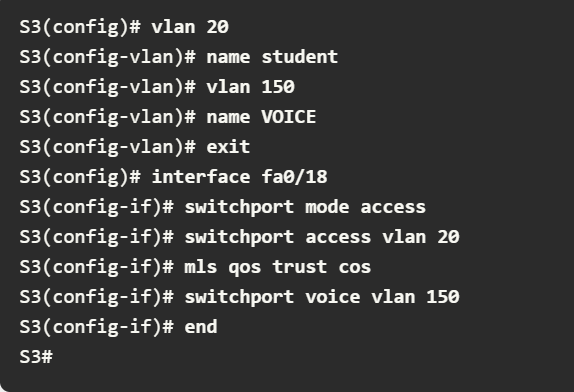


### Data and Voice VLAN Example

Use the **switchport voice vlan** *vlan-id* interface configuration command to assign a voice VLAN to a port.

LANs supporting voice traffic typically also have quality of service (QoS) enabled. Voice traffic must be labeled as trusted as soon as it enters the network. Use the **mls qos trust [cos | device cisco-phone | dscp | ip-precedence]** interface configuration command to set the trusted state of an interface, and to indicate which fields of the packet are used to classify traffic.

The configuration in the example creates the two VLANs (i.e., VLAN 20 and VLAN 150) and then assigns the F0/18 interface of S3 as a switchport in VLAN 20. It also assigns voice traffic to VLAN 150 and enables QoS classification based on the class of service (CoS) assigned by the IP phone.



**Note:** The implementation of QoS is beyond the scope of this course.

The **switchport access vlan** command forces the creation of a VLAN if it does not already exist on the switch. For example, VLAN 30 is not present in the **show vlan brief** output of the switch. If the **switchport access vlan 30** command is entered on any interface with no previous configuration, then the switch displays the following:

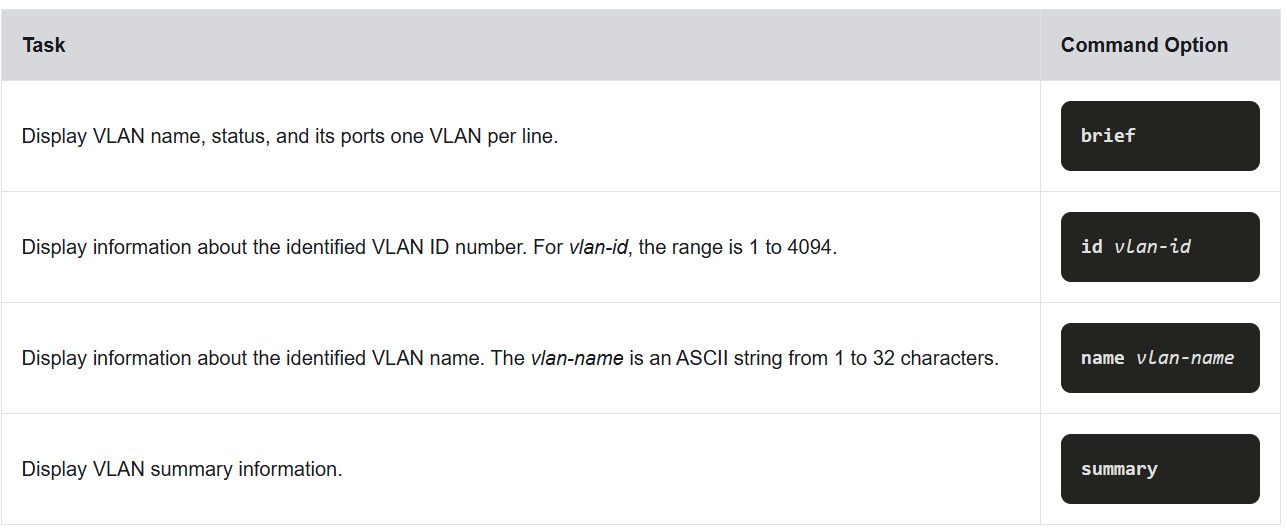


### Verify VLAN Information

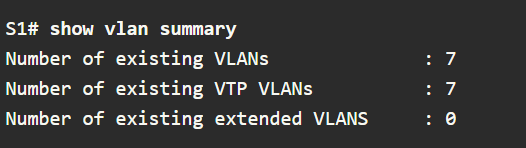
After a VLAN is configured, VLAN configurations can be validated using Cisco IOS **show** commands.

The **show vlan** command displays a list of all configured VLANs. The **show vlan** command can also be used with options. The complete syntax is **show vlan [brief** | **id** *vlan-id* | **name** *vlan-name* | **summary**].

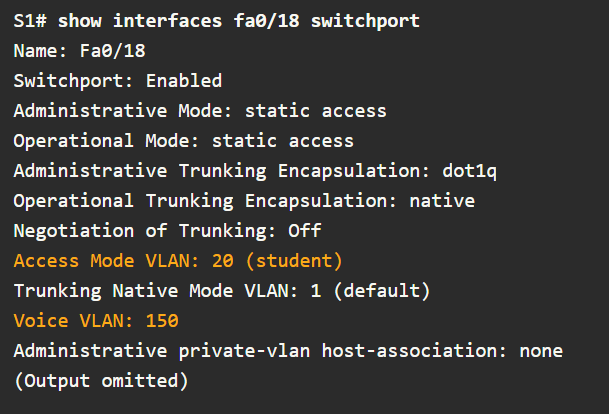
The table describes the **show vlan** command options.



The **show vlan summary** command displays the count of all configured VLANs.



Other useful commands are the **show interfaces** *interface-id* **switchport** and the **show interfaces vlan** *vlan-id* command. For example, the **show interfaces fa0/18 switchport** command can be used to confirm that the FastEthernet 0/18 port has been correctly assigned to data and voice VLANs.



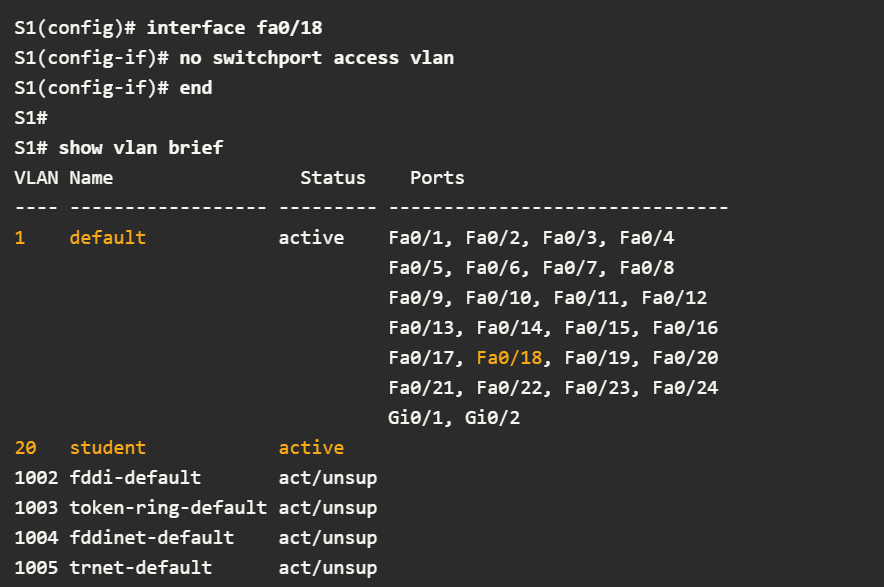
### Change VLAN Port Membership

There are a number of ways to change VLAN port membership.

If the switch access port has been incorrectly assigned to a VLAN, then simply re-enter the **switchport access vlan** *vlan-id* interface configuration command with the correct VLAN ID. For instance, assume Fa0/18 was incorrectly configured to be on the default VLAN 1 instead of VLAN 20. To change the port to VLAN 20, simply enter **switchport access vlan 20**.

To change the membership of a port back to the default VLAN 1, use the **no switchport access vlan** interface configuration mode command as shown.

In the output for example, Fa0/18 is configured to be on the default VLAN 1 as confirmed by the **show vlan brief** command.



Notice that VLAN 20 is still active, even though no ports are assigned to it.

The **show interfaces f0/18 switchport** output can also be used to verify that the access VLAN for interface F0/18 has been reset to VLAN 1 as shown in the output.



### Delete VLANs

The **no vlan** *vlan-id* global configuration mode command is used to remove a VLAN from the switch vlan.dat file.

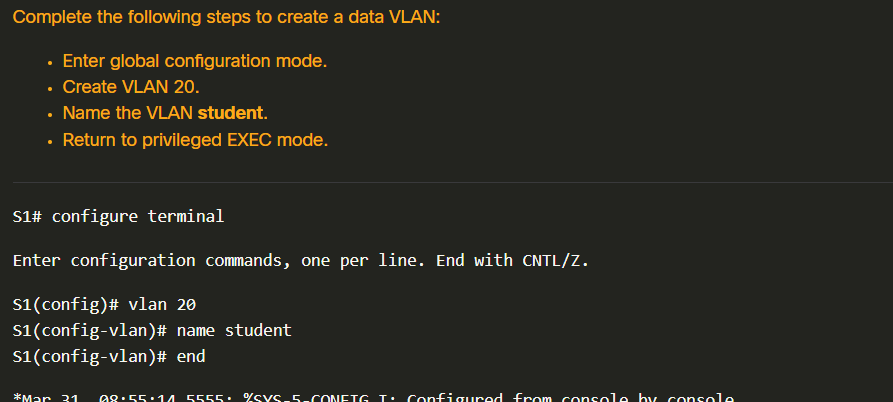
**Caution:** Before deleting a VLAN, reassign all member ports to a different VLAN first. Any ports that are not moved to an active VLAN are unable to communicate with other hosts after the VLAN is deleted and until they are assigned to an active VLAN.

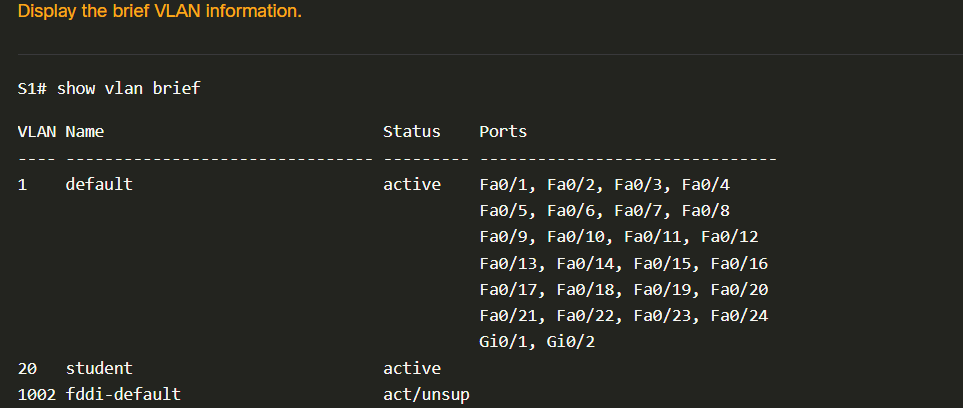
The entire vlan.dat file can be deleted using the ***delete flash:vlan.dat*** privileged EXEC mode command. The abbreviated command version (**delete vlan.dat**) can be used if the vlan.dat file has not been moved from its default location. After issuing this command and reloading the switch, any previously configured VLANs are no longer present. This effectively places the switch into its factory default condition with regard to VLAN configurations.

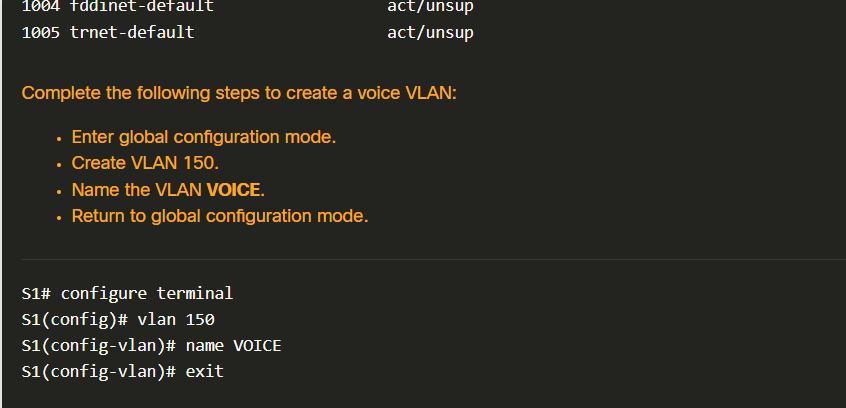
**Note:** To restore a Catalyst switch to its factory default condition, unplug all cables except the console and power cable from the switch. Then enter the **erase startup-config** privileged EXEC mode command followed by the **delete vlan.dat** command.

### Syntax Checker - VLAN Configuration

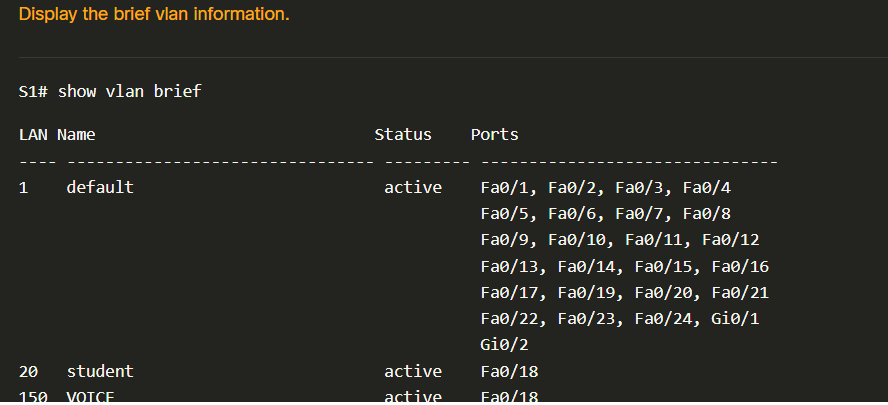
In this Syntax Checker activity, you will implement and verify a VLAN configuration for switch interfaces based on the specified requirements.

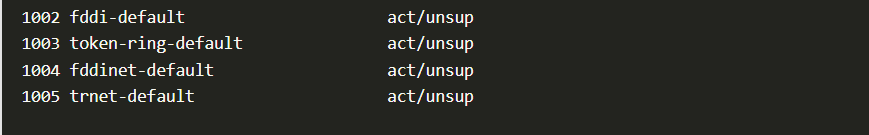


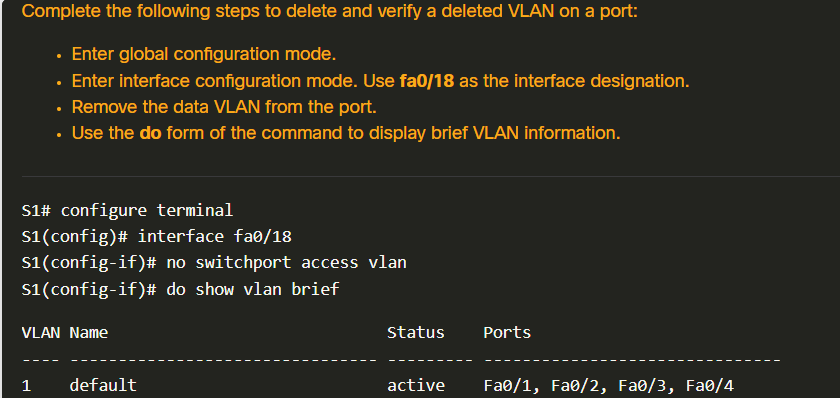


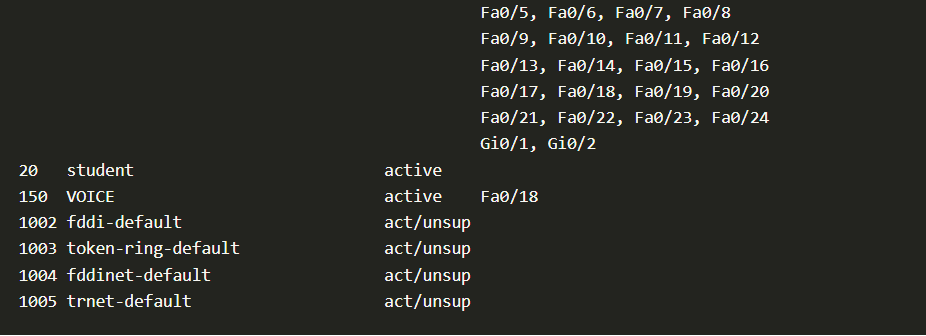


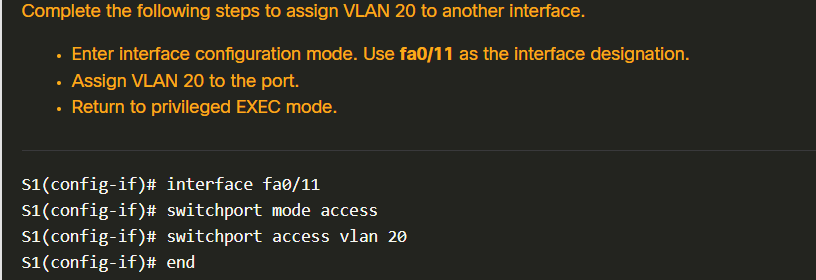


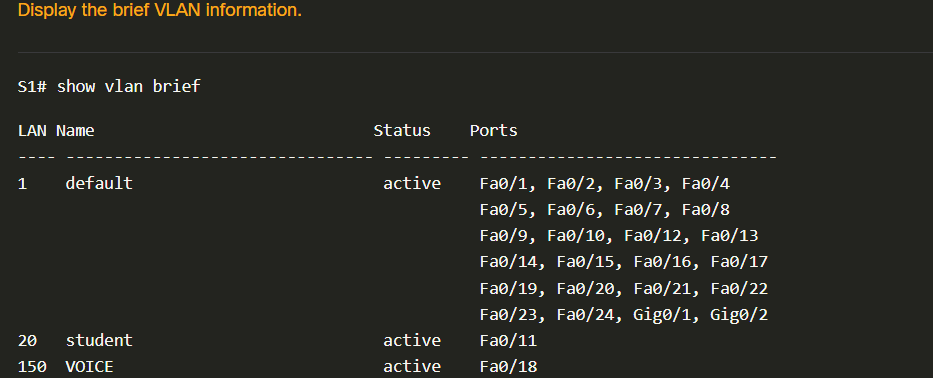


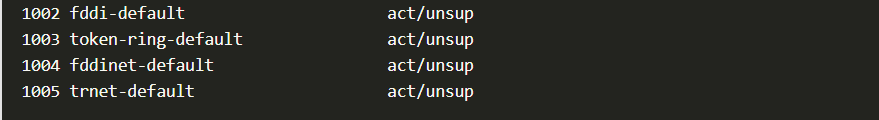


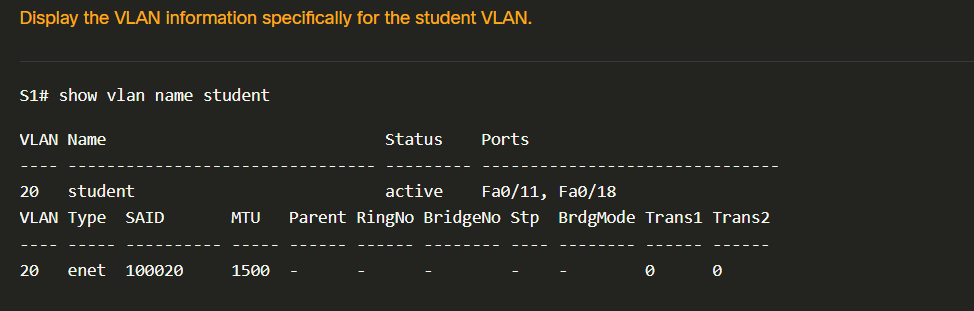


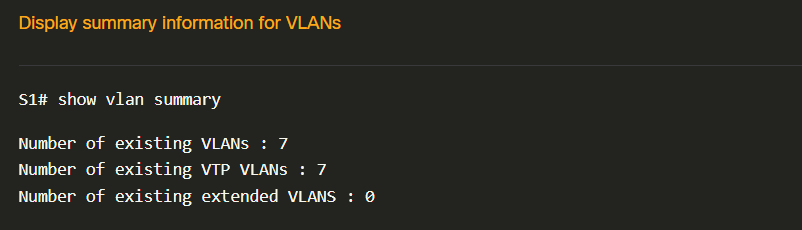


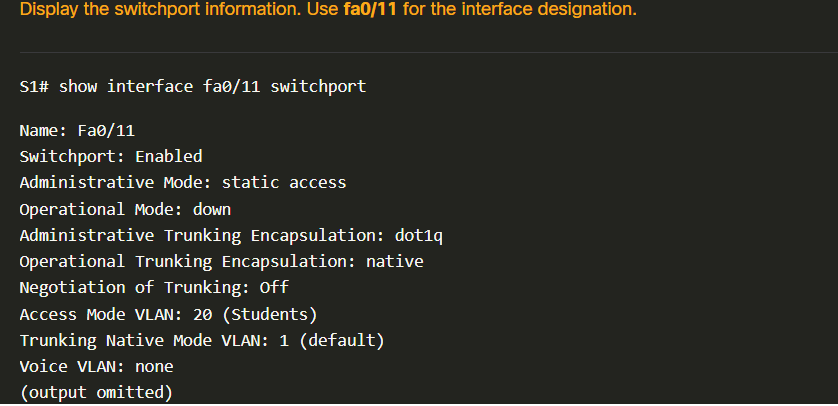










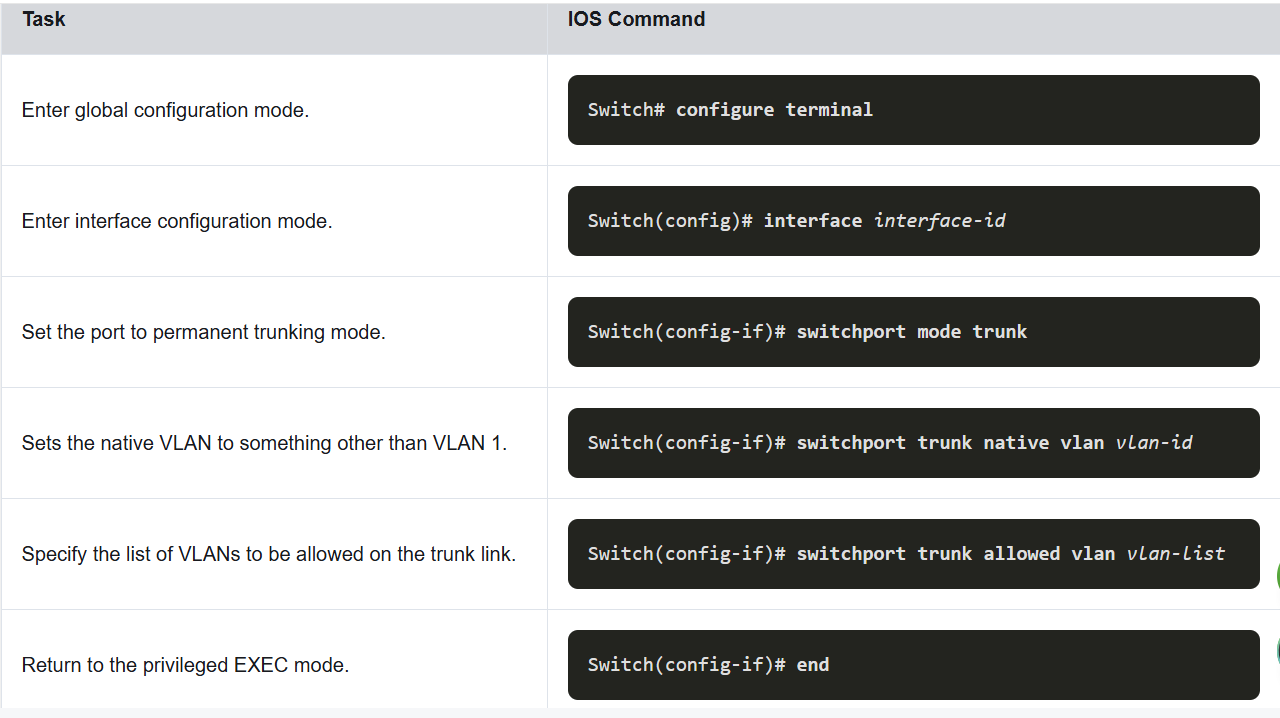


## VLAN Trunks

### Trunk Configuration Commands

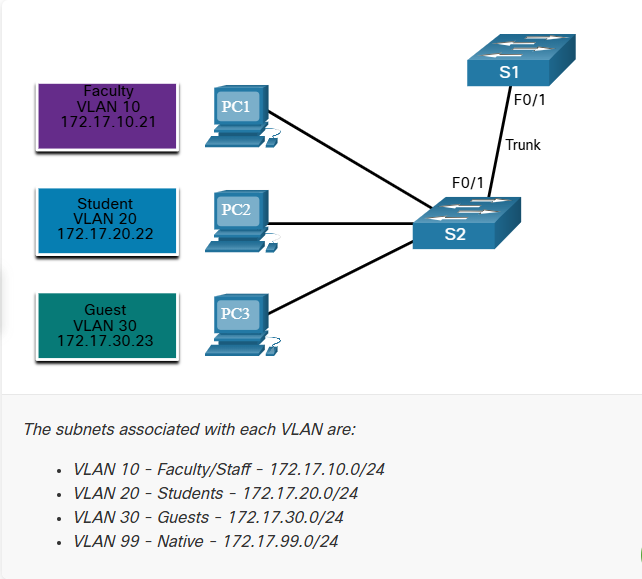
Now that you have configured and verified VLANs, it is time to configure and verify VLAN trunks. A VLAN trunk is a Layer 2 link between two switches that carries traffic for all VLANs (unless the allowed VLAN list is restricted manually or dynamically).

To enable trunk links, configure the interconnecting ports with the set of interface configuration commands shown in the table.



### Trunk Configuration Example

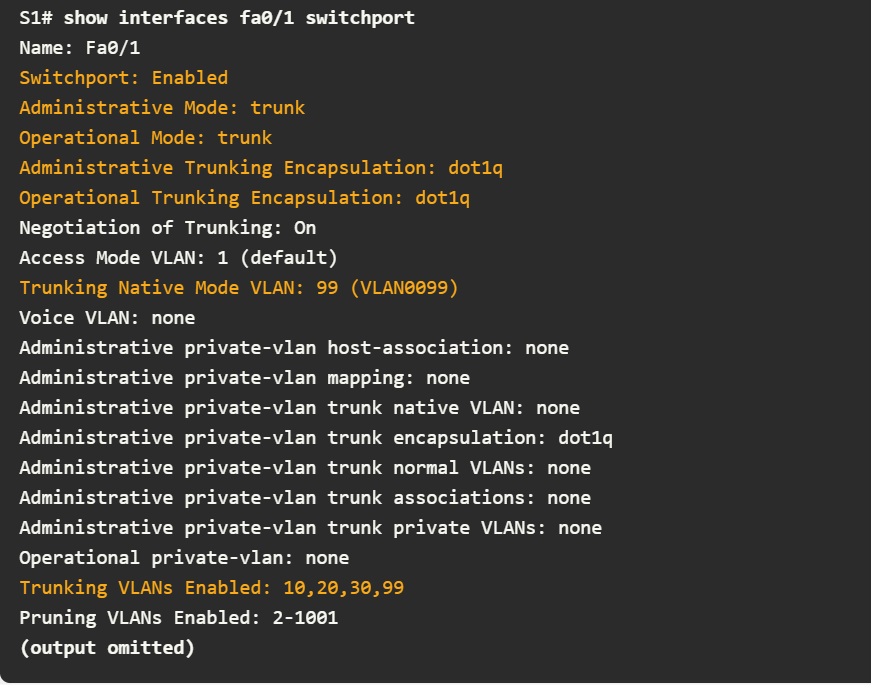
In the figure, VLANs 10, 20, and 30 support the Faculty, Student, and Guest computers (PC1, PC2, and PC3). The F0/1 port on switch S1 is configured as a trunk port and forwards traffic for VLANs 10, 20, and 30. VLAN 99 is configured as the native VLAN.



**Note:** This configuration assumes the use of Cisco Catalyst 2960 switches which automatically use 802.1Q encapsulation on trunk links. Other switches may require manual configuration of the encapsulation. Always configure both ends of a trunk link with the same native VLAN. If 802.1Q trunk configuration is not the same on both ends, Cisco IOS Software reports errors.

### Verify Trunk Configuration

The switch output displays the configuration of switch port F0/1 on switch S1. The configuration is verified with the **show interfaces** *interface-ID* **switchport** command.

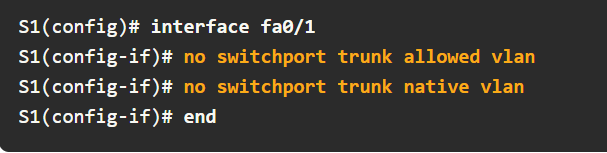


The top highlighted area shows that port F0/1 has its administrative mode set to **trunk**. The port is in trunking mode. The next highlighted area verifies that the native VLAN is VLAN 99. Further down in the output, the bottom highlighted area shows that VLANs 10, 20, 30, and 99 are enabled on the trunk.

**Note:** Another useful command for veryfing trunk interfaces is the **show interface trunk** command.

### Reset the Trunk to the Default State

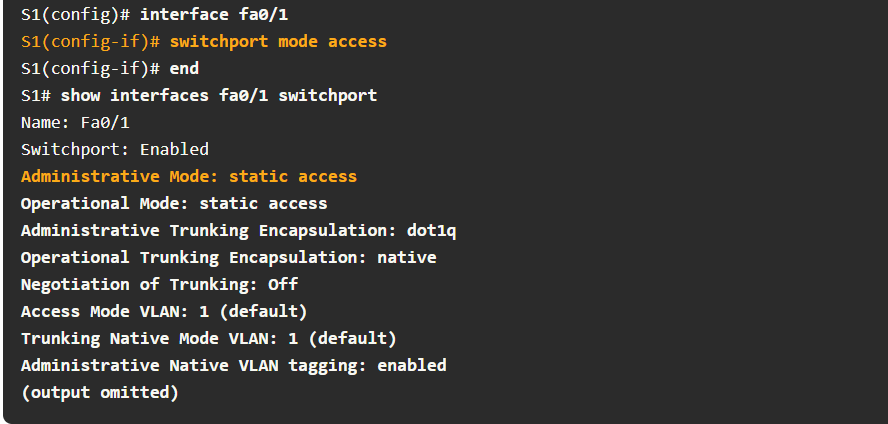
Use the **no switchport trunk allowed vlan** and the **no switchport trunk native vlan** commands to remove the allowed VLANs and reset the native VLAN of the trunk. When it is reset to the default state, the trunk allows all VLANs and uses VLAN 1 as the native VLAN. The example shows the commands used to reset all trunking characteristics of a trunking interface to the default settings.



The**show interfaces fa0/1 switchport** command reveals that the trunk has been reconfigured to a default state.



This sample output shows the commands used to remove the trunk feature from the F0/1 switch port on switch S1. The **show interfaces f0/1 switchport** command reveals that the F0/1 interface is now in static access mode.



## Dynamic Trunking Protocol

### Introduction to DTP

Some Cisco switches have a proprietary protocol that lets them automatically negotiate trunking with a neighboring device. This protocol is called Dynamic Trunking Protocol (DTP). DTP can speed up the configuration process for a network administrator. Ethernet trunk interfaces support different trunking modes. An interface can be set to trunking or nontrunking, or to negotiate trunking with the neighbor interface. Trunk negotiation is managed by DTP, which operates on a point-to-point basis only, between network devices.

DTP is a Cisco proprietary protocol that is automatically enabled on Catalyst 2960 and Catalyst 3650 Series switches. DTP manages trunk negotiation only if the port on the neighbor switch is configured in a trunk mode that supports DTP. Switches from other vendors do not support DTP.

**Caution:** Some internetworking devices might forward DTP frames improperly, which can cause misconfigurations. To avoid this, turn off DTP on Cisco switch interfaces that are connected to devices that do not support DTP.

The default DTP configuration for Cisco Catalyst 2960 and 3650 switches is dynamic auto.

To enable trunking from a Cisco switch to a device that does not support DTP, use the **switchport mode trunk** and **switchport nonegotiate** interface configuration mode commands. This causes the interface to become a trunk, but it will not generate DTP frames.

S1(config-if)# **switchport mode trunk**

S1(config-if)# **switchport nonegotiate**

To re-enable dynamic trunking protocol use the **switchport mode dynamic** **auto** command.

S1(config-if)# **switchport mode dynamic auto**

If the ports connecting two switches are configured to ignore all DTP advertisements with the **switchport mode trunk** and the **switchport nonegotiate** commands, the ports will stay in trunk port mode. If the connecting ports are set to dynamic auto, they will not negotiate a trunk and will stay in the access mode state, creating an inactive trunk link.

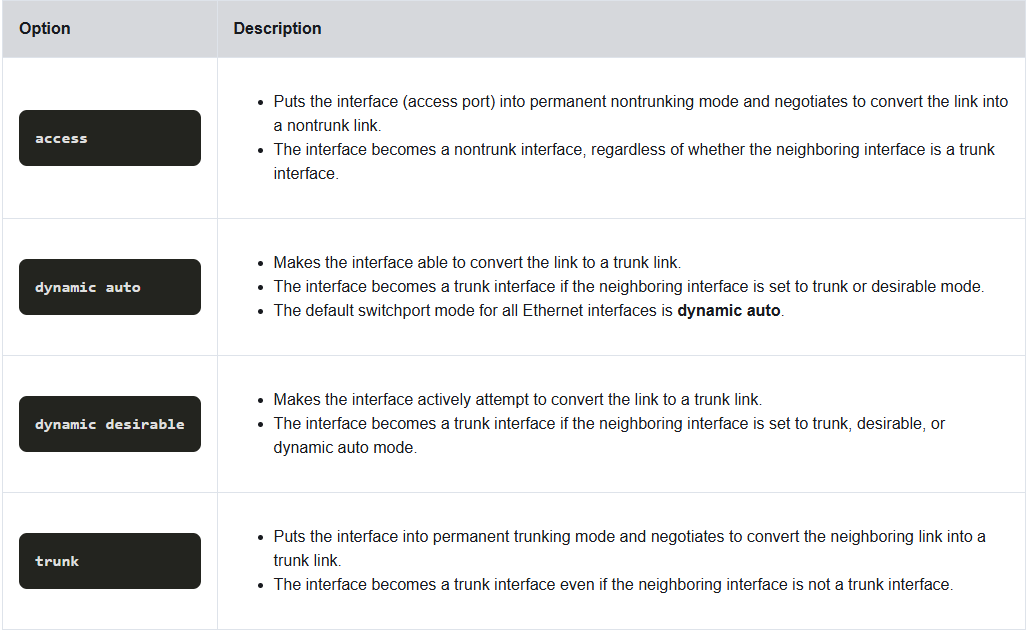
When configuring a port to be in trunk mode, use the **switchport mode trunk** command. Then there is no ambiguity about which state the trunk is in; it is always on.

### Negotiated Interface Modes

The **switchport mode** command has additional options for negotiating the interface mode. The full command syntax is the following:

Switch(config-if)# **switchport mode** { **access** | **dynamic** { **auto** | **desirable** } | **trunk** }

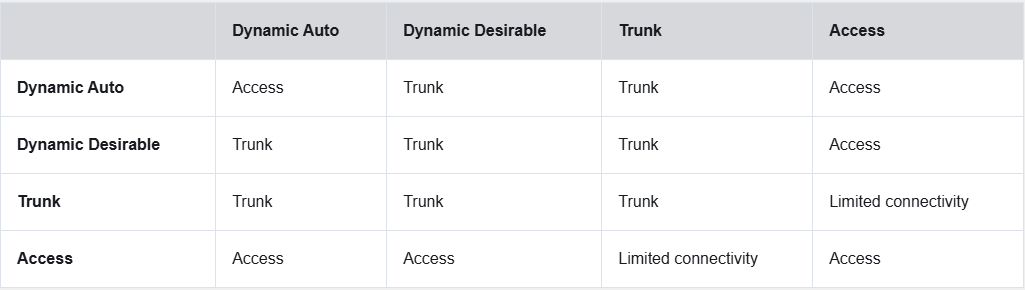
The options are described in the table.



Use the **switchport nonegotiate** interface configuration command to stop DTP negotiation. The switch does not engage in DTP negotiation on this interface. You can use this command only when the interface switchport mode is **access** or **trunk**. You must manually configure the neighboring interface as a trunk interface to establish a trunk link.

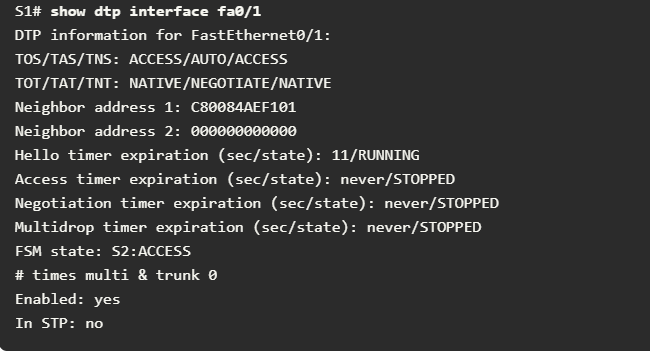
### Results of a DTP Configuration

The table illustrates the results of the DTP configuration options on opposite ends of a trunk link connected to Catalyst 2960 switch ports. Best practice is to configure trunk links statically whenever possible.



### Verify DTP Mode

The default DTP mode is dependent on the Cisco IOS Software version and on the platform. To determine the current DTP mode, issue the **show dtp interface** command as shown in the output.



**Note:** A general best practice is to set the interface to **trunk** and **nonegotiate** when a trunk link is required. On links where trunking is not intended, DTP should be turned off.

### Check Your Understanding - Dynamic Trunking Protocol

