# VLANs

**This chapter covers**

- How to divide a switch into multiple virtual switches with VLANs
- How to configure trunk ports to carry traffic in multiple VLANs
- Routing between VLANs with a router or multilayer switch

In chapter 11, we covered subnetting, which allows us to divide a network into smaller subnets. This is an example of network *segmentation*—the division of a network into smaller parts. Virtual LANs (VLANs, pronounced "V-LANs"), the topic of this chapter, can be likened to subnets in that they also allow us to divide up a network into smaller parts. With VLANs, we can divide a LAN (a broadcast domain) into smaller LANs, called VLANs. Whereas subnets allow us to segment the network at Layer 3, VLANs allow us to segment the network at Layer 2. In this chapter, we will cover three CCNA exam topics, all related to the topics of switches and VLANs:

- 1.1.b Layer 2 and Layer 3 switches
- 2.1 Configure and verify VLANs (normal range) spanning multiple switches
- 2.2 Configure and verify interswitch connectivity

## 12.1 Why we need VLANs

To understand a technology, it's important to understand why that technology exists—to understand the problem it solves. To demonstrate the role VLANs play in segmenting networks, let's examine a network without segmentation, a network with Layer 3 segmentation, and a network with both Layer 3 and Layer 2 segmentation.

### 12.1.1 Layer 3 segmentation with subnets

**Figure 12.1 An unsegmented LAN. All hosts are in the 172.16.1.0/24 network, and VLANs are not used to segment the LAN at layer 2. Hosts belonging to different departments can communicate with each other directly (by sending their packets in frames addressed directly to each other).**

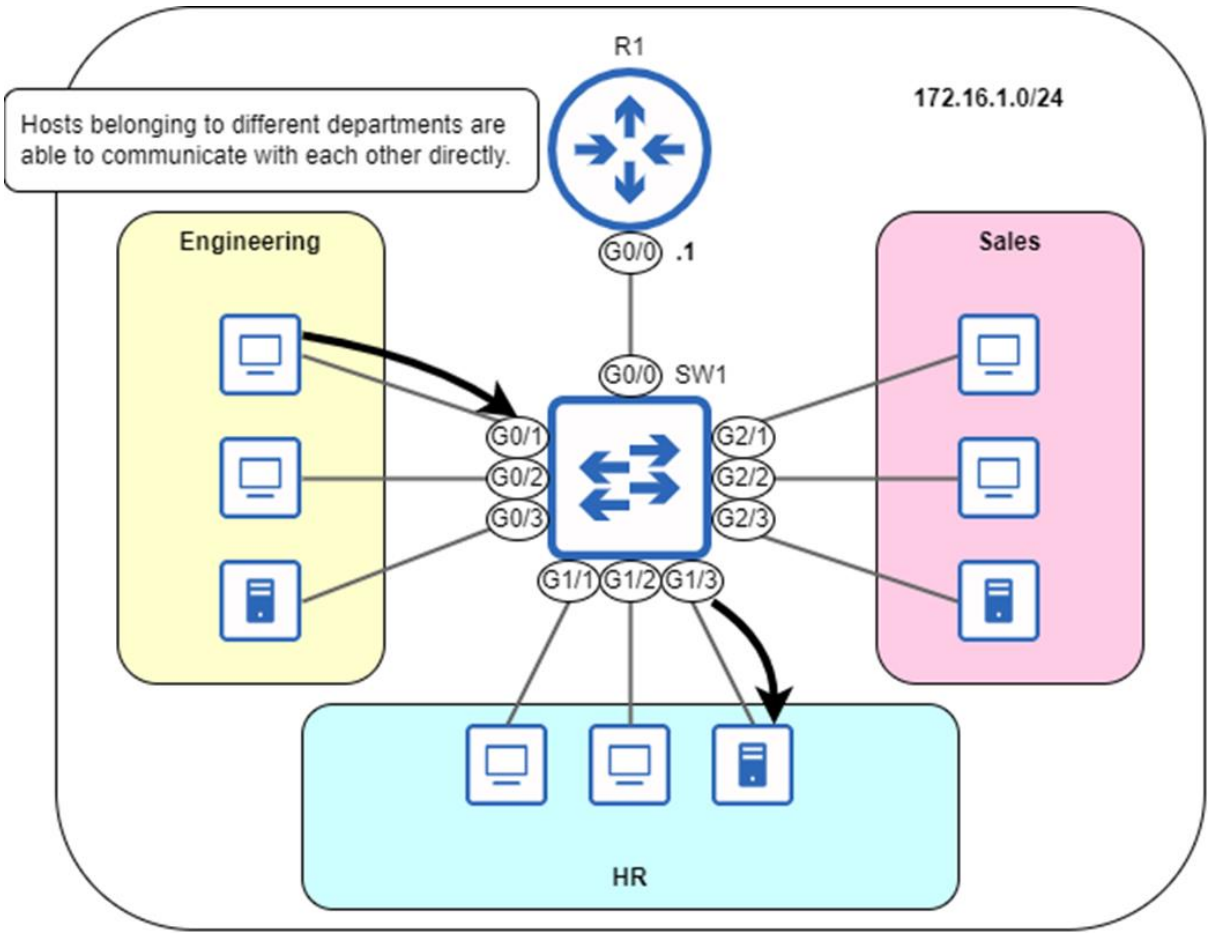
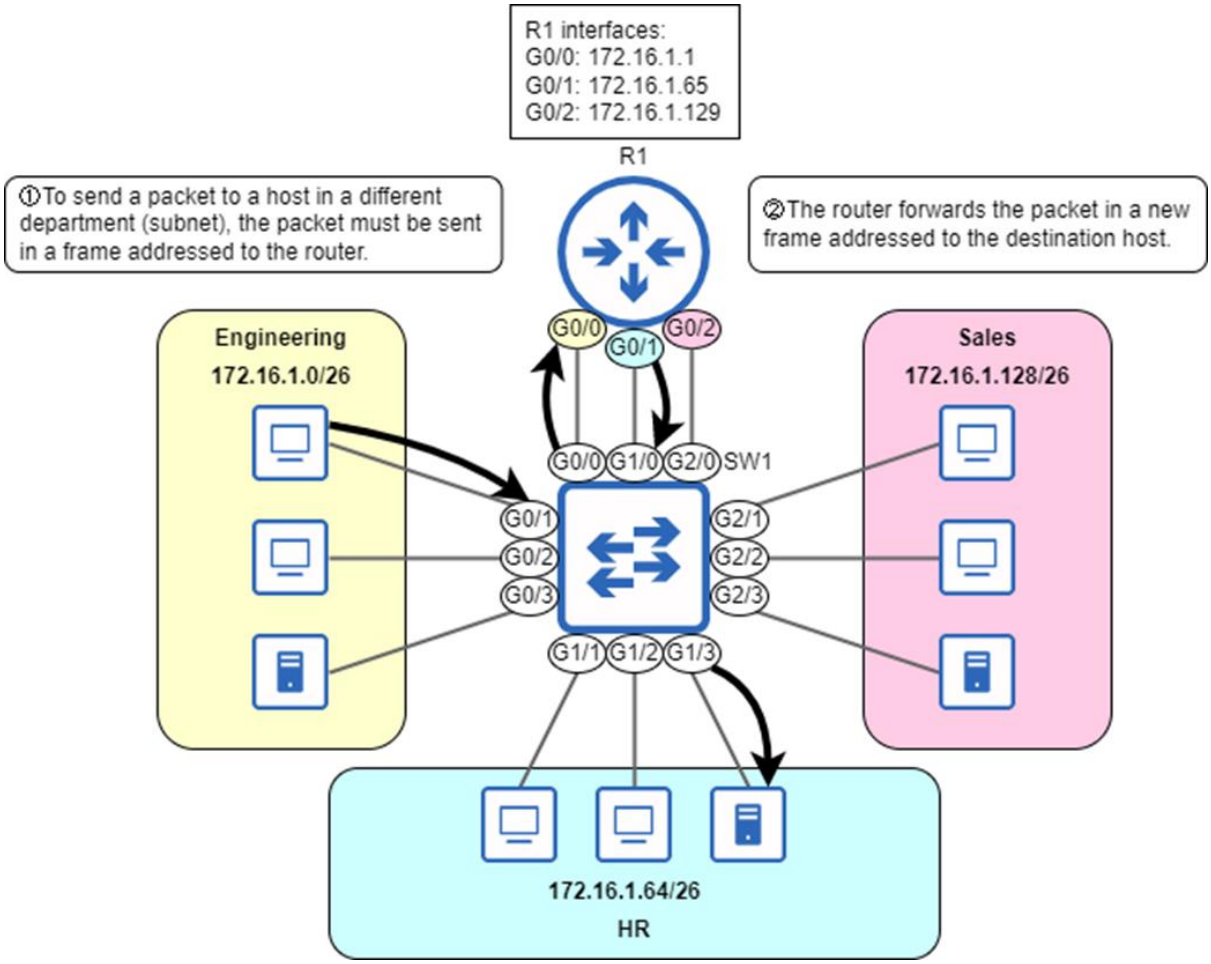


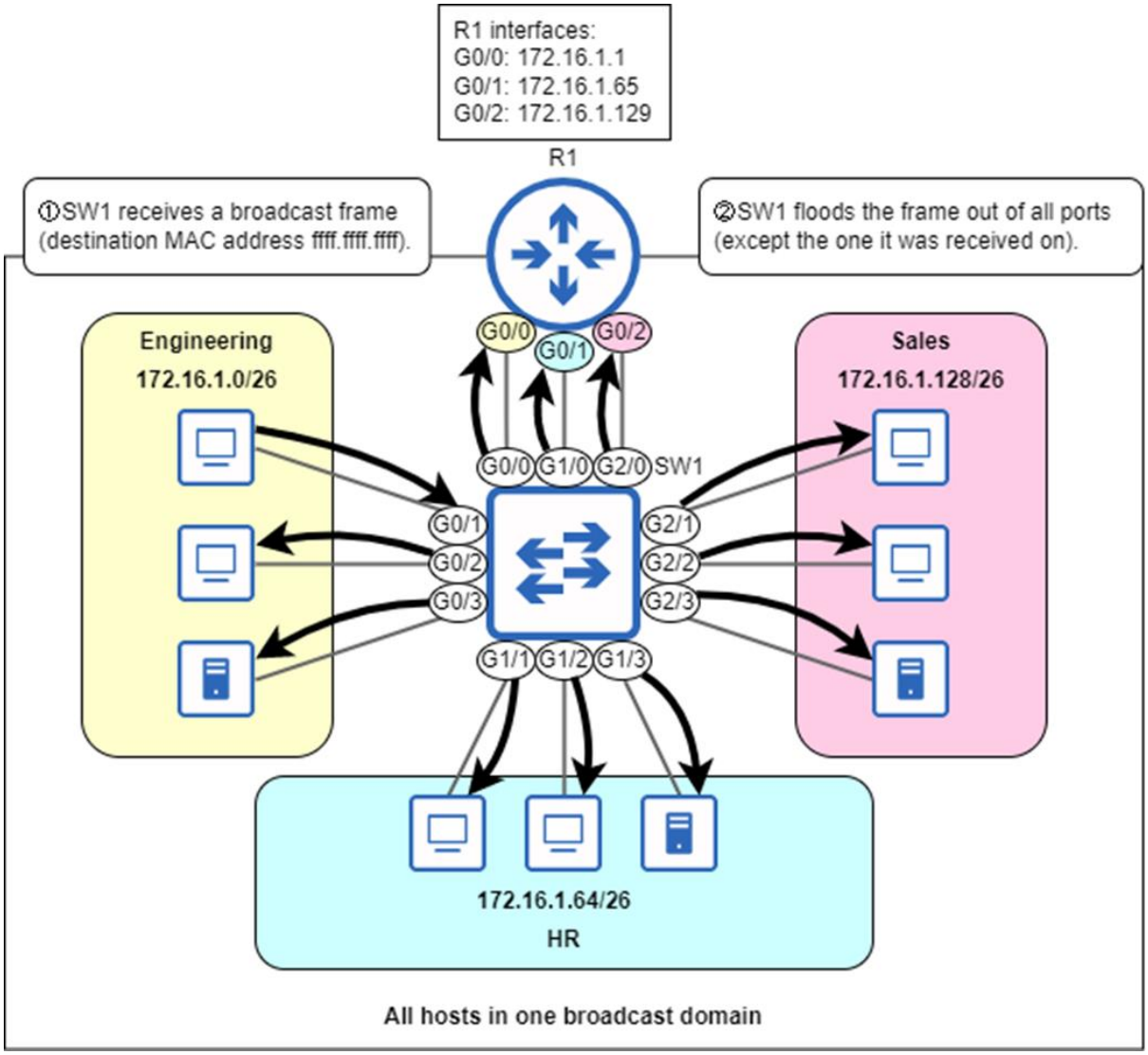

**Figure 12.2 A LAN segmented into three subnets. The engineering department uses subnet 172.16.1.0/26, the HR department uses subnet 172.16.1.64/26, and the sales department uses subnet 172.16.1.128/26. R1 has one interface in each subnet. Communication between hosts in different departments must go through R1.**




**Note:** You might be wondering how segmenting the LAN into separate subnets enhances security. By requiring traffic between departments to pass through the router, you can control which traffic is permitted and which is not; security policies can be implemented on the router to control traffic. Figure 12.2 depicted a PC in the engineering department accessing a server used by the HR department; this is an example of traffic you might want to restrict. You could choose to block all hosts outside the HR department from accessing the server or only allow specific types of communication with the server.

### 12.1.2 Layer 2 segmentation with VLANs

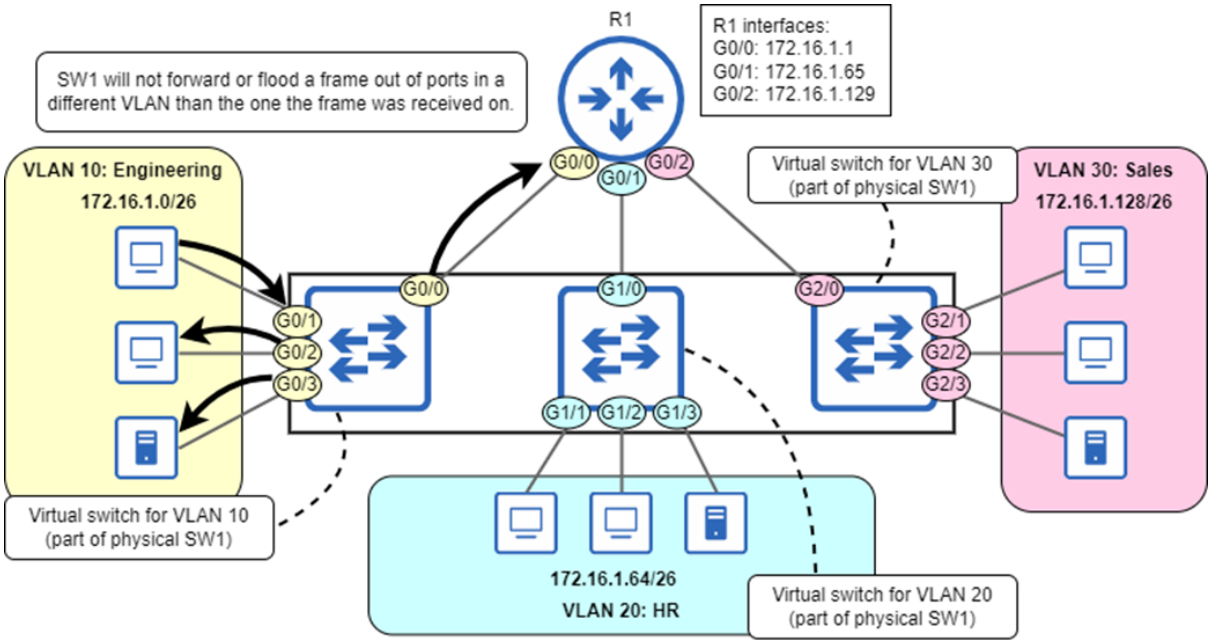
**Figure 12.3 Although hosts are divided into three subnets, at layer 2 they are still part of the same broadcast domain (LAN). 1) SW1 receives a broadcast frame from a host in the engineering department. 2) SW1 floods the frame out of all ports, except the one it was received on. The LAN has been segmented at layer 3, but not at layer 2.**




**Note:** From a security perspective, this is still not suitable — traffic from hosts in one subnet can reach hosts in other subnets. Furthermore, all hosts being in the same broadcast domain can have negative effects on network performance; the unnecessary flooding of frames out of all ports can cause or worsen network congestion. To solve these issues, we should segment the network at layer 2, and we can use VLANs to do so.

VLANs allow us to divide a single physical switch into multiple virtual switches, thereby dividing the broadcast domain into multiple broadcast domains. Figure 12.4 demonstrates this concept. By assigning each of SW1's ports to a specific VLAN, SW1 is divided into three virtual switches: one for VLAN 10, one for VLAN 20, and one for VLAN 30.

**Figure 12.4 By assigning SW1's interfaces to three separate VLANs, SW1 is divided into three virtual switches, each a separate broadcast domain. G0/0, G0/1, G0/2, and G0/3 are part of VLAN 10. G1/0, G1/1, G1/2, and G1/3 are part of VLAN 20. G2/0, G2/1, G2/2, and G2/3 are part of VLAN 30. SW1 will not forward or flood a frame out of ports in a different VLAN than the port the frame was received on.**




**Note:** We have now successfully segmented the LAN at both layer 3 (with subnets) and layer 2 (with VLANs). SW1 will not forward or flood frames between VLANs — hosts in separate VLANs can only communicate with each other through R1. As a general rule, there should be a one-to-one relationship between subnets and VLANs, as shown in figure 12.4 — one subnet per VLAN. If you continue your studies beyond the CCNA, you will find cases where there are multiple subnets associated with a single VLAN, but for the CCNA, you can assume that they are one-to-one.

## 12.2 Configuring VLANs and access ports

### 12.2.1 Creating and naming VLANs

SW1# show vlan brief

VLAN Name Status Ports

---- -------------------------------- --------- -------------------------------

1 default active Gi0/0, Gi0/1, Gi0/2, Gi0/3

Gi1/0, Gi1/1, Gi1/2, Gi1/3

Gi2/0, Gi2/1, Gi2/2, Gi2/3

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

SW1# show mac address-table

Mac Address Table

```
----------------------------------------
Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  1    5254.0000.04c5    DYNAMIC     Gi2/0
  1    5254.000e.a694    DYNAMIC     Gi2/1
  1    5254.000f.d41a    DYNAMIC     Gi1/0
  1    5254.0011.fbcf    DYNAMIC     Gi0/1
. . .
```

**Note:** To configure a VLAN, use the vlan vlan-id command from global configuration mode (*vlan-id* is a number). That will take you to VLAN configuration mode, from which you can also configure the VLAN's name with the name vlan-name command. In the following example, I create and name VLANs 10, 20, and 30 on SW1, and then confirm with show vlan brief (leaving VLANs 1002-1005 out of the output to save space).

```
SW1(config)# vlan 10
SW1(config-vlan)# name Engineering
SW1(config-vlan)# vlan 20
SW1(config-vlan)# name HR
SW1(config-vlan)# vlan 30
SW1(config-vlan)# name Sales
SW1(config-vlan)# end
SW1# show vlan brief
VLAN Name                         Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                      active    Gi0/0, Gi0/1, Gi0/2, Gi0/3
                                            Gi1/0, Gi1/1, Gi1/2, Gi1/3
                                            Gi2/0, Gi2/1, Gi2/2, Gi2/3
10   Engineering                   active
20   HR                           active
30   Sales                        active
. . .
```

**Note:** In the previous example, the status of each VLAN is active. However, you can temporarily disable a VLAN by using the shutdown command in VLAN configuration mode.

In the following example, I disable VLAN 10 on SW1 and confirm with show vlan brief. Notice that the status changes to act/lshut (active, locally shutdown).

```
SW1(config)# vlan 10

SW1(config-vlan)# shutdown

SW1(config-vlan)# end

SW1# show vlan brief

VLAN Name                              Status    Ports

---- -------------------------------- --------- -------------------------------

. . .

10   Engineering                       act/lshut

. . .
```

### 12.2.2 Assigning ports to VLANs

**Configure SW1's ports in access mode.**

1. Configure the access mode VLAN of the ports.
2. An *access port* is a switch port that belongs to a single VLAN, as opposed to a *trunk port*, which carries traffic in multiple VLANs (we will cover trunk ports in section 12.3). By default, Cisco switch ports use a protocol called *Dynamic Trunking Protocol* (DTP) to automatically determine whether each port should operate in access mode or trunk mode. We will cover DTP in chapter 13, but for now just know that it is best practice to manually configure access or trunk mode, rather than letting DTP automatically determine interfaces' status.

You can manually configure a switch port to operate in access mode with the switchport mode access command in interface configuration mode. Then, use the switchport access vlan vlan-id command to configure which VLAN the port belongs to. In the following example, I configure SW1's G0/0, G0/1, G0/2, and G0/3 interfaces as access ports in VLAN 10, its G1/0, G1/1, G1/2, and G1/3 interfaces as access ports in VLAN 20, and its G2/0, G2/1, G2/2, and G2/3 ports as access ports in VLAN 30.

```
SW1(config)# interface range g0/0-3

SW1(config-if-range)# switchport mode access

SW1(config-if-range)# switchport access vlan 10

SW1(config-if-range)# interface range g1/0-3

SW1(config-if-range)# switchport mode access

SW1(config-if-range)# switchport access vlan 20

SW1(config-if-range)# interface range g2/0-3

SW1(config-if-range)# switchport mode access
```

SW1(config-if-range)# switchport access vlan 30

**Note:** We have now finished configuring SW1! It will forward and flood frames between hosts in each VLAN, but not between VLANs — each VLAN is a separate broadcast domain. Keep in mind that VLANs are configured on the switch ports; although it's common to say that an end host is *in VLAN X*, that host is not actually aware of what VLAN it is in — VLANs are a concept used by switches, not end hosts like PCs.

### 12.3 Connecting switches with trunk ports

**Figure 12.5 SW1 and SW2 are connected by a trunk link, which can carry traffic in multiple VLANs. SW1 and SW2 are two physical switches, each consisting of three virtual switches — one for each VLAN. 1) PC1 (connected to SW1) sends a frame addressed to PC10's MAC. 2) SW1 forwards the frame out of its G0/0 port, which is in trunk mode. It adds a tag to the frame, indicating that the frame is in VLAN 10. 3) SW2 forwards the frame out of its G0/1 port (untagged).**

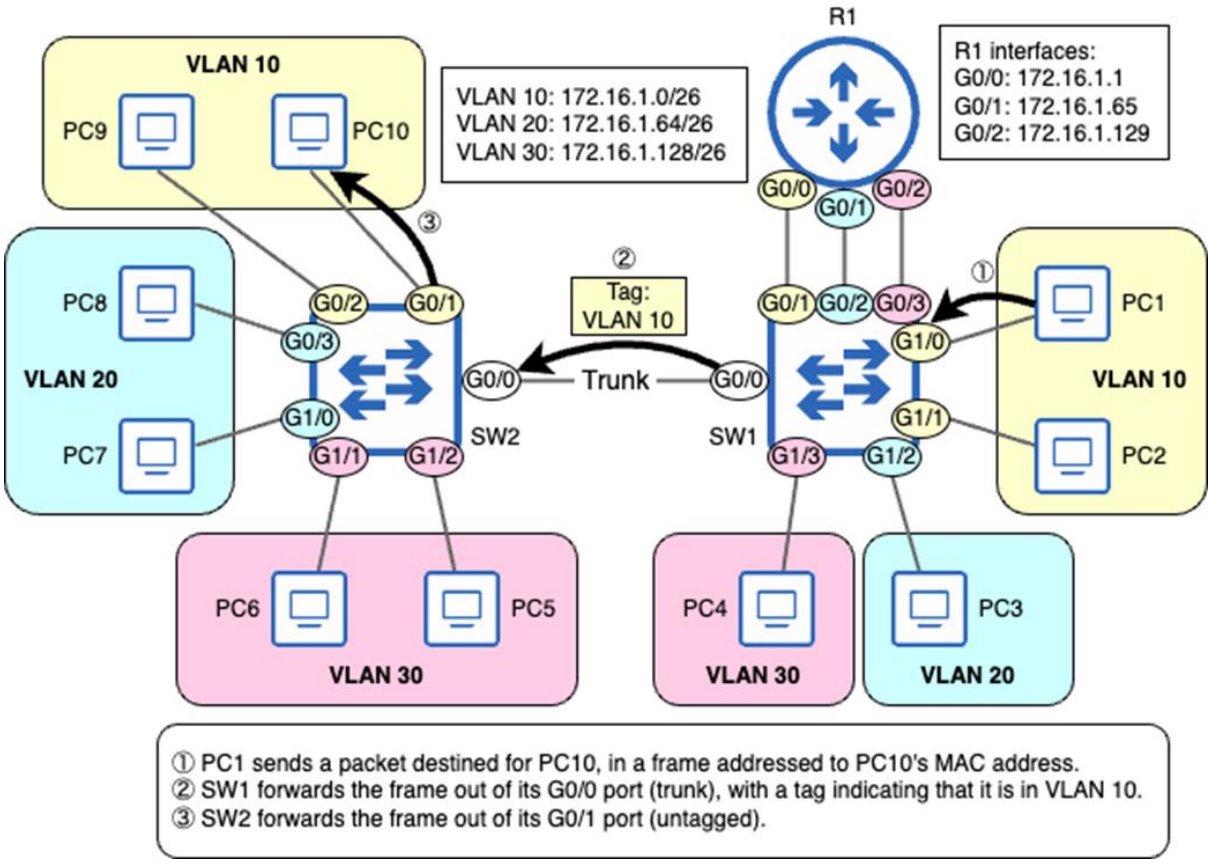
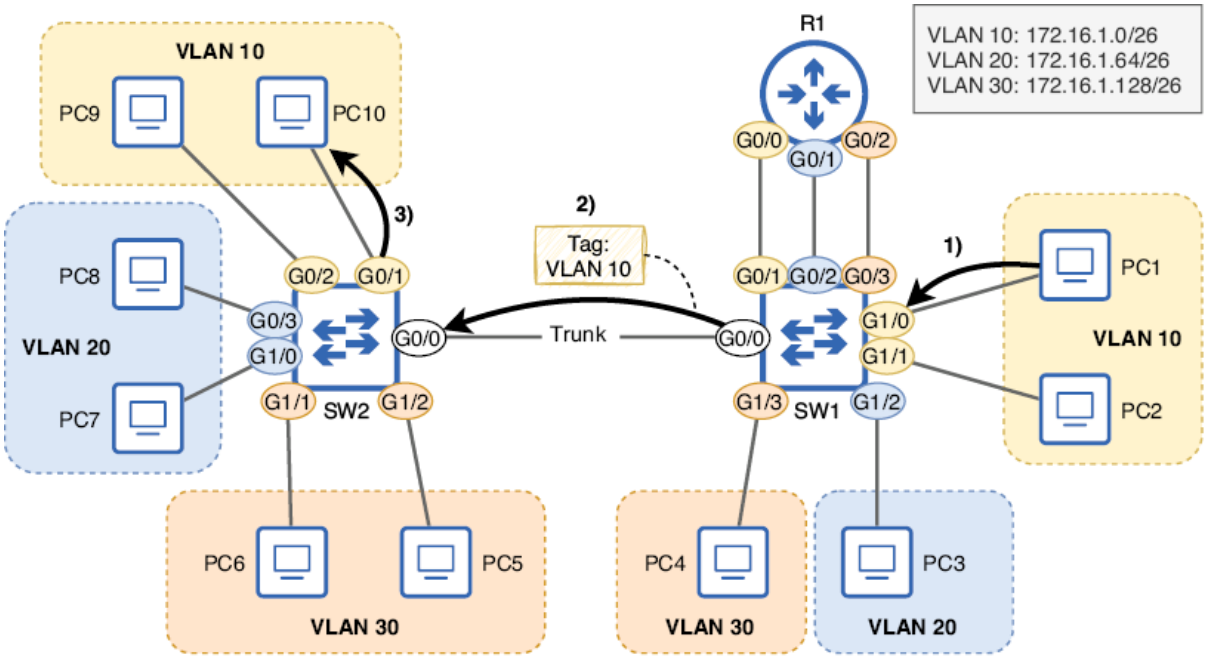


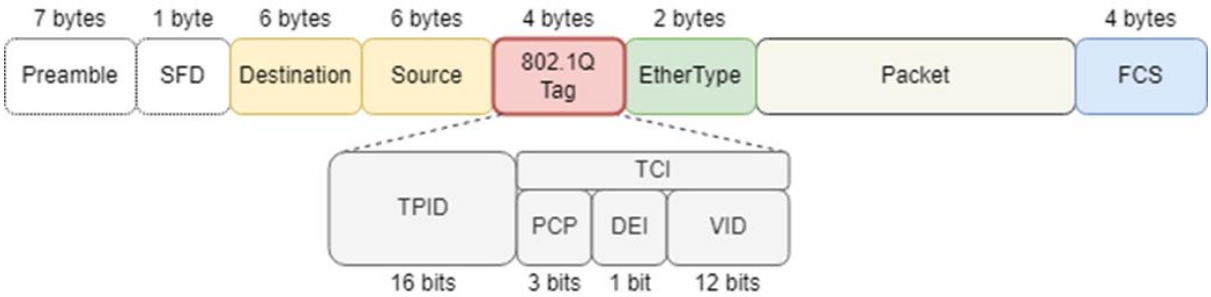

**Figure 12.5 SW1 and SW2 are connected by a trunk link, which can carry traffic in multiple VLANs. SW1 and SW2 are two physical switches, each consisting of three virtual switches—one for each VLAN. (1) PC1 (connected to SW1) sends a frame addressed to PC10's MAC. (2) SW1 forwards the frame out of its G0/0 port, which is in trunk mode. It adds a tag to the frame, indicating that the frame is in VLAN 10. (3) SW2 forwards the frame out of its G0/1 port (untagged).**




**Note:** That's how trunk ports work: the switch forwarding a frame adds a tag before sending it out of the trunk port; for that reason, another name for a trunk port is a *tagged port*. The switch receiving the frame then checks the tag and assigns the frame to the VLAN specified by the tag; if the frame's destination is in a different VLAN than the one specified by the tag, the switch will drop the frame.

Likewise, another name for an access port is an *untagged port*; frames forwarded by an access port are not tagged to indicate the VLAN, and frames received by an access port are assigned to the VLAN specified in the switchport access vlan command. Because access ports are associated with only one VLAN, a tag is not necessary to identify which VLAN frames that are sent and received by the port belong to.

**Figure 12.6 The 802.1Q tag's position in an Ethernet frame, and the field of the tag. The fields are TPID (Tag Protocol Identifier) and TCI (Tag Control Information). TCI contains three subfields: PCP (Priority Code Point), DEI (Drop Eligible Indicator), and VID (VLAN Identifier).**




The second half of the 802.1Q is the *Tag Control Information* (TCI), which contains three subfields: PCP, DEI, and VID. The *Priority Code Point* (PCP) field is three bits in length and can be used to mark frames as higher or lower in priority; this is used for *Quality of Service* (QoS), a topic we will cover in chapter 35. The *Drop Eligible Indicator* (DEI) field is

a single bit in length, and is also used for QoS; it can be used to indicate frames that can be dropped if the network is congested.

The *VLAN Identifier* (VID) field is perhaps the most important; it's the field that indicates which VLAN the frame is in. It is 12 bits in length, and that's why there are 4096 VLANs in total ($2^{12}$=4096).

**Cisco Inter-Switch Link**

ISL is now considered deprecated and is not supported on new Cisco switches. However, you may still encounter Cisco switches that support both 802.1Q and ISL; in such cases, an extra command is required when configuring trunk ports, as we will cover in section 12.3.2. Although you don't have to know ISL itself for the CCNA exam, you should understand how it impacts trunk configuration on switches that support it (by requiring an extra command).

### 12.3.2 Configuring trunk ports

```
SW1(config-if)# switchport trunk encapsulation dot1q
SW1(config-if)# switchport mode trunk
SW1(config-if)#
```

**Note:** After configuring a port as a trunk, it will no longer appear in the output of show vlan brief. The example below demonstrates this; G0/0 is not present in the output. Note that I configured SW1's access ports in their appropriate VLANs, according to figure 12.5.

```
SW1# show vlan brief
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi2/0, Gi2/1, Gi2/2, Gi2/3
10   Engineering                      active    Gi0/1, Gi1/0, Gi1/1
20   HR                               active    Gi0/2, Gi1/2
30   Sales                            active    Gi0/3, Gi1/3


SW1# show interfaces trunk
Port        Mode         Encapsulation  Status        Native vlan
Gi0/0       on           802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094

Port        Vlans allowed and active in management domain
```

Gi0/0 1,10,20,30

. . .

The second part of the output lists the VLANs allowed on each trunk port (Vlans allowed on trunk). As indicated by 1-4094, all VLANs are allowed on a trunk port by default; this means that traffic in all VLANs can be forwarded and received by the port. However, the following part lists the VLANs that are allowed and exist on the switch (Vlans allowed and active in management domain). VLAN 1 exists by default, and I created VLANs 10, 20, and 30, so those four are listed here. If a VLAN does not exist on a switch, it cannot forward traffic in that VLAN; therefore, although all VLANs are allowed on the trunk, SW1 can only forward traffic in VLANs 1, 10, 20, and 30.

**Note:** The *management domain* referred to in the line Vlans allowed and active in management domain is a reference to the *VLAN Trunking Protocol* (VTP) domain. VTP is one of the topics of chapter 13, so I won't mention it any further in this chapter.

Although all VLANs are allowed on a trunk port by default, it is considered best practice to allow only the necessary VLANs. This can help to limit the size of broadcast domains; if a VLAN isn't allowed on a trunk, broadcast (and unknown unicast) frames in that VLAN won't be flooded out of the interface. The command to configure the list of VLANs allowed on the trunk is switchport trunk allowed vlan, and then there are several possible keywords and arguments as shown in the following example:

**Modifying the list of allowed VLANs**

WORD allows you to specify the list of VLANs allowed on the trunk as an argument, such as switchport trunk allowed vlan 10,20,30; this will allow only VLANs 10, 20, and 30 on the trunk; this is the desired state for the network we saw in figure 12.5, which uses only VLANs 10, 20, and 30. I demonstrate this configuration in the following example:

SW1(config-if)# switchport trunk allowed vlan

WORD VLAN IDs of the allowed VLANs when this port is

in trunking mode

add add VLANs to the current list

all all VLANs

except all VLANs except the following

none no VLANs

remove remove VLANs from the current list

The other options are keywords, and for the CCNA exam, it's important to understand how each keyword functions. add and remove are used to modify the current list of allowed VLANs. In the following example, I add VLAN 1, and remove VLAN 30 from the list of allowed VLANs; the list of allowed VLANs then changes to 1, 10, and 20.

```
SW1(config-if)# switchport trunk allowed vlan 10,20,30
SW1(config-if)# do show interfaces trunk
. . .
Port        Vlans allowed on trunk
Gi0/0       10,20,30
. . .
```

The all and none keywords are self-explanatory; all allows all VLANs (the default setting), and none allows no VLANs, preventing the port from forwarding or receiving any traffic. In the following example, I demonstrate both keywords:

```
SW1(config-if)# switchport trunk allowed vlan add 1
SW1(config-if)# switchport trunk allowed vlan remove 30
SW1(config-if)# do show interfaces trunk
. . .
Port        Vlans allowed on trunk
Gi0/0       1,10,20
. . .
```

The final keyword is except, which allows all VLANs except the VLAN(s) you specify as an argument. In the following example, I return the list of allowed VLANs to the desired state (allowing only VLANs 10, 20, and 30) by using the except keyword and specifying all VLANs except 10, 20, and 30 (a bit unconventional, but this is just a demonstration!).

```
SW1(config-if)# switchport trunk allowed vlan all
SW1(config-if)# do show interfaces trunk
. . .
Port        Vlans allowed on trunk
Gi0/0       1-4094
. . .
SW1(config-if)# switchport trunk allowed vlan none
SW1(config-if)# do show interfaces trunk
. . .
Port        Vlans allowed on trunk
Gi0/0       none
. . .
```

**Don’t forget add!**

SW1(config-if)# switchport trunk allowed vlan except

1-9,11-19,21-29,31-4094

SW1(config-if)# do show interfaces trunk

. . .

Port Vlans allowed on trunk

Gi0/0 10,20,30

. . .

**The native VLAN**

As mentioned in section 12.2, access ports (untagged ports) send and receive frames without 802.1Q tags. If an access port receives a tagged frame on an untagged port, it will drop the frame. Trunk ports (tagged ports), on the other hand, send and receive frames with 802.1Q tags to indicate which VLAN each frame belongs to, but what happens if a switch receives an untagged frame on a trunk port? The native VLAN is the answer to that question.

SW1# show interfaces trunk

Port Mode Encapsulation Status Native vlan

Gi0/0 on 802.1q trunking 1

. . .

SW1(config-if)# **switchport trunk native vlan 30** SW1(config-if)# **show interfaces trunk**
Port Mode Encapsulation Status Native vlan
Gi0/0 on 802.1q trunking 30 . . .

Figure 12.7 shows how traffic in the native VLAN is forwarded over a trunk link. The frame from PC1 to PC10 (both in VLAN 10) is tagged when SW1 forwards it to SW2. The frame from PC4 to PC5 (both in VLAN 30), however, is not tagged when SW1 forwards it to SW2; VLAN 30 is the native VLAN on SW1’s G0/0 port. Likewise, VLAN 30 is the native VLAN on SW2’s G0/0 port, so when the frame is received by SW2, SW2 assigns the frame to VLAN 30 and forwards it to the destination (which is also in VLAN 30).

SW1(config-if)# switchport trunk native vlan 30

SW1(config-if)# show interfaces trunk

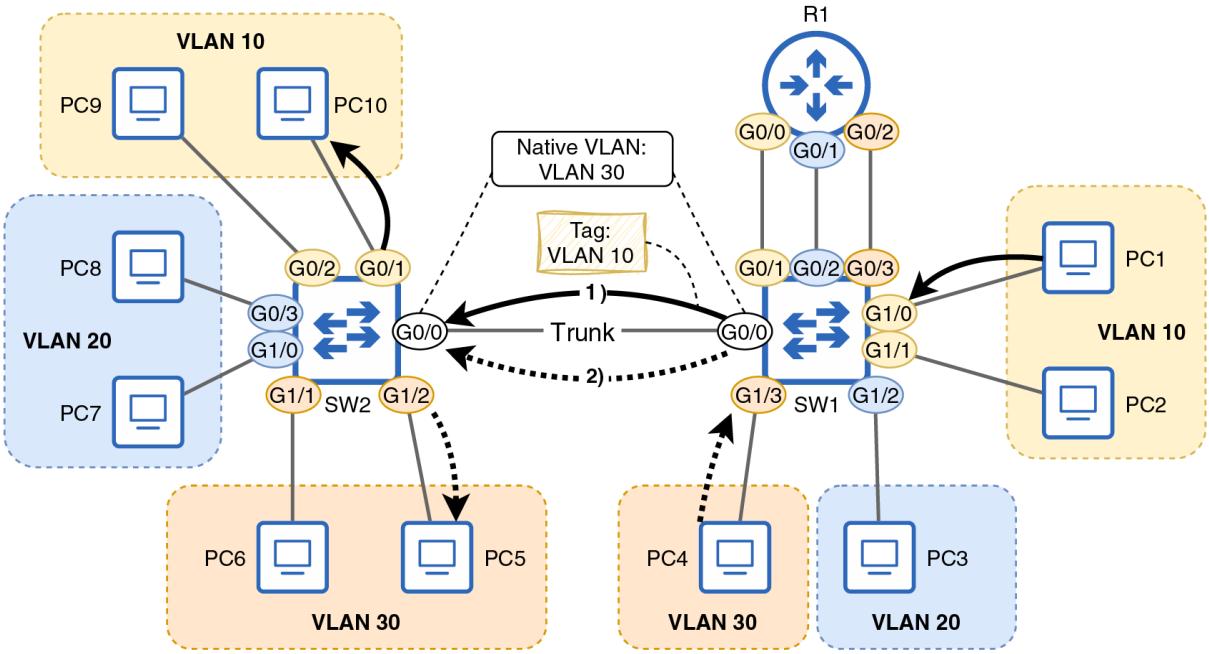
Port Mode Encapsulation Status Native vlan

Gi0/0 on 802.1q trunking 30

. . .

**Figure 12.7 Frames forwarded over a trunk link in the native VLAN and a non-native VLAN. (1) PC1’s frame to PC10 is tagged over the trunk link because VLAN 10 is not**

**the native VLAN. (2) PC4's frame to PC5 is untagged over the trunk link because VLAN 30 is the native VLAN.**



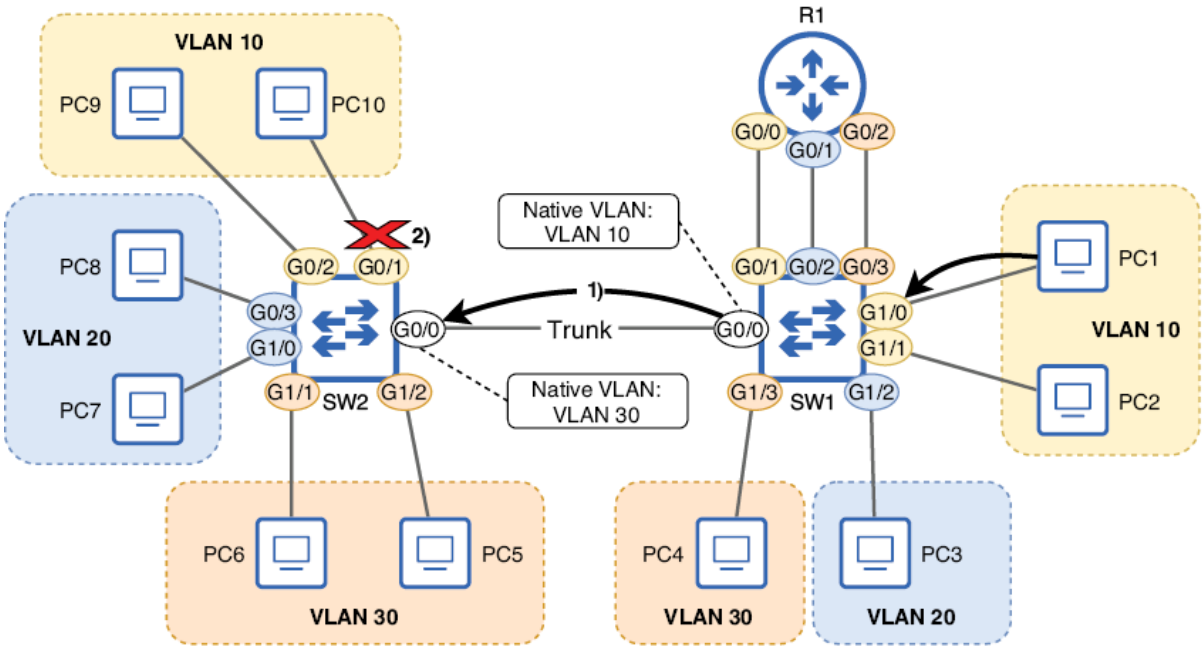

### Note

Because the native VLAN is configured on each switch's ports, it is possible to configure a different native VLAN on each end of a link. However, this is a misconfiguration, and should not be done. Make sure the native VLAN matches on both ends of the link! Figure 12.8 shows one example of what can happen when there is a native VLAN mismatch. SW1 G0/0's native VLAN is 10, but SW2 G0/0's native VLAN is 30. When PC1 sends a frame to PC10, SW1 forwards the frame untagged to SW2. However, when SW2 receives the untagged frame it assigns the frame to VLAN 30 (SW2 G0/0's native VLAN). Because the frame's destination is connected to SW2's G0/1 (an access port in VLAN 10), SW2 cannot forward the frame; SW2 must drop the frame. When traffic crosses from one VLAN to another like this, it is called *VLAN hopping*.

### Native VLAN mismatch

In addition to resulting in dropped frames, VLAN hopping can also cause frames to reach destinations in the wrong VLAN. Following figure 12.8's example, if PC1 (VLAN 10) sends a broadcast frame, SW1 will flood the frame out of G0/0 (as well as G0/1 and G1/1). SW2, upon receiving the untagged broadcast frame, will assign it to VLAN 30. This time the frame's destination is not in any particular VLAN — it's the broadcast MAC address (ffff.ffff.ffff). As a result, SW2 will flood the frame out of G1/1 and G1/2, which are in a different VLAN (VLAN 30) than the frame's sender (PC1, in VLAN 10).

**Figure 12.8 A native VLAN mismatch resulting in frames not reaching their destination. SW1 G0/0's native VLAN is VLAN 10, and SW2 G0/0's is VLAN 30. (1) PC1's frame to PC10 is untagged over the trunk link because VLAN 10 is SW1 G0/0's native VLAN. (2) When SW2 receives the frame, it assigns the frame to VLAN 30 (SW2 G0/0's native VLAN) and therefore cannot forward the frame to its destination (in VLAN 10).**




Cisco switches typically run *Per-VLAN Spanning Tree Plus* (PVST+) or *Rapid Per-VLAN Spanning Tree Plus* (Rapid-PVST+). If there is a native VLAN mismatch, these protocols will prevent traffic from being forwarded over the trunk in the mismatched VLANs, and display a message indicating so. We will cover PVST+ and Rapid-PVST+ in chapters 14 and 15, respectively. *Cisco Discovery Protocol* (CDP) can also detect native VLAN mismatches, but will not block traffic in the mismatched VLANs; it will only display messages indicating the mismatch. We will cover CDP in chapter 26.

**Note:** The native VLAN was developed to accommodate devices that do not support 802.1Q tagging, such as hubs. However, these days there is usually no need to use the native VLAN, and its use can render the network vulnerable to a *VLAN hopping attack* using *double tagging* (topics for part 9 of this book: Security Fundamentals).

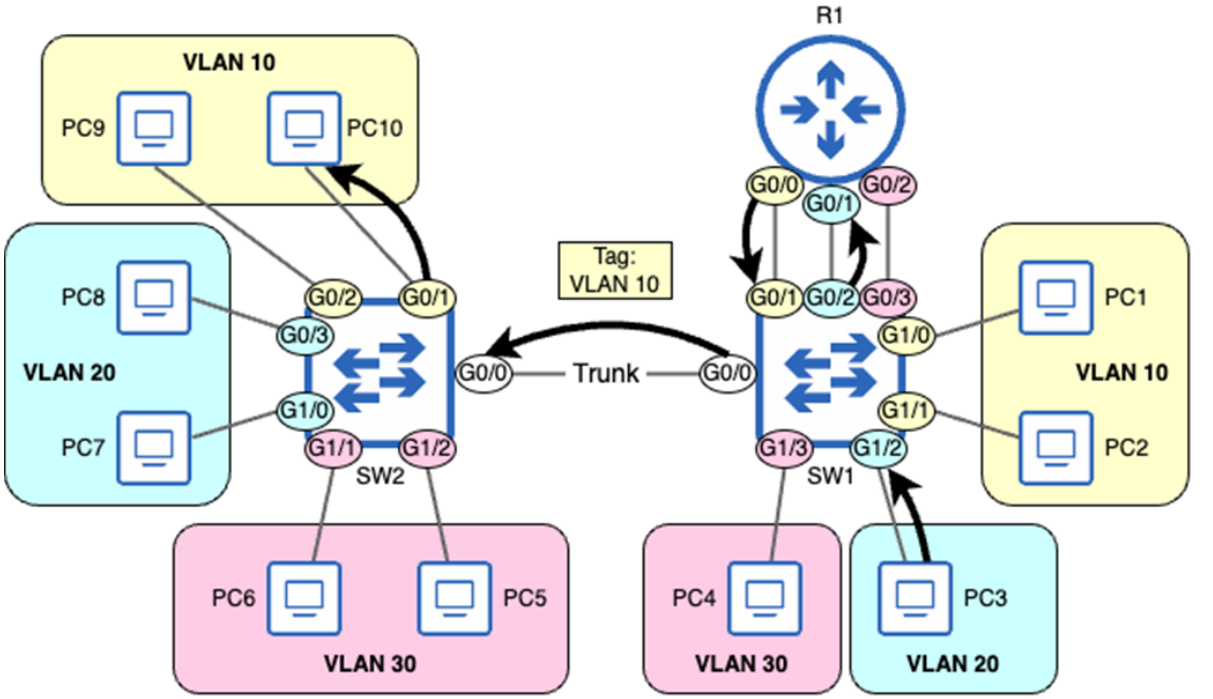
**Exam Tip:** Remember that as a best practice for security: configure an unused VLAN (that isn't the default of VLAN 1) as the native VLAN on your trunk ports.

Even after segmenting a LAN into multiple subnets and VLANs, we usually still want the subnets/VLANs to be able to communicate with each other (and external networks). Although routing is a layer 3 concept, and VLANs are a layer 2 concept, the term *inter-VLAN routing* is used to refer to routing between subnets in a LAN that is segmented using VLANs.

### 12.4 Inter-VLAN routing

The following examples show how R1 and SW1 can be configured to enable inter-VLAN routing in this manner:

**Figure 12.9 PC3 (in VLAN 20) sends a packet to PC10 (in VLAN 10). 1) PC3 sends the packet in a frame addressed to its default gateway (R1 G0/1). SW2 forwards it out of its G0/2 port (untagged). 2) R1 routes the packet, forwarding it out of G0/0 in a new frame addressed to PC10. The frame is forwarded to PC10 by SW1 and SW2. It is tagged only when crossing the trunk link from SW1 G0/0 to SW 2 G0/0.**




However, this method of inter-VLAN routing is not common for the same reason it's not common to connect switches using access ports: in a LAN with many VLANs, you'll soon run out of physical ports on your devices. Instead, one of the following options is usually preferred:

R1(config)# interface g0/0

R1(config-if)# ip address 172.16.1.1 255.255.255.192

R1(config-if)# no shutdown

R1(config-if)# interface g0/1

R1(config-if)# ip address 172.16.1.65 255.255.255.192

R1(config-if)# no shutdown

R1(config-if)# interface g0/2

R1(config-if)# ip address 172.16.1.129 255.255.255.192

R1(config-if)# no shutdown

SW1(config)# interface g0/1

SW1(config-if)# switchport mode access

SW1(config-if)# switchport access vlan 10

SW1(config-if)# interface g0/2

SW1(config-if)# switchport mode access

SW1(config-if)# switchport access vlan 20

SW1(config-if)# interface g0/3

SW1(config-if)# switchport mode access

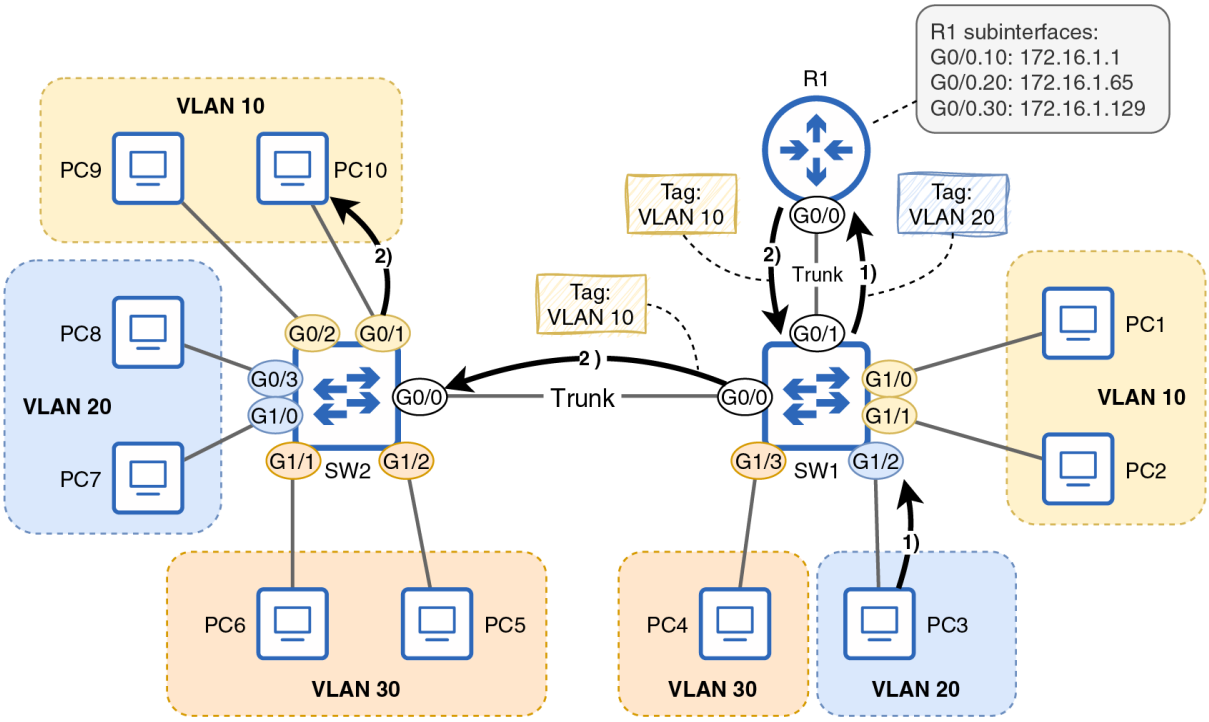
SW1(config-if)# switchport access vlan 30

Multilayer switch (a switch that can also route packets)

### 12.4.1 Router on a stick

*Router on a stick* (ROAS) is a method of inter-VLAN routing that involves creating a trunk link between a switch and a router; a single physical router interface can be divided into multiple virtual *subinterfaces*, each with its own IP address. These subinterfaces send and receive tagged frames, like a trunk port on a switch. Figure 12.10 shows how the same packet from PC3 to PC10 can be routed using ROAS.

**Figure 12.10 PC3 (in VLAN 20) sends a packet to PC10 (in VLAN 10), and the packet is routed using the router on a stick method. R1's G0/0 interface has three subinterfaces: G0/0.10 (VLAN 10, 172.16.1.1), G0/0.20 (VLAN 20, 172.16.1.65), and G0/0.30 (VLAN 30, 172.16.1.129). PC3's frame to R1 is tagged in VLAN 20 over the trunk link from SW1 G0/1 and R1 G0/0. R1's frame to PC10 is tagged in VLAN 10 over the trunk link from R1 G0/0 to SW1 G0/1, and the trunk link from SW1 G0/0 to SW2 G0/0.**




**Note:** Let's see how to configure ROAS as shown in figure 12.10. SW1's side of the connection is a trunk port, just like we configured in section 12.3. In the following example I configure SW1 G0/1 as a trunk port, allow only the necessary VLANs, and change the native VLAN to an unused VLAN.

**Configuring ROAS**

SW1(config)# interface g0/1

SW1(config-if)# switchport trunk encapsulation dot1q

SW1(config-if)# switchport mode trunk

SW1(config-if)# switchport trunk allowed vlan 10,20,30

SW1(config-if)# switchport trunk native vlan 999

R1(config)# interface g0/0

R1(config-if)# no shutdown

R1(config-if)# interface g0/0.10

R1(config-subif)#

After the above configurations, any frames R1 receives on its G0/0 interface that are tagged with VLAN 10 will be sent to the G0/0.10 subinterface, and any frames sent by the VLAN 10 subinterface will be tagged with VLAN 10.

R1(config-subif)# encapsulation dot1q 10

R1(config-subif)# ip address 172.16.1.1 255.255.255.192

The number used to identify the subinterface (the .10 in G0/0.10) does not have to match the VLAN ID; the number has no significance beyond identifying the subinterface. It's the encapsulation dot1q command that tells the router which VLAN to associate with this subinterface. However, I recommend that you match these two numbers; there's no reason not to.

R1(config-subif)# interface g0/0.20

R1(config-subif)# encapsulation dot1q 20

R1(config-subif)# ip address 172.16.1.65 255.255.255.192

R1(config-subif)# interface g0/0.30

R1(config-subif)# encapsulation dot1q 30

R1(config-subif)# ip address 172.16.1.129 255.255.255.192

R1(config-subif)# do show ip interface brief

| Interface | IP-Address | OK? | Method | Status | Protocol |
|---|---|---|---|---|---|
| GigabitEthernet0/0 | unassigned | YES | manual | up | up |
| GigabitEthernet0/0.10 | 172.16.1.1 | YES | manual | up | up |

GigabitEthernet0/0.20 172.16.1.65 YES manual up up

GigabitEthernet0/0.30 172.16.1.129 YES manual up up

. . .

The ROAS configuration is now complete; R1 can route traffic between the three subnets/VLANs in the LAN, using the single physical trunk connection with SW1. Note that I didn't do any configurations related to the native VLAN on R1's side of the connection; if not using the native VLAN, there is no need to do any particular configurations on the router.

If you decide to use the native VLAN over the ROAS trunk, there are two methods to configure the router's side of the connection:

**Configuring the native VLAN with ROAS**

Configure the IP address for the native VLAN on the physical interface, not a subinterface

Let's try both. In the following example, I show the ROAS configuration once again, this time configuring VLAN 10 as the native VLAN by adding the native keyword to the encapsulation dot1q command. Aside from that, the configurations are identical to the previous examples.

In the following example, I use the second method of configuring the native VLAN on the router. I don't configure a subinterface for VLAN 10, but rather configure the native VLAN's IP address on the G0/0 interface itself; the encapsulation dot1q command is not necessary for VLAN 10 in this case, although it's still needed on the subinterfaces of the non-native VLANs (VLANs 20 and 30).

R1(config)# interface g0/0

R1(config-if)# no shutdown

R1(config-if)# interface g0/0.10

R1(config-subif)# encapsulation dot1q 10 native

R1(config-subif)# ip address 172.16.1.1 255.255.255.192

R1(config-subif)# interface g0/0.20

R1(config-subif)# encapsulation dot1q 20

R1(config-subif)# ip address 172.16.1.65 255.255.255.192

R1(config-subif)# interface g0/0.30

R1(config-subif)# encapsulation dot1q 30

R1(config-subif)# ip address 172.16.1.129 255.255.255.192

R1(config)# interface g0/0

R1(config-if)# no shutdown

```
R1(config-if)# ip address 172.16.1.1 255.255.255.192
R1(config-if)# interface g0/0.20
R1(config-subif)# encapsulation dot1q 20
R1(config-subif)# ip address 172.16.1.65 255.255.255.192
R1(config-subif)# interface g0/0.30
R1(config-subif)# encapsulation dot1q 30
R1(config-subif)# ip address 172.16.1.129 255.255.255.192
```

**Note:** The third option for inter-VLAN routing, and perhaps the most popular (although ROAS is common as well), is to use a multilayer switch. A *multilayer switch* (also called a *layer 3 switch*) is a switch that is also capable of routing packets; it's a switch with a router built in.

### 12.4.2 Multilayer switching

A standard switch that only forwards frames can be called a *layer 2 switch*. However, these days almost all switches have some degree of layer 3 capabilities, so the difference between a multilayer switch and a layer 2 switch is often determined by how you use the switch, rather than the switch itself.

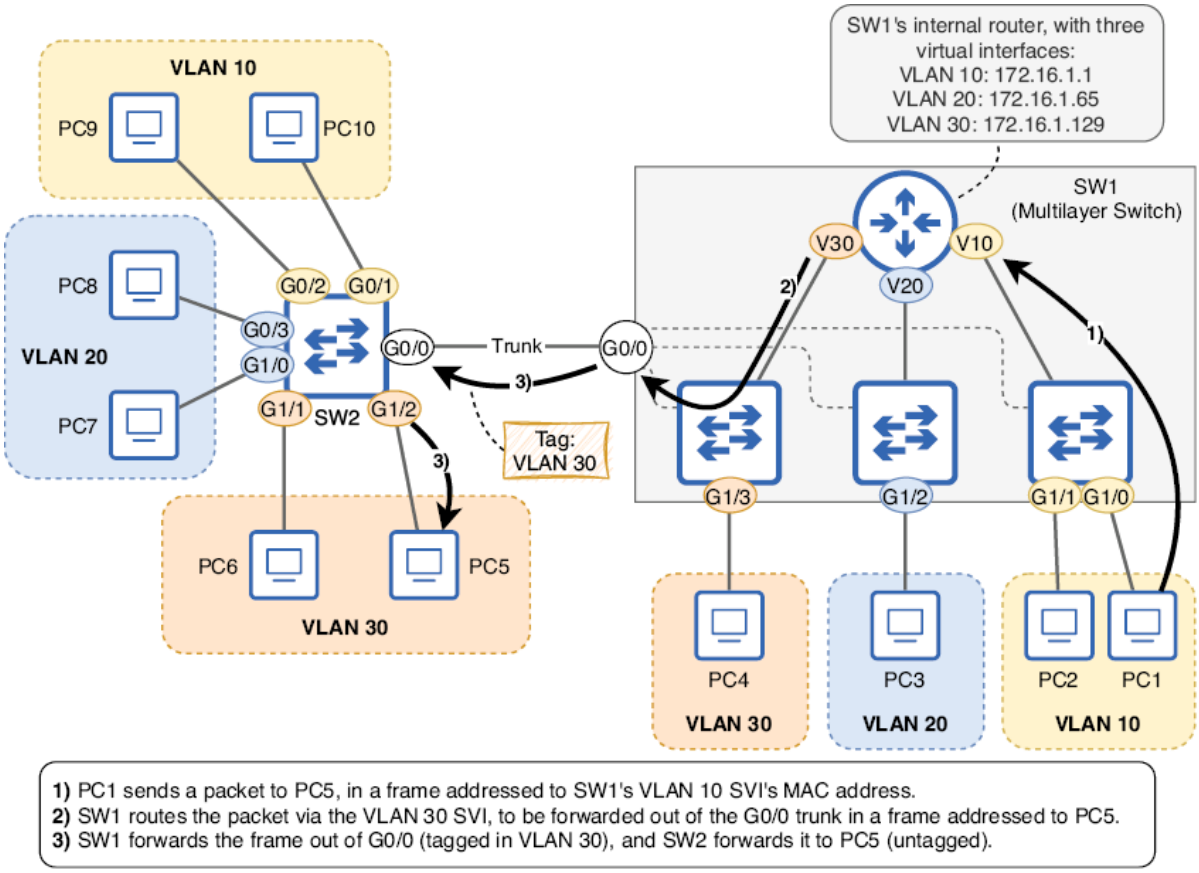
Multilayer switches perform inter-VLAN routing using virtual interfaces called *switch virtual interfaces* (SVIs). Each SVI is an interface on the multilayer switch's built-in router, and hosts in each VLAN use the IP address of their VLAN's SVI as their default gateway. Figure 12.11 shows the internal logic of how SW1 (now a multilayer switch) routes a packet from PC1 to PC8. PC1 sends the packet in a frame addressed to SW1's VLAN 10 SVI (each SVI has a unique MAC address). SW1's internal router routes the packet via the VLAN 20 SVI and forwards it out of the G0/0 trunk port in a frame (tagged in VLAN 20) addressed to PC8's MAC, and SW2 forwards the frame to PC8 (untagged).

**Inter-VLAN routing via SVIs**

**Note**

R1 is no longer present in the figure 12.11 diagram; if we configure SVIs on SW1, there is no need to rely on an external router for inter-VLAN routing.

**Figure 12.11 SW1, a multilayer switch, routes a packet from PC1 to PC5. SW1 has three SVIs: VLAN 10 (172.16.1.1), VLAN 20 (172.16.1.65), and VLAN 30 (172.16.1.129), allowing SW1 to route packets internally, without relying on an external router.**




**Note**

On some switches, SVIs may be administratively disabled by default. In that case, use no shutdown to enable each SVI.

SW1(config)# ip routing

SW1(config)# interface vlan 10

SW1(config-if)# ip address 172.16.1.1 255.255.255.192

SW1(config-if)# interface vlan 20

SW1(config-if)# ip address 172.16.1.65 255.255.255.192

SW1(config-if)# interface vlan 30

SW1(config-if)# ip address 172.16.1.129 255.255.255.192

**Note**

For an SVI to function, it must be in an up/up state (referring to the Status and Protocol columns in the output of show ip interface brief), just like a physical interface. For an SVI to be in an up/up state, there are four requirements; refer to this list if you need to troubleshoot an SVI that won't reach an up/up state:

The VLAN associated with the SVI must exist on the switch (must be created with the vlan vlan-id command).

SW1# show ip route

. . .

172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks

C 172.16.1.0/26 is directly connected, Vlan10

L 172.16.1.1/32 is directly connected, Vlan10

C 172.16.1.64/26 is directly connected, Vlan20

L 172.16.1.65/32 is directly connected, Vlan20

C 172.16.1.128/26 is directly connected, Vlan30

L 172.16.1.129/32 is directly connected, Vlan30

an access port associated with the VLAN (using the switchport access vlan command) in an up/up state

a trunk port that allows the VLAN (using the switchport trunk allowed vlan command) in an up/up state

The VLAN must be enabled (must not have the shutdown command applied).

The SVI must be enabled (must not have the shutdown command applied).

**Exam Tip**

Make sure you understand the difference between a VLAN and an SVI. A *VLAN* is a layer 2 concept — a virtual broadcast domain that divides up a virtual switch. An *SVI* is a virtual layer 3 interface that is associated with a VLAN. To create a VLAN, use the vlan command. To create an SVI, use the interface vlan command.

As the following example shows, SW1's SVIs are currently in an up/up state:

**Exam Tip**

To demonstrate the requirements, in the following example I violated one requirement for each of the VLAN 10, VLAN 20, and VLAN 30 SVIs: I deleted VLAN 10 from SW1 (requirement 1), disabled VLAN 20 with shutdown (requirement 3), and disabled SW1's G1/3 port (an access port in VLAN 30) and removed VLAN 30 from G0/0's list of allowed VLANs (requirement 2). As a result, all three SVIs move to an up/down state; they will no longer be able to route packets.

SW1(config)# **no vlan 10**

SW1(config)# **vlan 20**

SW1(config-vlan)# **shutdown**

SW1(config-vlan)# **interface g1/3**

SW1(config-if)# **shutdown**

SW1(config-if)# **interface g0/0**

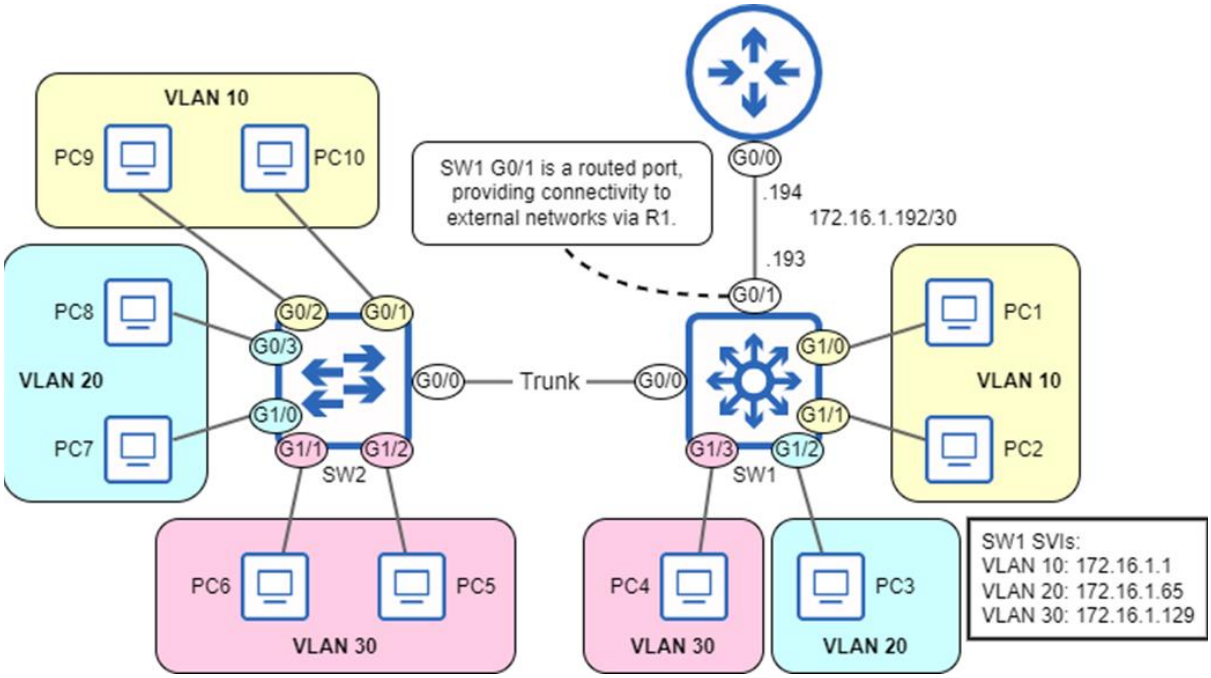
SW1(config-if)# **switchport trunk allowed vlan remove 30**

SW1(config-if)# **do show ip interface brief | include Vlan** SW1# show ip interface brief | include Vlan

| | | | |
|---|---|---|---|
| Vlan10 | 172.16.1.1 | YES manual up | up |
| Vlan20 | 172.16.1.65 | YES manual up | up |
| Vlan30 | 172.16.1.129 | YES manual up | up |

Routing within a LAN is important, but it's also essential for hosts in the LAN to be able to reach external networks such as the Internet, or another LAN in the corporate network. To provide external connectivity, it's common to use a routed port on a multilayer switch.
A *routed port* is a physical port on a multilayer switch that has been configured to function like a router's interface. Figure 12.12 shows how SW1's G0/1 port can be used as a routed port, providing connectivity to external networks via R1.

- **Figure 12.12 SW1, a multilayer switch, uses a routed port (G0/1) to provide connectivity to external networks (via R1, in this case). Like a router interface, SW1 G0/1 is configured with an IP address: 172.16.1.193.**




- **Note**
- SW1's icon in figure 12.12 is a new one. Network diagrams typically use an icon like this to represent multilayer switches, differentiating them from layer 2 switches.

To configure a routed port, use the no switchport command in interface configuration mode; then, you can configure an IP address just like on a router's interface. In the following

example, I configure SW1 G0/1 as a routed port with an IP address, and then check SW1's routing table:

SW1(config)# no vlan 10

SW1(config)# interface g1/3

SW1(config-if)# shutdown

SW1(config-if)# interface g0/0

SW1(config-if)# switchport trunk allowed vlan remove 30

SW1(config-if)# vlan 20

SW1(config-vlan)# shutdown

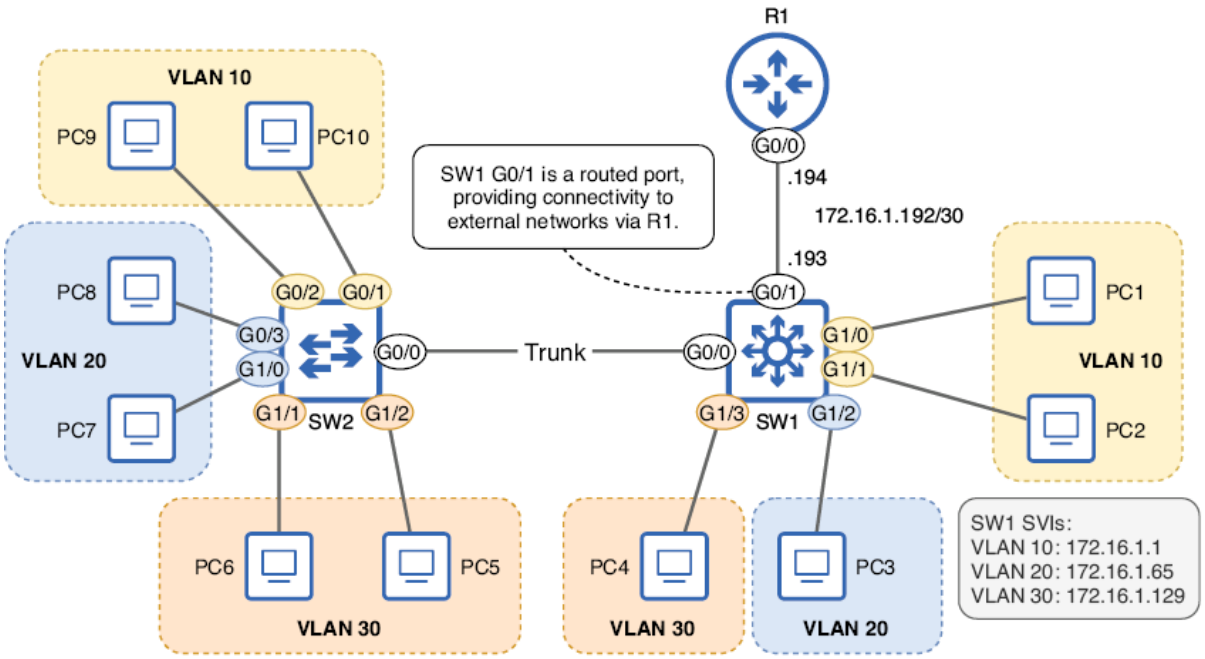
SW1(config-vlan)# do show ip interface brief | include Vlan

| | | | |
|---|---|---|---|
| Vlan10 | 172.16.1.1 | YES manual up | down |
| Vlan20 | 172.16.1.65 | YES manual up | down |
| Vlan30 | 172.16.1.129 | YES manual up | down |

**Using routed ports for external connectivity**

SW1(config)# **ip route 0.0.0.0 0.0.0.0 172.16.1.194**

**Figure 12.12 SW1, a multilayer switch, uses a routed port (G0/1) to provide connectivity to external networks (via R1, in this case). Like a router interface, SW1 G0/1 is configured with an IP address: 172.16.1.193.**




**Note**

**Exam Scenarios**

VLANs are one of the major topics of the CCNA exam, so you can expect a few VLAN-related questions on the CCNA exam. Although the actual questions on the CCNA exam are protected by Cisco's NDA, below are a few questions demonstrating how your understanding of VLANs might be tested on the CCNA exam:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

SW1(config)# **interface g0/1**

SW1(config-if)# **no switchport**

SW1(config-if)# **ip address 172.16.1.193 255.255.255.252**

SW1(config-if)# **do show ip route**

. . .

172.16.0.0/16 is variably subnetted, 8 subnets, 3 masks

C 172.16.1.0/26 is directly connected, Vlan10

L 172.16.1.1/32 is directly connected, Vlan10

C 172.16.1.64/26 is directly connected, Vlan20

L 172.16.1.65/32 is directly connected, Vlan20

C 172.16.1.128/26 is directly connected, Vlan30

L 172.16.1.129/32 is directly connected, Vlan30

C 172.16.1.192/30 is directly connected, GigabitEthernet0/1

L 172.16.1.193/32 is directly connected, GigabitEthernet0/1

**1**

**2**

**3**

copy

Examine the configuration of SW1's G0/0 interface below.

1

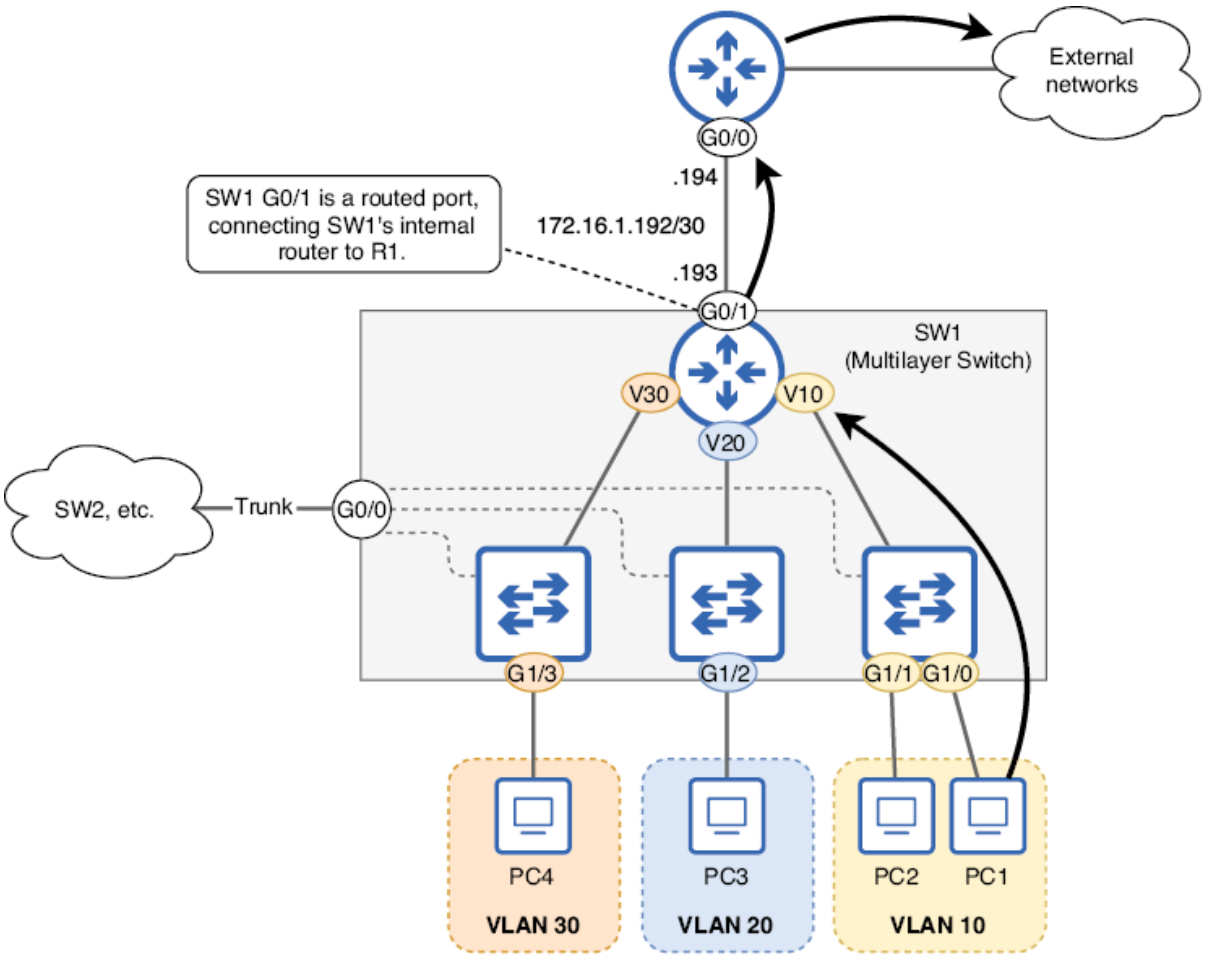
SW1(config)# **ip route 0.0.0.0 0.0.0.0 172.16.1.194**

**1**

copy

Which of the following statements are true? (select two)

**Figure 12.13 A host in the LAN sends a packet to an external destination, routed by SW1. G0/1 is a routed port, connecting SW1's internal router to R1.**




**Exam scenarios**

C) SW1 will assign untagged frames received on G0/0 to VLAN 5.

1. (multiple choice, multiple answers)

The challenging part of this question is that G0/0 has configurations related both to access ports and trunk ports. The switchport access vlan 5 command implies that SW1 will assign untagged frames received on G0/0 to VLAN 5. However, the switchport trunk native vlan 10 command implies that SW1 will assign them to VLAN 10. The key to this question is that the switchport mode command specifies trunk, so G0/0 is operating as a trunk port (and A is one of the correct answers). Therefore, the switchport access vlan 5 command will not affect G0/0; it is only significant if G0/0 is operating as an access port. So, the second correct answer is D; SW1 will assign untagged frames received on G0/0 to VLAN 10 (the native VLAN).

interface GigabitEthernet0/0

switchport access vlan 5

switchport trunk native vlan 10

switchport mode trunk

Which of the following statements are true? (select two)

A. SW1 G0/0 is a trunk port.

B. SW1 G0/0 is an access port.

3) (lab simulation)

A lab simulation might provide you a network diagram, and ask you to configure access ports and trunk ports as appropriate. Remember the basic configurations of each:

Access ports: switchport mode access, switchport access vlan vlan-id

2. (drag and drop)

**12.5 Summary**

| | |
|---|---|
| (A) Related to access ports | Default VLAN |
| (B) Related to trunk ports | |
| (C) VLAN 1 by default, and can be changed | Native VLAN |
| (D) VLAN 1 by default, and cannot be changed | |

VLANs divide a broadcast domain (LAN) into multiple broadcast domains, by dividing a physical switch into multiple virtual switches. Frames sent by a host in one VLAN cannot be forwarded/flooded to hosts in another VLAN.

3. (lab simulation)

Use the show vlan brief command to view the VLANs that exist on the switch, and which ports are in each VLAN.

- VLANs 1 and 1002-1005 exist by default, and cannot be deleted. VLAN 1 is the *default VLAN* — the VLAN that all ports are in by default. VLANS 1002-1005 are reserved for use by *FDDI* and *Token Ring* — two legacy data link layer technologies.
- Use the vlan vlan-id command to create a VLAN, and then the name vlan-name command to give the VLAN an optional name (the default name is *VLANxxxx*). You can use the shutdown command to temporarily disable the VLAN, or no vlan vlan-id to delete the VLAN.

**Summary**

- *Network segmentation* is the process of dividing a network into smaller parts and provides network security and performance benefits. Subnets can be used to segment a network at Layer 3, and *virtual LANs* (VLANs) can be used to segment a network at Layer 2.
- VLANs divide a broadcast domain (LAN) into multiple broadcast domains by dividing a physical switch into multiple virtual switches. Frames sent by a host in one VLAN cannot be forwarded/flooded to hosts in another VLAN.
- Although there can be multiple subnets per VLAN, for the CCNA exam, you can assume a one-to-one relationship (one subnet per VLAN).
- Use the **show vlan brief** command to view the VLANs that exist on the switch and which ports are in each VLAN.
- VLANs 1 and 1002–1005 exist by default and cannot be deleted. VLAN 1 is the *default VLAN*—the VLAN that all ports are in by default. VLANS 1002–1005 are reserved for use by *FDDI* and *Token Ring*—two legacy Data Link Layer technologies.
- Use the **vlan** *vlan-id* command to create a VLAN, and then the **name** *vlan-name* command to give the VLAN an optional name (the default name is *VLANxxxx*). You can use the **shutdown** command to temporarily disable the VLAN or **no vlan** *vlan-id* to delete the VLAN.
- An *access port* is a switch port that sends and receives traffic in a single VLAN. Access ports are also called *untagged ports* because they send and receive frames without VLAN tags.
- Use the **switchport mode access** command to configure a port in access mode. Then, use the **switchport access vlan** *vlan-id* command to configure which VLAN the port belongs to. If you assign a port to a VLAN that doesn't exist yet on the switch, the switch will automatically create the VLAN.

- A *trunk port* is a switch port that sends and receives traffic in multiple VLANs. Trunk ports differentiate between VLANs by adding a VLAN tag to each frame using the *IEEE 802.1Q* protocol.
- The 802.1Q tag is 4 bytes in length and is added between the Source and EtherType fields of the Ethernet header. The main fields of the 802.1Q tag are TPID and TCI.
- The *Tag Protocol Identifier* (TPID) field always contains the value 0x8100; it is used to identify 802.1Q-tagged frames.
- The *Tag Control Information* (TCI) field consists of three subfields: *Priority Code Point* (PCP) and *Drop Eligible Indicator* (DEI) are used for *Quality of Service* (QoS). The *VLAN Identifier* (VID) field is used to indicate which VLAN the frame is in. The VID field is 12 bits in length, and for that reason, there are 4,096 ($2^{12}$) VLANs in total.
- To configure a trunk port, use the **switchport mode trunk** command. If the switch supports both 802.1Q and ISL, you must use the **switchport trunk encapsulation dot1q** command first; if the switch only supports 802.1Q, this command is not needed.
- Use the **show interfaces trunk** command to verify trunk ports, including information such as which VLANs are allowed on each trunk port.
- By default, all VLANs are allowed on a trunk port, meaning it can forward and receive frames in all VLANs.
- Use the **switchport trunk allowed vlan** command to specify the VLANs allowed on a trunk. You can specify the list of VLANs or use the keywords **add**, **all**, **except**, **none**, or **remove**.
- The *native VLAN* is the VLAN that is untagged on a trunk port. Untagged frames received on a trunk port are assigned to the native VLAN, and frames in the native VLAN are forwarded untagged. The native VLAN of a trunk port is VLAN 1 by default.
- The native VLAN can be configured with the **switchport trunk native vlan** *vlan-id* command. The command is configured per port, so each port on a switch can have a different native VLAN, but make sure the native VLAN matches on both sides of a trunk connection.
- It is recommended that you configure an unused VLAN (that is not the default of VLAN 1) as the native VLAN, which is equivalent to disabling it.
- *Inter-VLAN routing* is the process of routing between subnets in a LAN that is segmented using VLANs. Inter-VLAN routing can be performed by an external router or by a *multilayer switch* (a switch that has routing capabilities).
- A router can perform inter-VLAN routing by using a separate interface per subnet/VLAN or by *router on a stick* (ROAS), in which a trunk link connects the router and switch.

- ROAS uses virtual subinterfaces. To configure a subinterface, use the **interface** command and add a period and a number to identify the subinterface to the end of the interface name (i.e., **interface g0/0.10**). The subinterface identifier does not have to match the VLAN ID.
- Configure a subinterface's VLAN with the **encapsulation dot1q** *vlan-id* command. Then, configure an IP address in the same manner as on a router.
- If using the native VLAN over the ROAS trunk, use the **encapsulation dot1q** *vlan-id* **native** command on the native VLAN's subinterface. Or, configure the native VLAN's IP address on the physical interface (the **encapsulation dot1q** command is not necessary on the physical interface).
- A multilayer switch can also be called a *Layer 3 switch* (in contrast to a standard *Layer 2 switch*). A multilayer switch uses *switch virtual interfaces* (SVIs) to perform inter-VLAN routing. Each SVI is associated with a VLAN and can be configured with the **interface vlan** *vlan-id* command.
- Use the **ip routing** command on a multilayer switch to allow the switch to route packets.
- A physical port on a multilayer switch can be configured as a *routed port*, which functions like a router interface. Use the **no switchport** command to convert a switch port to a routed port, and then configure an IP address on it.
- To forward packets to external destinations, multilayer switches need routes, just like routers—either static routes or routes learned via a dynamic routing protocol (such as OSPF).