Part I: Ethernet LANs

The ICND1 half of the CCNA R&S exam topics introduces the basics of Ethernet LANs and LAN switching. Part I of this ICND2 Cert Guide builds on that knowledge with six more chapters about LANs and LAN switching.

Part I discusses two major topics in depth, to a configuration, verification, and troubleshooting level on each: Spanning Tree Protocol (STP) and VLAN Trunking Protocol (VTP). Chapters 2 and 3 get fairly deep on STP, a switch feature that requires basic configuration skills, but one that requires a lot of thought to master verification and troubleshooting. Chapter 5 discusses VTP and how it can be used to advertise VLAN configuration around a network of switches, again to a troubleshooting depth.

Besides the major focus on STP and VTP, Chapter 6 introduces a small set of new topics: 802.1x, AAA authentication, DHCP snooping, and switch stacking.

Beyond the four chapters that focus on completely new material (Chapters 2, 3, 5, and 6), Chapters 1 and 4 revisit some topics you will already be comfortable with if you remember most of what you learned for the ICND1 half of the CCNA R&S certification. Chapter 1 discusses VLANs and VLAN trunks, topics duplicated in ICND1 and ICND2 exam topics. So, this book includes the same chapter in both books. For those of you who read the *CCENT/CCNA ICND1 100-105 Official Cert Guide*, specifically Chapter 11 of that book, then use Chapter 1 of this book as a review. Make sure you recall the details, and move quickly through that chapter.

Chapter 1. Implementing Ethernet Virtual LANs

This chapter covers the following exam topics:

1.0 LAN Switching Technologies

1.1 Configure, verify, and troubleshoot VLANs (normal/extended range) spanning multiple switches

1.1.a Access ports (data and voice)

1.1.b Default VLAN

1.2 Configure, verify, and troubleshoot interswitch connectivity

1.2.a Add and remove VLANs on a trunk

1.2.b DTP and VTP (v1&v2)

Virtual LANs (VLAN) have an impact on many parts of a switch’s logic. Frame forwarding happens per VLAN. MAC learning adds MAC table entries, and those entries include the associated VLAN. Even Spanning Tree Protocol (STP), a big focus in Part I of this book, often happens per-VLAN.

This chapter examines how many switch core features work in the context of VLANs. The chapter breaks the topics into concepts in the first section of the chapter, with configuration and verification in the second half. The topics include VLANs, VLAN trunking, routing between VLANs, plus voice and data VLANs. (Chapter 4, “LAN Troubleshooting,” revisits some of these topics from a troubleshooting perspective.)

For you ICND1 Cert Guide readers, note that this chapter is identical to the ICND1 100-105 Cert Guide’s Chapter 11. Both the ICND1 and ICND2 exams include specific exam topics about most of the content in this chapter. By using the exact same chapter for duplicate exam topics between the two books, hopefully those of you who remember a lot about these topics can move quickly through this chapter. For those who do not remember as much, just treat it as a normal chapter.

“Do I Know This Already?” Quiz

Take the quiz (either here, or use the PCPT software) if you want to use the score to help you decide how much time to spend on this chapter. The answers are at the bottom of the page following the quiz, and the explanations are in DVD Appendix C and in the PCPT software.

Foundation Topics Section	Questions
Virtual LAN Concepts	1–3
VLAN and VLAN Trunking Configuration and Verification	4–6

Table 1-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

1. In a LAN, which of the following terms best equates to the term *VLAN*?

a. Collision domain

b. Broadcast domain

c. Subnet

d. Single switch

e. Trunk

2. Imagine a switch with three configured VLANs. How many IP subnets are required, assuming that all hosts in all VLANs want to use TCP/IP?

a. 0

b. 1

c. 2

d. 3

e. You cannot tell from the information provided.

3. Switch SW1 sends a frame to switch SW2 using 802.1Q trunking. Which of the answers describes how SW1 changes or adds to the Ethernet frame before forwarding the frame to SW2?

a. Inserts a 4-byte header and does change the MAC addresses

b. Inserts a 4-byte header and does not change the MAC addresses

c. Encapsulates the original frame behind an entirely new Ethernet header

d. None of the other answers are correct.

4. Imagine that you are told that switch 1 is configured with the **dynamic auto** parameter for trunking on its Fa0/5 interface, which is connected to switch 2. You have to configure switch 2. Which of the following settings for trunking could allow trunking to work? (Choose two answers.)

a. on

b. dynamic auto

c. dynamic desirable

d. access

5. A switch has just arrived from Cisco. The switch has never been configured with any VLANs, but VTP has been disabled. An engineer gets into configuration mode and issues the **vlan 22** command, followed by the **name Hannahs-VLAN** command. Which of the following are true? (Choose two answers.)

a. VLAN 22 is listed in the output of the **show vlan brief** command.

b. VLAN 22 is listed in the output of the **show running-config** command.

c. VLAN 22 is not created by this process.

d. VLAN 22 does not exist in that switch until at least one interface is assigned to that VLAN.

6. Which of the following commands identify switch interfaces as being trunking interfaces: interfaces that currently operate as VLAN trunks? (Choose two answers.)

a. show interfaces

b. show interfaces switchport

c. show interfaces trunk

d. show trunks

Answers to the “Do I Know This Already?” quiz:

1 B **2** D **3** B **4** A, C **5** A, B **6** B, C

Foundation Topics

Virtual LAN Concepts

Before understanding VLANs, you must first have a specific understanding of the definition of a LAN. For example, from one perspective, a LAN includes all the user devices, servers, switches, routers, cables, and wireless access points in one location. However, an alternative narrower definition of a LAN can help in understanding the concept of a virtual LAN:

> A LAN includes all devices in the same broadcast domain.

A broadcast domain includes the set of all LAN-connected devices, so that when any of the devices sends a broadcast frame, all the other devices get a copy of the frame. So, from one perspective, you can think of a LAN and a broadcast domain as being basically the same thing.

Without VLANs, a switch considers all its interfaces to be in the same broadcast domain. That is, for one switch, when a broadcast frame entered one switch port, the switch forwarded that broadcast frame out all other ports. With that logic, to create two different LAN broadcast domains, you had to buy two different Ethernet LAN switches, as shown in Figure 1-1.

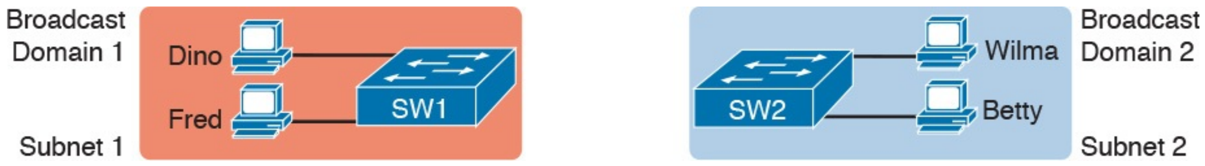


Figure 1-1 *Creating Two Broadcast Domains with Two Physical Switches and No VLANs*

With support for VLANs, a single switch can accomplish the same goals of the design in Figure 1-1—to create two broadcast domains—with a single switch. With VLANs, a switch can configure some interfaces into one broadcast domain and some into another, creating multiple broadcast domains. These individual broadcast domains created by the switch are called virtual LANs (VLAN).

For example, in Figure 1-2, the single switch creates two VLANs, treating the ports in each VLAN as being completely separate. The switch would never forward a frame sent by Dino (in VLAN 1) over to either Wilma or Betty (in VLAN 2).

Key Topic

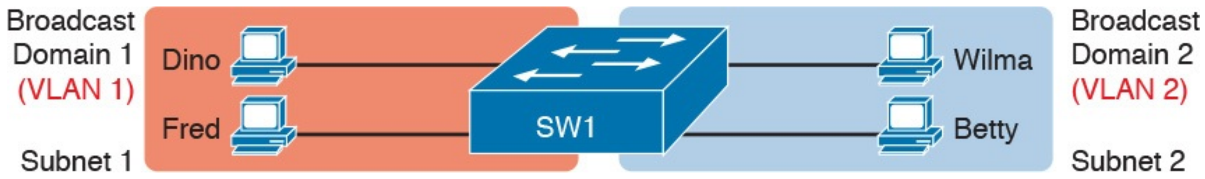


Figure 1-2 *Creating Two Broadcast Domains Using One Switch and VLANs*

Designing campus LANs to use more VLANs, each with a smaller number of devices, often helps improve the LAN in many ways. For example, a broadcast sent by one host in a VLAN will be received and processed by all the other hosts in the VLAN—but not by hosts in a different VLAN. Limiting the number of hosts that receive a single broadcast frame reduces the number of hosts that waste effort processing unneeded broadcasts. It also reduces security risks, because fewer hosts see frames sent by any one host. These are just a few reasons for separating hosts into different VLANs. The following list summarizes the most common reasons for choosing to create smaller broadcast domains (VLANs):

Key Topic

- To reduce CPU overhead on each device by reducing the number of devices that receive each broadcast frame
- To reduce security risks by reducing the number of hosts that receive

copies of frames that the switches flood (broadcasts, multicasts, and unknown unicasts)

- To improve security for hosts that send sensitive data by keeping those hosts on a separate VLAN
- To create more flexible designs that group users by department, or by groups that work together, instead of by physical location
- To solve problems more quickly, because the failure domain for many problems is the same set of devices as those in the same broadcast domain
- To reduce the workload for the Spanning Tree Protocol (STP) by limiting a VLAN to a single access switch

This chapter does not examine all the reasons for VLANs in more depth. However, know that most enterprise networks use VLANs quite a bit. The rest of this chapter looks closely at the mechanics of how VLANs work across multiple Cisco switches, including the required configuration. To that end, the next section examines VLAN trunking, a feature required when installing a VLAN that exists on more than one LAN switch.

Creating Multiswitch VLANs Using Trunking

Configuring VLANs on a single switch requires only a little effort: You simply configure each port to tell it the VLAN number to which the port belongs. With multiple switches, you have to consider additional concepts about how to forward traffic between the switches.

When using VLANs in networks that have multiple interconnected switches, the switches need to use *VLAN trunking* on the links between the switches. VLAN trunking causes the switches to use a process called *VLAN tagging*, by which the sending switch adds another header to the frame before sending it over the trunk. This extra trunking header includes a *VLAN identifier* (VLAN ID) field so that the sending switch can associate the frame with a particular VLAN ID, and the receiving switch can then know in what VLAN each frame belongs.

Figure 1-3 shows an example that demonstrates VLANs that exist on multiple switches, but it does not use trunking. First, the design uses two VLANs: VLAN 10 and VLAN 20. Each switch has two ports assigned to each VLAN, so each VLAN exists in both switches. To forward traffic in VLAN 10 between the two switches, the design includes a link between switches, with that link fully inside VLAN 10. Likewise, to support VLAN 20 traffic between switches, the design uses a second link between switches, with that link inside VLAN 20.

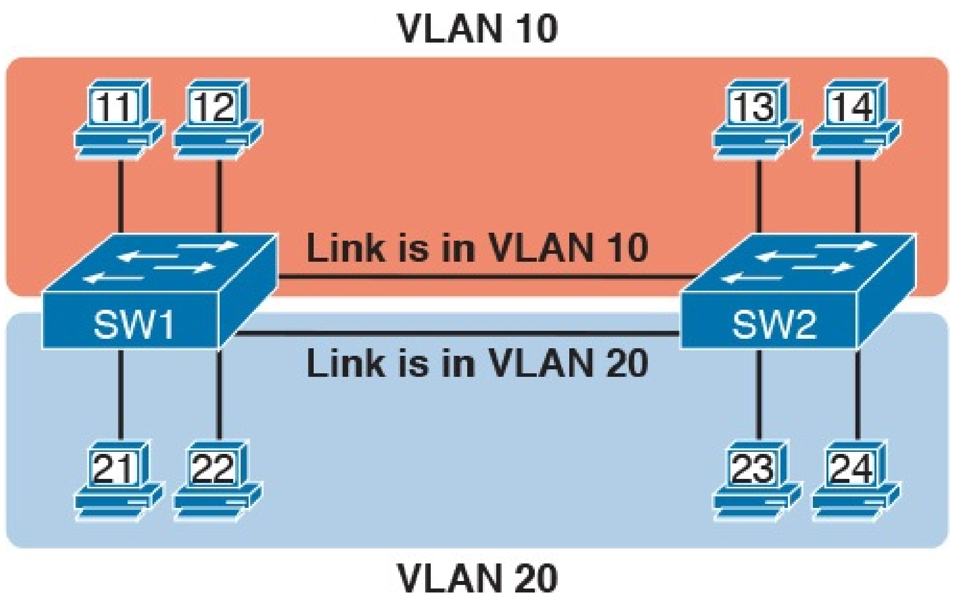


Figure 1-3 *Multiswitch VLAN Without VLAN Trunking*

The design in Figure 1-3 functions perfectly. For example, PC11 (in VLAN 10) can send a frame to PC14. The frame flows into SW1, over the top link (the one that is in VLAN 10) and over to SW2.

The design shown in Figure 1-3 works, but it simply does not scale very well. It requires one physical link between switches to support every VLAN. If a design needed 10 or 20 VLANs, you would need 10 or 20 links between switches, and you would use 10 or 20 switch ports (on each switch) for those links.

VLAN Tagging Concepts

VLAN trunking creates one link between switches that supports as many VLANs as you need. As a VLAN trunk, the switches treat the link as if it were a part of all the VLANs. At the same time, the trunk keeps the VLAN traffic separate, so frames in VLAN 10 would not go to devices in VLAN 20, and vice versa, because each frame is identified by VLAN number as it crosses the trunk. Figure 1-4 shows the idea, with a single physical link between the two switches.

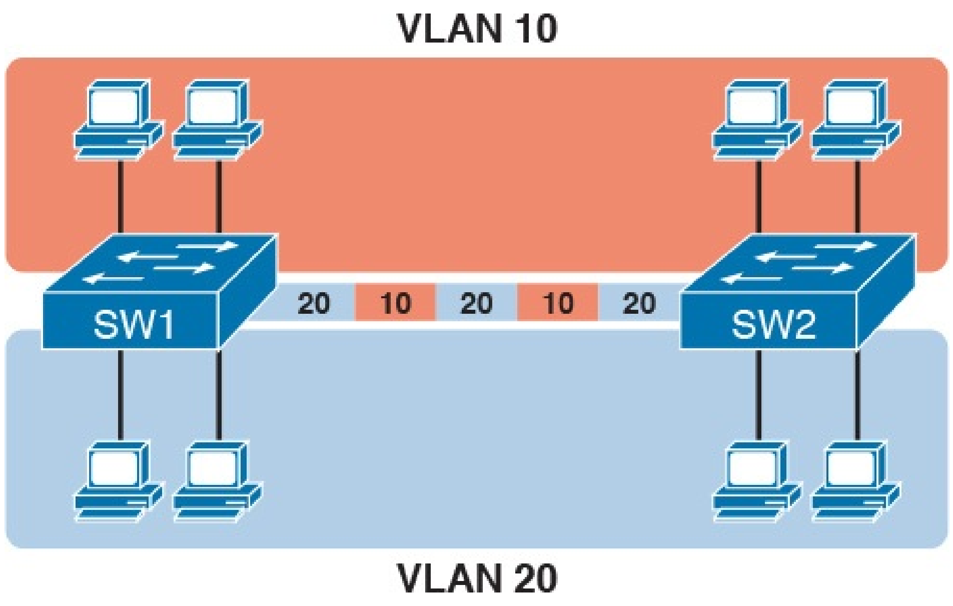


Figure 1-4 *Multiswitch VLAN with Trunking*

The use of trunking allows switches to pass frames from multiple VLANs over a single physical connection by adding a small header to the Ethernet frame. For example, Figure 1-5 shows PC11 sending a broadcast frame on interface Fa0/1 at Step 1. To flood the frame, switch SW1 needs to forward the broadcast frame to switch SW2. However, SW1 needs to let SW2 know that the frame is part of VLAN 10, so that after the frame is received, SW2 will flood the frame only into VLAN 10, and not into VLAN 20. So, as shown at Step 2, before sending the frame, SW1 adds a VLAN header to the original Ethernet frame, with the VLAN header listing a VLAN ID of 10 in this case.

Key Topic

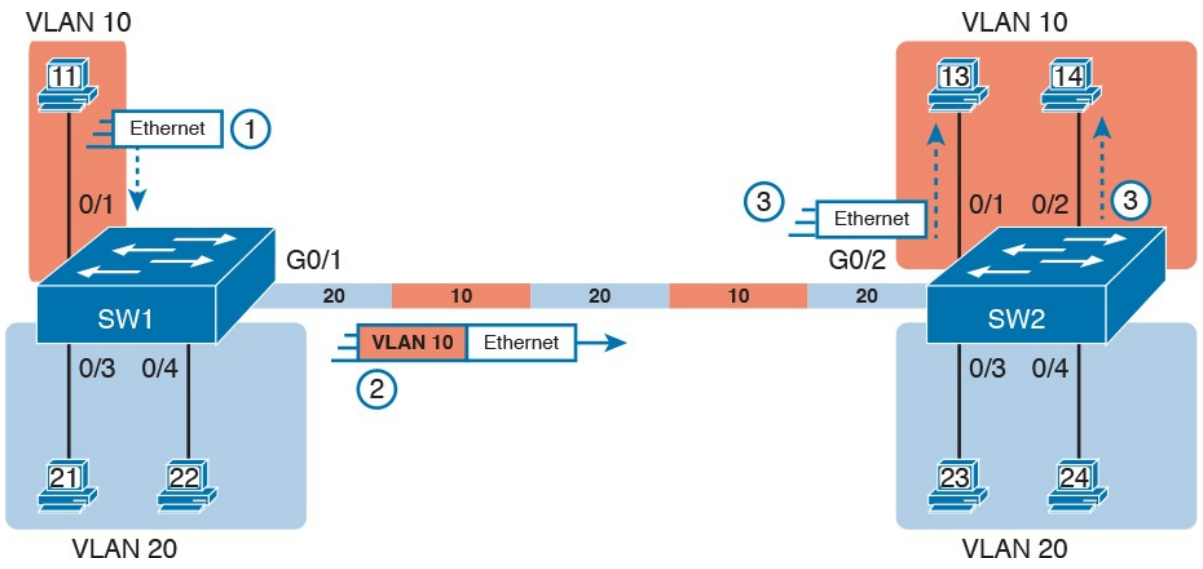


Figure 1-5 *VLAN Trunking Between Two Switches*

When SW2 receives the frame, it understands that the frame is in VLAN 10. SW2 then removes the VLAN header, forwarding the original frame out its interfaces in VLAN 10 (Step 3).

For another example, consider the case when PC21 (in VLAN 20) sends a broadcast. SW1 sends the broadcast out port Fa0/4 (because that port is in VLAN 20) and out Gi0/1 (because it is a trunk, meaning that it supports multiple different VLANs). SW1 adds a trunking header to the frame, listing a VLAN ID of 20. SW2 strips off the trunking header after determining that the frame is part of VLAN 20, so SW2 knows to forward the frame out only ports Fa0/3 and Fa0/4, because they are in VLAN 20, and not out ports Fa0/1 and Fa0/2, because they are in VLAN 10.

The 802.1Q and ISL VLAN Trunking Protocols

Cisco has supported two different trunking protocols over the years: Inter-Switch Link (ISL) and IEEE 802.1Q. Cisco created ISL long before 802.1Q, in part because the IEEE had not yet defined a VLAN trunking standard. Years later, the IEEE completed work on the 802.1Q standard, which defines a different way to do trunking. Today, 802.1Q has become the more popular trunking protocol, with Cisco not even supporting ISL in some of its newer models of LAN switches, including the 2960 switches used in the examples in this book.

While both ISL and 802.1Q tag each frame with the VLAN ID, the details differ. 802.1Q inserts an extra 4-byte 802.1Q VLAN header into the original frame’s Ethernet header, as shown at the top of Figure 1-6. As for the fields in the 802.1Q header, only the 12-bit VLAN ID field inside the 802.1Q header matters for topics discussed in this book. This 12-bit field supports a theoretical maximum of 2^{12} (4096) VLANs, but in practice it supports a maximum of 4094. (Both 802.1Q and ISL use 12 bits to tag the VLAN ID, with two reserved values [0 and 4095].)

Key Topic

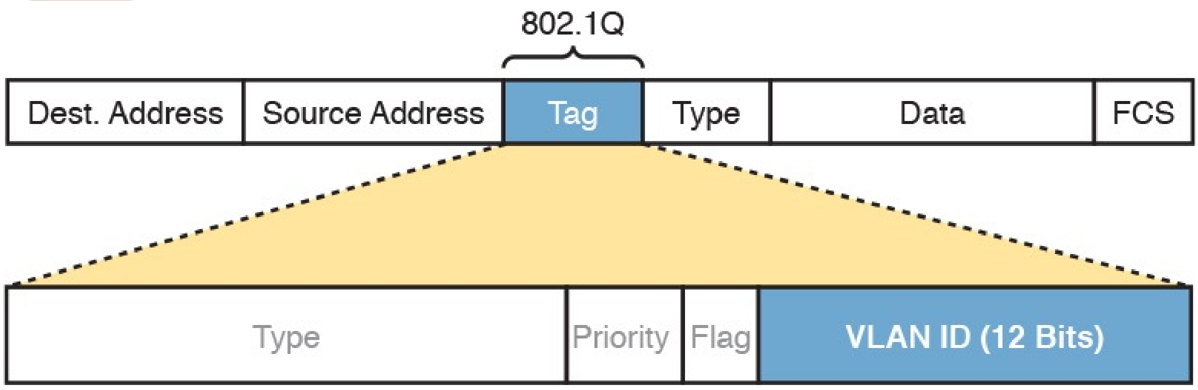


Figure 1-6 *802.1Q Trunking*

Cisco switches break the range of VLAN IDs (1–4094) into two ranges: the normal range and the extended range. All switches can use normal-range VLANs with values from 1 to 1005. Only some switches can use extended-range VLANs with VLAN IDs from 1006 to 4094. The rules for which switches can use extended-range VLANs depend on the configuration of the VLAN Trunking Protocol (VTP), which is discussed briefly in the section "VLAN Trunking Configuration," later in this chapter.

802.1Q also defines one special VLAN ID on each trunk as the *native VLAN* (defaulting to use VLAN 1). By definition, 802.1Q simply does not add an 802.1Q header to frames in the native VLAN. When the switch on the other side of the trunk receives a frame that does not have an 802.1Q header, the receiving switch knows that the frame is part of the native VLAN. Note that because of this behavior, both switches must agree on which VLAN is the native VLAN.

The 802.1Q native VLAN provides some interesting functions, mainly to support connections to devices that do not understand trunking. For example, a Cisco switch could be cabled to a switch that does not understand 802.1Q trunking. The Cisco switch could send frames in the native VLAN—meaning that the frame has no trunking header—so that the other switch would understand the frame. The native VLAN concept gives switches the capability of at least passing traffic in one VLAN (the native VLAN), which can allow some basic functions, like reachability to telnet into a switch.

Forwarding Data Between VLANs

If you create a campus LAN that contains many VLANs, you typically still need all devices to be able to send data to all other devices. This next topic discusses some concepts about how to route data between those VLANs.

First, it helps to know a few terms about some categories of LAN switches. All the Ethernet switch functions described in the ICND1 Cert Guide use the details and logic defined by OSI Layer 2 protocols. For example, many chapters of the ICND1 Cert Guide discussed how LAN switches receive Ethernet frames (a Layer 2 concept), look at the destination Ethernet MAC address (a Layer 2 address), and forward the Ethernet frame out some other interface. This chapter has already discussed the concept of VLANs as broadcast domains, which is yet another Layer 2 concept.

While some LAN switches work just as described in the ICND1 Cert Guide, some LAN switches have even more functions. LAN switches that forward data based on Layer 2 logic often go by the name *Layer 2 switch*. However, some other switches can do some functions like a router, using additional logic defined by Layer 3 protocols. These switches go by the name *multilayer switch*, or *Layer 3 switch*. This section first discusses how to forward data between VLANs when using Layer 2 switches and ends with a brief discussion of how to use Layer 3 switches.

Routing Packets Between VLANs with a Router

When including VLANs in a campus LAN design, the devices in a VLAN need to be in the same subnet. Following the same design logic, devices in different VLANs need to be in different subnets. For example, in Figure 1-7, the two PCs on the left sit in VLAN 10, in subnet 10. The two PCs on the right sit in a different VLAN (20), with a different subnet (20).

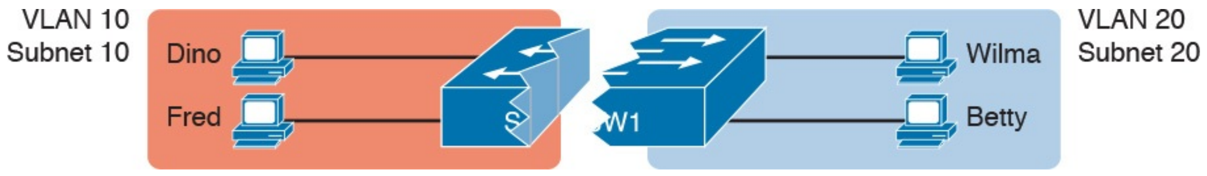


Figure 1-7 *Layer 2 Switch Does Not Route Between the VLANs*

Note

The figure refers to subnets somewhat generally, like "subnet 10," just so the subnet numbers do not distract. Also, note that the subnet numbers do not have to be the same number as the VLAN numbers.

Figure 1-7 shows the switch as if it were two switches broken in two to emphasize the point that Layer 2 switches will not forward data between two VLANs. When configured with some ports in VLAN 10 and others in VLAN 20, the switch acts like two separate switches in which it will forward traffic. In fact, one goal of VLANs is to separate traffic in one VLAN from another, preventing frames in one VLAN from leaking over to other VLANs. For example, when Dino (in VLAN 10) sends any Ethernet frame, if SW1 is a Layer 2 switch, that switch will not forward the frame to the PCs on the right in VLAN 20.

The network as a whole needs to support traffic flowing into and out of each VLAN, even though the Layer 2 switch does not forward frames outside a VLAN. The job of forwarding data into and out of a VLAN falls to routers. Instead of switching Layer 2 Ethernet frames between the two VLANs, the network must route Layer 3 packets between the two subnets.

That previous paragraph has some very specific wording related to Layers 2 and 3, so take a moment to reread and reconsider it for a moment. The Layer 2 logic does not let the Layer 2 switch forward the Layer 2 protocol data unit (L2PDU), the Ethernet frame, between VLANs. However, routers can route Layer 3 PDUs (L3PDU), packets, between subnets as their normal job in life.

For example, Figure 1-8 shows a router that can route packets between subnets 10 and 20. The figure shows the same Layer 2 switch as shown in Figure 1-7, with the same perspective of the switch being split into parts with two different VLANs, and with the same PCs in the same VLANs and subnets. Now Router R1 has one LAN physical interface connected to the switch and assigned to VLAN 10, and a second physical interface connected to the switch and assigned to VLAN 20. With an interface connected to each subnet, the Layer 2 switch can keep doing its job—forwarding frames inside a VLAN, while the router can do its job—routing IP packets between the subnets.

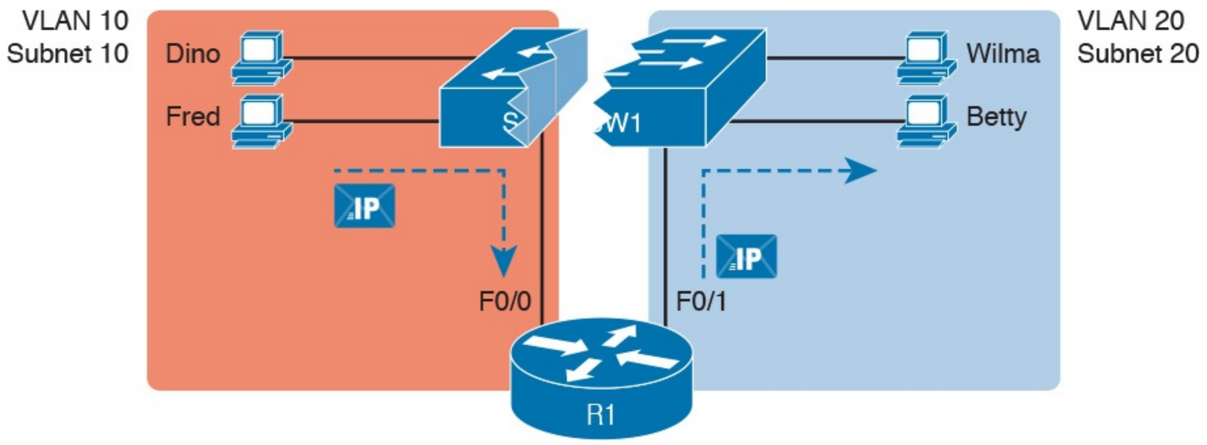


Figure 1-8 *Routing Between Two VLANs on Two Physical Interfaces*

The figure shows an IP packet being routed from Fred, which sits in one VLAN/subnet, to Betty, which sits in the other. The Layer 2 switch forwards two different Layer 2 Ethernet frames: one in VLAN 10, from Fred to R1's F0/0 interface, and the other in VLAN 20, from R1's F0/1 interface to Betty. From a Layer 3 perspective, Fred sends the IP packet to its default router (R1), and R1 routes the packet out another interface (F0/1) into another subnet where Betty resides.

While the design shown in Figure 1-8 works, it uses too many physical interfaces, one per VLAN. A much less expensive (and much preferred) option uses a VLAN trunk between the switch and router, requiring only one physical link between the router and switch, while supporting all VLANs. Trunking can work between any two devices that choose to support it: between two switches, between a router and a switch, or even between server hardware and a switch.

Figure 1-9 shows the same design idea as Figure 1-8, with the same packet being sent from Fred to Betty, except now R1 uses VLAN trunking instead of a separate link for each VLAN.

Key Topic

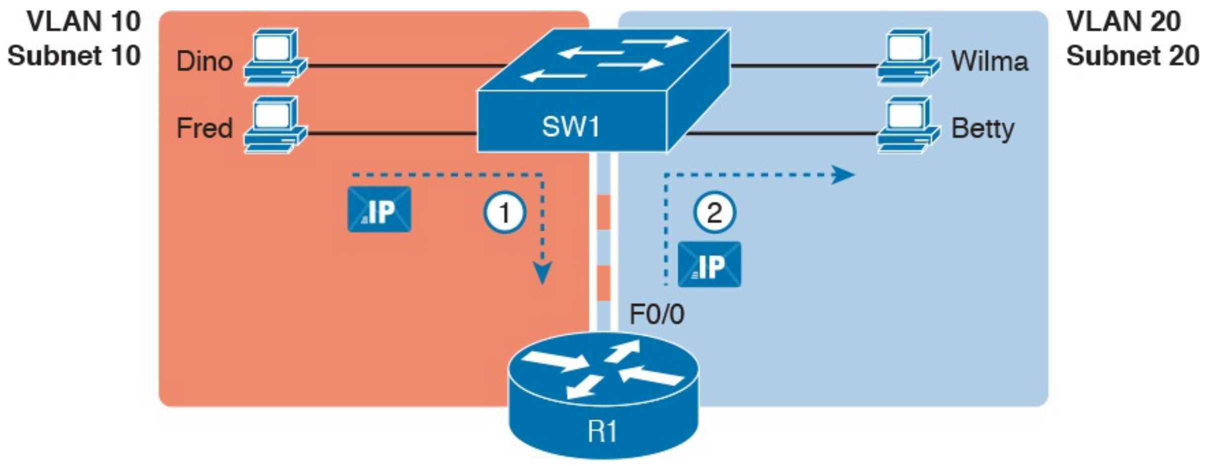


Figure 1-9 *Routing Between Two VLANs Using a Trunk on the Router*

Note

Because the router has a single physical link connected to the LAN switch, this design is sometimes called a *router-on-a-stick*.

As a brief aside about terminology, many people describe the concept in Figures 1-8 and 1-9 as "routing packets between VLANs." You can use that phrase, and people know what you mean. However, note that this phrase is not literally true, because it refers to routing packets (a Layer 3 concept) and VLANs (a Layer 2 concept). It just takes fewer words to say something like "routing between VLANs" rather than the literally true but long "routing Layer 3 packets between Layer 3 subnets, with those subnets each mapping to a Layer 2 VLAN."

Routing Packets with a Layer 3 Switch

Routing packets using a physical router, even with the VLAN trunk in the router-on-a-stick model shown in Figure 1-9, still has one significant problem: performance. The physical link puts an upper limit on how many bits can be routed, and less expensive routers tend to be less powerful, and might not be able to route a large enough number of packets per second (pps) to keep up with the traffic volumes.

The ultimate solution moves the routing functions inside the LAN switch hardware. Vendors long ago started combining the hardware and software features of their Layer 2 LAN switches, plus their Layer 3 routers, creating products called *Layer 3 switches* (also known as *multilayer switches*). Layer 3 switches can be configured to act only as a Layer 2 switch, or they can be configured to do both Layer 2 switching as well as Layer 3 routing.

Today, many medium- to large-sized enterprise campus LANs use Layer 3 switches to route packets between subnets (VLANs) in a campus.

In concept, a Layer 3 switch works a lot like the original two devices on which the Layer 3 switch is based: a Layer 2 LAN switch and a Layer 3 router. In fact, if you take the concepts and packet flow shown in Figure 1-8, with a separate Layer 2 switch and Layer 3 router, and then imagine all those features happening inside one device, you have the general idea of what a Layer 3 switch does. Figure 1-10 shows that exact concept, repeating many details of Figure 1-8, but with an overlay that shows the one Layer 3 switch doing the Layer 2 switch functions and the separate Layer 3 routing function.

Key Topic

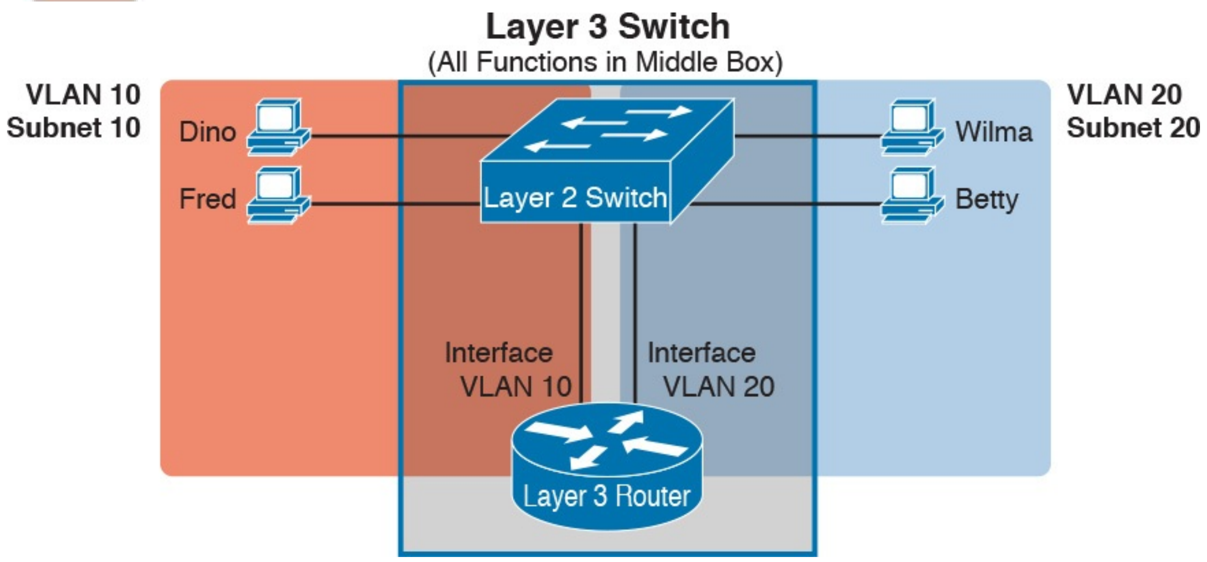


Figure 1-10 *Multilayer Switch: Layer 2 Switching with Layer 3 Routing in One Device*

This chapter introduces the core concepts of routing IP packets between VLANs (or more accurately, between the subnets on the VLANs). Chapter 19, “IPv4 Routing in the LAN,” shows how to configure designs that use an external router with router-on-a-stick. This chapter now turns its attention to configuration and verification tasks for VLANs and VLAN trunks.

VLAN and VLAN Trunking Configuration and Verification

Cisco switches do not require any configuration to work. You can purchase Cisco switches, install devices with the correct cabling, turn on the switches, and they work. You would never need to configure the switch, and it would work fine, even if you interconnected switches, until you needed more than one VLAN. But if you want to use VLANs—and most enterprise networks do—you need to add some configuration.

This chapter separates the VLAN configuration details into two major sections. The first section looks at how to configure access interfaces, which are switch interfaces that do not use VLAN trunking. The second part shows how to configure interfaces that do use VLAN trunking.

Creating VLANs and Assigning Access VLANs to an Interface

This section shows how to create a VLAN, give the VLAN a name, and assign interfaces to a VLAN. To focus on these basic details, this section shows examples using a single switch, so VLAN trunking is not needed.

For a Cisco switch to forward frames in a particular VLAN, the switch must be configured to believe that the VLAN exists. In addition, the switch must have nontrunking interfaces (called *access interfaces*) assigned to the VLAN, and/or trunks that support the VLAN. The configuration steps for access interfaces are as follows, with the trunk configuration shown later in the section "VLAN Trunking Configuration":

Config Checklist

Step 1. To configure a new VLAN, follow these steps:

- **A.** From configuration mode, use the **vlan** *vlan-id* command in global configuration mode to create the VLAN and to move the user into VLAN configuration mode.
- **B.** (Optional) Use the **name** *name* command in VLAN configuration mode to list a name for the VLAN. If not configured, the VLAN name is VLANZZZZ, where *ZZZZ* is the four-digit decimal VLAN ID.

Step 2. For each access interface (each interface that does not trunk, but instead belongs to a single VLAN), follow these steps:

- **A.** Use the **interface** *type number* command in global configuration mode to move into interface configuration mode for each desired interface.
- **B.** Use the **switchport access vlan** *id-number* command in interface configuration mode to specify the VLAN number associated with that interface.
- **C.** (Optional) Use the **switchport mode access** command in interface configuration mode to make this port always operate in access mode (that is, to not trunk).

While the list might look a little daunting, the process on a single switch is actually pretty simple. For example, if you want to put the switch's ports in three VLANs—11, 12, and 13—you just add three **vlan** commands: **vlan 11**, **vlan 12**, and **vlan 13**. Then, for each interface, add a **switchport access vlan 11** (or **12** or **13**) command to assign that interface to the proper VLAN.

Note

The term *default VLAN* (as shown in the exam topics) refers to the default setting on the **switchport access vlan** *vlan-id* command, and that default is VLAN ID 1. In other words, by default, each port is assigned to access VLAN 1.

VLAN Configuration Example 1: Full VLAN Configuration

Example 1-1 shows the configuration process of adding a new VLAN and assigning access interfaces to that VLAN. Figure 1-11 shows the network used in the example, with one LAN switch (SW1) and two hosts in each of three VLANs (1, 2, and 3). The example shows the details of the two-step process for VLAN 2 and the interfaces in VLAN 2, with the configuration of VLAN 3 deferred until the next example.

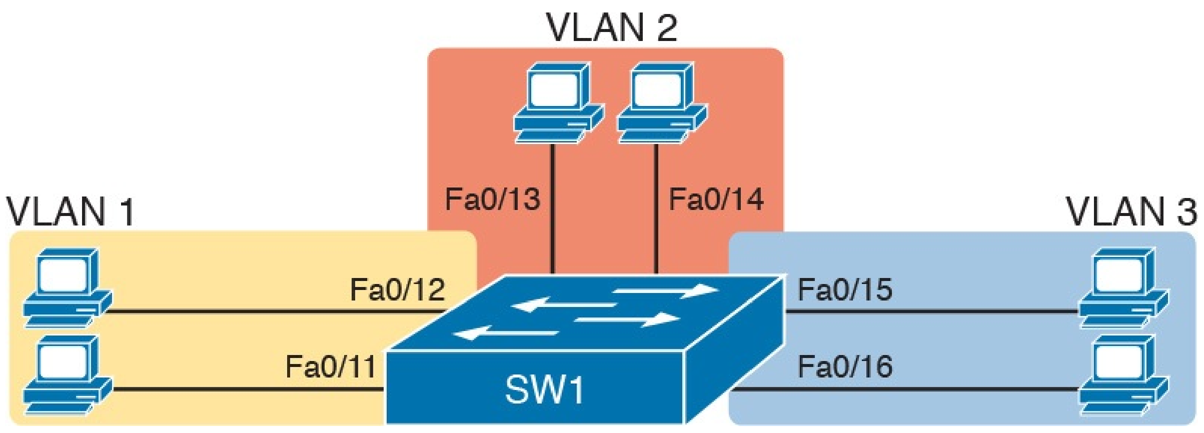


Figure 1-11 *Network with One Switch and Three VLANs*

Example 1-1 *Configuring VLANs and Assigning VLANs to Interfaces*

Click here to view code image

```
SW1# show vlan brief
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
------------------------
1    default                          active    Fa0/1,
Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5,
Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9,
Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13,
Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17,
Fa0/18, Fa0/19, Fa0/20
```

```
                                                Fa0/21,
Fa0/22, Fa0/23, Fa0/24
                                                Gi0/1,
Gi0/2
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
! Above, VLANs 2 and 3 do not yet exist. Below, VLAN 2
is added, with name Freds-vlan,
! with two interfaces assigned to VLAN 2.

SW1# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW1(config)# vlan 2
SW1(config-vlan)# name Freds-vlan
SW1(config-vlan)# exit
SW1(config)# interface range fastethernet 0/13 - 14
SW1(config-if)# switchport access vlan 2
SW1(config-if)# switchport mode access
SW1(config-if)# end

! Below, the show running-config command lists the
interface subcommands on
! interfaces Fa0/13 and Fa0/14.
SW1# show running-config
! Many lines omitted for brevity
! Early in the output:
vlan 2
 name Freds-vlan
!
! more lines omitted for brevity
interface FastEthernet0/13
 switchport access vlan 2
 switchport mode access
!
interface FastEthernet0/14
 switchport access vlan 2
 switchport mode access
!
SW1# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
------------------------
1    default                          active    Fa0/1,
Fa0/2, Fa0/3, Fa0/4
```

```
                                                Fa0/5,
Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9,
Fa0/10, Fa0/11, Fa0/12
                                                Fa0/15,
Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19,
Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23,
Fa0/24, Gi0/1, Gi0/2
2    Freds-vlan                       active    Fa0/13,
Fa0/14
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

SW1# show vlan id 2
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
------------------------
2   Freds-vlan                        active    Fa0/13,
Fa0/14

VLAN Type  SAID       MTU   Parent RingNo BridgeNo
Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ----
-------- ------ ------
2    enet  100010     1500  -      -      -        -

Remote SPAN VLAN
----------------
Disabled

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

The example begins with the **show vlan brief** command, confirming the default settings of five nondeletable VLANs, with all interfaces assigned to VLAN 1. (VLAN 1 cannot be deleted, but can be used. VLANs 1002–1005 cannot be deleted and cannot be used as access VLANs today.) In particular, note that this 2960 switch has 24 Fast Ethernet ports (Fa0/1–Fa0/24) and two Gigabit Ethernet ports (Gi0/1 and Gi0/2), all of which are listed as being in VLAN 1 per that first command's output.

Next, the example shows the process of creating VLAN 2 and assigning interfaces Fa0/13 and Fa0/14 to VLAN 2. Note in particular that the example uses the **interface range** command, which causes the **switchport access vlan 2** interface subcommand to be applied to both interfaces in the range, as confirmed in the **show running-config** command output at the end of the example.

After the configuration has been added, to list the new VLAN, the example repeats the **show vlan brief** command. Note that this command lists VLAN 2, name Freds-vlan, and the interfaces assigned to that VLAN (Fa0/13 and Fa0/14). The **show vlan id 2** command that follows then confirms that ports Fa0/13 and Fa0/14 are assigned to VLAN 2.

The example surrounding Figure 1-11 uses six switch ports, all of which need to operate as access ports. That is, each port should not use trunking, but instead should be assigned to a single VLAN, as assigned by the **switchport access vlan** *vlan-id* command. However, as configured in Example 1-1, these interfaces could negotiate to later become trunk ports, because the switch defaults to allow the port to negotiate trunking and decide whether to act as an access interface or as a trunk interface.

For ports that should always act as access ports, add the optional interface subcommand **switchport mode access**. This command tells the switch to only allow the interface to be an access interface. The upcoming section "VLAN Trunking Configuration" discusses more details about the commands that allow a port to negotiate whether it should use trunking.

Note

The book includes a video that works through a different VLAN configuration example as well. You can find the video on the DVD and on the companion website.

VLAN Configuration Example 2: Shorter VLAN Configuration

Example 1-1 shows several of the optional configuration commands, with a side effect of being a bit longer than is required. Example 1-2 shows a much briefer alternative configuration, picking up the story where Example 1-1 ended and showing the addition of VLAN 3 (as shown in Figure 1-11). Note that SW1 does not know about VLAN 3 at the beginning of this example.

Example 1-2 *Shorter VLAN Configuration Example (VLAN 3)*

Click here to view code image

```
SW1# configure terminal
```

```
Enter configuration commands, one per line.  End with
CNTL/Z.
SW1(config)# interface range Fastethernet 0/15 - 16
SW1(config-if-range)# switchport access vlan 3
% Access VLAN does not exist. Creating vlan 3
SW1(config-if-range)# ^Z

SW1# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
------------------------
1    default                          active    Fa0/1,
Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5,
Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9,
Fa0/10, Fa0/11, Fa0/12
                                                Fa0/17,
Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21,
Fa0/22, Fa0/23, Fa0/24
                                                Gi0/1,
Gi0/2
2    Freds-vlan                       active    Fa0/13,
Fa0/14
3    VLAN0003                         active    Fa0/15,
Fa0/16
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

Example 1-2 shows how a switch can dynamically create a VLAN—the equivalent of the **vlan** *vlan-id* global config command—when the **switchport access vlan** interface subcommand refers to a currently unconfigured VLAN. This example begins with SW1 not knowing about VLAN 3. When the **switchport access vlan 3** interface subcommand was used, the switch realized that VLAN 3 did not exist, and as noted in the shaded message in the example, the switch created VLAN 3, using a default name (VLAN0003). No other steps are required to create the VLAN. At the end of the process, VLAN 3 exists in the switch, and interfaces Fa0/15 and Fa0/16 are in VLAN 3, as noted in the shaded part of the **show vlan brief** command output.

VLAN Trunking Protocol

Before showing more configuration examples, you also need to know something about a Cisco protocol and tool called the VLAN Trunking Protocol (VTP). VTP is a Cisco proprietary tool on Cisco switches that advertises each VLAN configured in one switch (with the **vlan** *number* command) so that all the other switches in the campus learn about that VLAN. However, for various reasons, many enterprises choose not to use VTP.

Each switch can use one of three VTP modes: server, client, or transparent. Switches use either VTP server or client mode when the switch wants to use VTP for its intended purpose of dynamically advertising VLAN configuration information. However, with many Cisco switches and IOS versions, VTP cannot be completely disabled on a Cisco switch; instead, the switch disables VTP by using VTP transparent mode.

Chapter 5, "VLAN Trunking Protocol," discusses how to make use of VTP. Chapters 1 through 4 mostly ignore VTP. To that end, all examples in this book use switches that have been set either to use VTP transparent mode (with the **vtp mode transparent** global command) or to disable it (with the **vtp mode off** global command). Both options allow the administrator to configure both standard- and extended-range VLANs, and the switch lists the **vlan** commands in the running-config file.

Finally, on a practical note, if you happen to do lab exercises with real switches or with simulators, and you see unusual results with VLANs, check the VTP status with the **show vtp status** command. If your switch uses VTP server or client mode, you will find:

- The server switches can configure VLANs in the standard range only (1–1005).
- The client switches cannot configure VLANs.
- Both servers and clients may be learning new VLANs from other switches, and seeing their VLANs deleted by other switches, because of VTP.
- The **show running-config** command does not list any **vlan** commands.

If possible in lab, switch to VTP transparent mode and ignore VTP for your switch configuration practice until you are ready to focus on how VTP works when studying for the ICND2 exam topics.

Note

Do not change VTP settings on any switch that also connects to the production network until you know how VTP works as explained in Chapter 5.

VLAN Trunking Configuration

Trunking configuration between two Cisco switches can be very simple if you just statically configure trunking. For example, if two Cisco 2960 switches connect to each other, they support only 802.1Q and not ISL. You could literally add one interface subcommand for the switch interface on each side of the link (**switchport mode trunk**), and you would create a VLAN trunk that supported all the VLANs known to each switch.

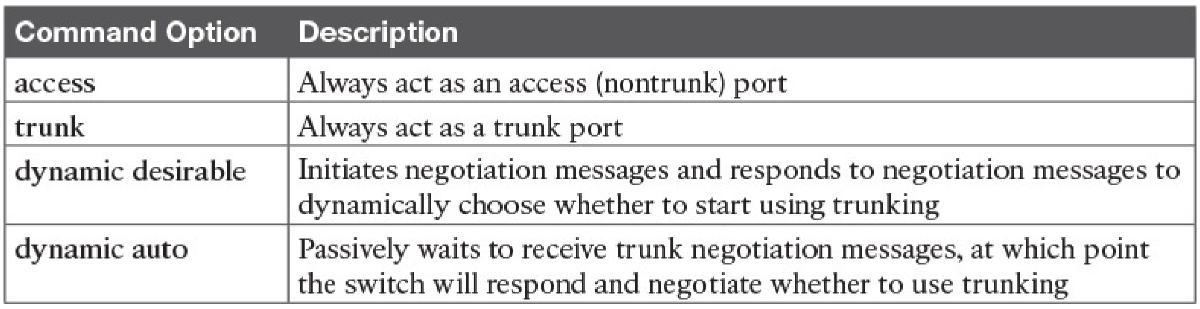
However, trunking configuration on Cisco switches includes many more options, including several options for dynamically negotiating various trunking settings. The configuration can either predefine different settings or tell the switch to negotiate the settings, as follows:

- **The type of trunking:** IEEE 802.1Q, ISL, or negotiate which one to use
- **The administrative mode:** Whether to always trunk, always not trunk, or negotiate

First, consider the type of trunking. Cisco switches that support ISL and 802.1Q can negotiate which type to use, using the Dynamic Trunking Protocol (DTP). If both switches support both protocols, they use ISL; otherwise, they use the protocol that both support. Today, many Cisco switches do not support the older ISL trunking protocol. Switches that support both types of trunking use the **switchport trunk encapsulation** {**dot1q** | **isl** | **negotiate**} interface subcommand to either configure the type or allow DTP to negotiate the type.

DTP can also negotiate whether the two devices on the link agree to trunk at all, as guided by the local switch port's administrative mode. The administrative mode refers to the configuration setting for whether trunking should be used. Each interface also has an *operational* mode, which refers to what is currently happening on the interface, and might have been chosen by DTP's negotiation with the other device. Cisco switches use the **switchport mode** interface subcommand to define the administrative trunking mode, as listed in Table 1-2.

Key Topic



Command Option	Description
access	Always act as an access (nontrunk) port
trunk	Always act as a trunk port
dynamic desirable	Initiates negotiation messages and responds to negotiation messages to dynamically choose whether to start using trunking
dynamic auto	Passively waits to receive trunk negotiation messages, at which point the switch will respond and negotiate whether to use trunking

Table 1-2 Trunking Administrative Mode Options with the **switchport mode** Command

For example, consider the two switches shown in Figure 1-12. This figure shows an expansion of the network shown in Figure 1-11, with a trunk to a new switch (SW2) and with parts of VLANs 1 and 3 on ports attached to SW2. The two switches use a Gigabit Ethernet link for the trunk. In this case, the trunk does not dynamically form by default, because both (2960) switches default to an administrative mode of *dynamic auto*, meaning that neither switch initiates the trunk negotiation process. By changing one switch to use *dynamic desirable* mode, which does initiate the negotiation, the switches negotiate to use trunking, specifically 802.1Q because the 2960s support only 802.1Q.

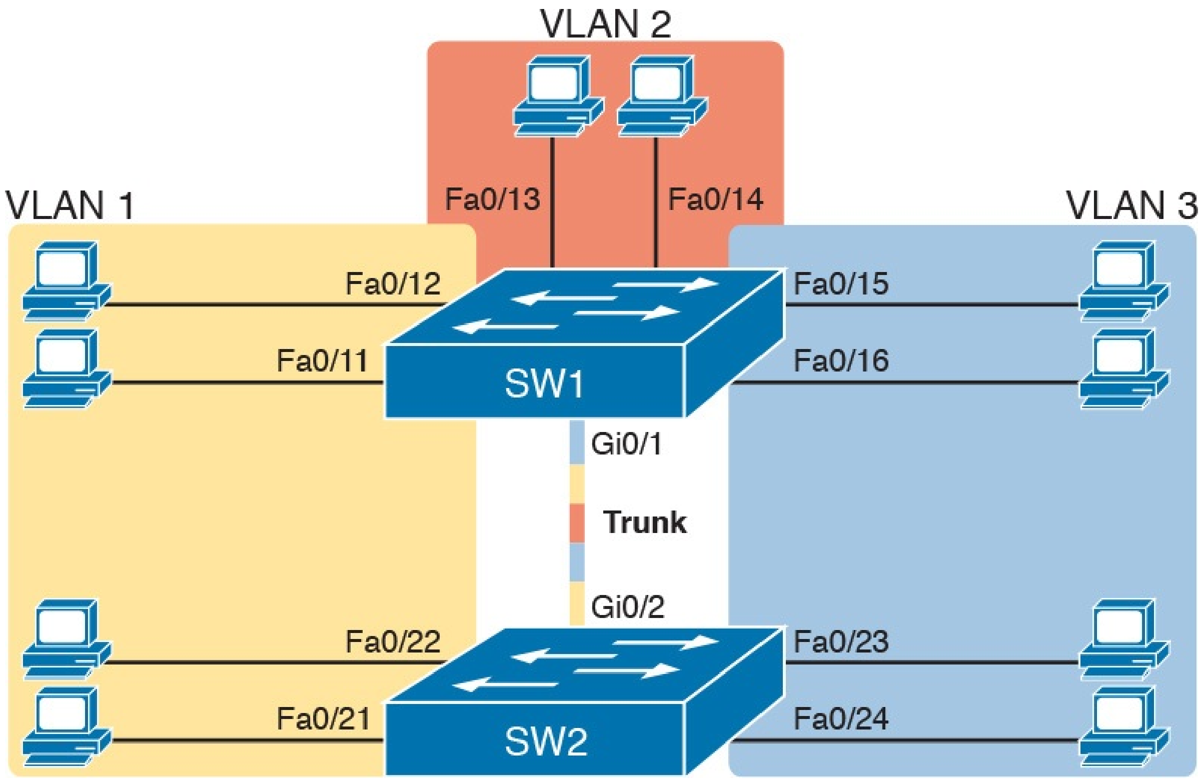


Figure 1-12 *Network with Two Switches and Three VLANs*

Example 1-3 begins by showing the two switches in Figure 1-12 with the default configuration so that the two switches do not trunk.

Example 1-3 *Initial (Default) State: Not Trunking Between SW1 and SW2*

Click here to view code image

```
SW1# show interfaces gigabit 0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: dynamic auto
```

```
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging:
enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none

! Note that the next command results in a single empty
line of output.
SW1# show interfaces trunk
SW1#
```

First, focus on the highlighted items from the output of the **show interfaces switchport** command at the beginning of Example 1-3. The output lists the default administrative mode setting of dynamic auto. Because SW2 also defaults to dynamic auto, the command lists SW1's operational status as "access," meaning that it is not trunking. ("Dynamic auto" tells both switches to sit there and wait on the other switch to start the negotiations.) The third shaded line points out the only supported type of trunking (802.1Q) on this 2960 switch. (On a switch that supports both ISL and 802.1Q, this value would by default list "negotiate," to mean that the type of encapsulation is negotiated.) Finally, the operational trunking type is listed as "native," which is a reference to the 802.1Q native VLAN.

The end of the example shows the output of the **show interfaces trunk** command, but with no output. This command lists information about all interfaces that currently operationally trunk; that is, it lists interfaces that currently use VLAN trunking. With no interfaces listed, this command also confirms that the link between switches is not trunking.

Next, consider Example 1-4, which shows the new configuration that enables trunking. In this case, SW1 is configured with the **switchport mode dynamic desirable** command, which asks the switch to both negotiate as well as to begin the negotiation process, rather than waiting on the other device. As soon as the command is issued, log messages appear showing that the interface goes down and then back up again, which happens when the interface transitions from access mode to trunk mode.

Example 1-4 *SW1 Changes from Dynamic Auto to Dynamic Desirable*

Click here to view code image

```
SW1# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW1(config)# interface gigabit 0/1
SW1(config-if)# switchport mode dynamic desirable
SW1(config-if)# ^Z
SW1#
%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed state to
  down
%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed state to
  up
SW1# show interfaces gigabit 0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
! lines omitted for brevity

! The next command formerly listed a single empty line
of output; now it lists
```

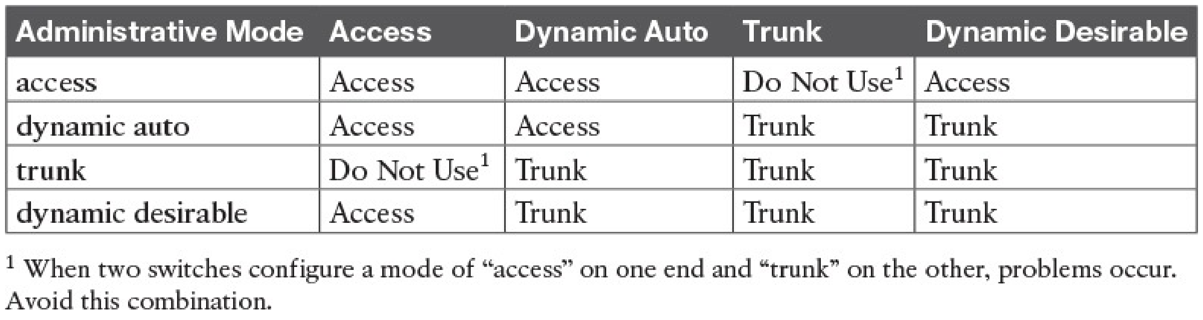
```
! information about the 1 operational trunk.
SW1# show interfaces trunk

Port        Mode             Encapsulation  Status        Na
vlan
Gi0/1       desirable        802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/1       1-4094

Port        Vlans allowed and active in management
domain
Gi0/1       1-3

Port        Vlans in spanning tree forwarding state and
not pruned
Gi0/1       1-3

SW1# show vlan id 2
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
------------------------
2    Freds-vlan                       active    Fa0/13,
Fa0/14, G0/1

VLAN Type  SAID       MTU   Parent RingNo BridgeNo
Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ----
-------- ------ ------
2    enet  100010     1500  -      -      -        -

Remote SPAN VLAN
----------------
Disabled

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

To verify whether trunking is working now, the middle of Example 1-4 lists the **show interfaces switchport** command. Note that the command still lists the administrative settings, which denote the configured values along with the operational settings, which list what the switch is currently doing. In this case, SW1 now claims to be in an operational mode of *trunk*, with an operational trunking encapsulation of dot1Q.

The end of the example shows the output of the **show vlan id 2** command, which now lists G0/1, confirming that G0/1 is now operationally trunking. The next section discusses the meaning of the output of this command.

For the exams, you should be ready to interpret the output of the **show interfaces switchport** command, realize the administrative mode implied by the output, and know whether the link should operationally trunk based on those settings. Table 1-3 lists the combinations of the trunking administrative modes and the expected operational mode (trunk or access) resulting from the configured settings. The table lists the administrative mode used on one end of the link on the left, and the administrative mode on the switch on the other end of the link across the top of the table.

Key Topic



Administrative Mode	Access	Dynamic Auto	Trunk	Dynamic Desirable
access	Access	Access	Do Not Use[1]	Access
dynamic auto	Access	Access	Trunk	Trunk
trunk	Do Not Use[1]	Trunk	Trunk	Trunk
dynamic desirable	Access	Trunk	Trunk	Trunk

[1] When two switches configure a mode of "access" on one end and "trunk" on the other, problems occur. Avoid this combination.

Table 1-3 Expected Trunking Operational Mode Based on the Configured Administrative Modes

Finally, before leaving the discussion of configuring trunks, Cisco recommends disabling trunk negotiation on most ports for better security. The majority of switch ports on most switches will be used to connect to users. As a matter of habit, you can disable DTP negotiations altogether using the **switchport nonegotiate** interface subcommand.

Implementing Interfaces Connected to Phones

This next topic is a strange topic, at least in the context of access links and trunk links. In the world of IP telephony, telephones use Ethernet ports to connect to an Ethernet network so they can use IP to send and receive voice traffic sent via IP packets. To make that work, the switch's Ethernet port acts like an access port—but at the same time, the port acts like a trunk in some ways. This last topic of the chapter works through those main concepts.

Data and Voice VLAN Concepts

Before IP telephony, a PC could sit on the same desk as a phone. The phone happened to use UTP cabling, with that phone connected to some voice device (often called a *voice switch* or a *private branch exchange [PBX]*). The PC, of course, connected using an unshielded twisted-pair (UTP) cable to the usual LAN switch that sat in the wiring closet—sometimes in the same wiring closet as the voice switch. Figure 1-13 shows the idea.

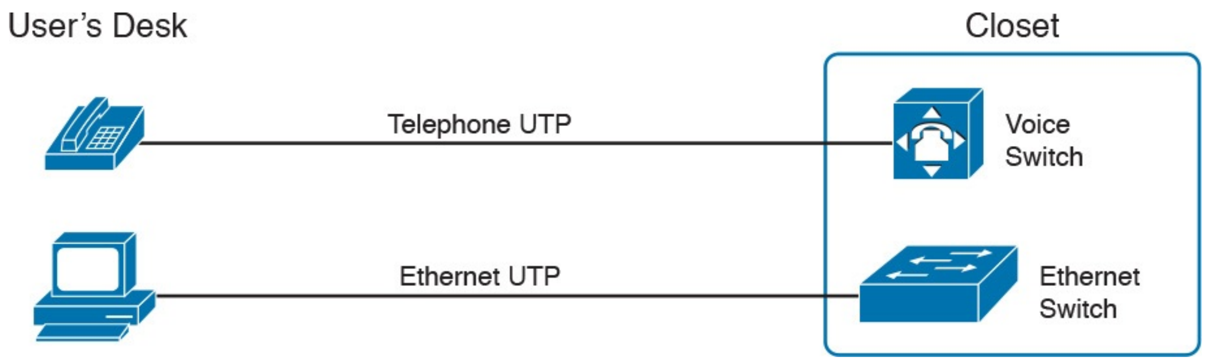


Figure 1-13 *Before IP Telephony: PC and Phone, One Cable Each, Connect to Two Different Devices*

The term *IP telephony* refers to the branch of networking in which the telephones use IP packets to send and receive voice as represented by the bits in the data portion of the IP packet. The phones connect to the network like most other end-user devices, using either Ethernet or Wi-Fi. These new IP phones did not connect via cable directly to a voice switch, instead connecting to the IP network using an Ethernet cable and an Ethernet port built in to the phone. The phones then communicated over the IP network with software that replaced the call setup and other functions of the PBX. (The current product from Cisco that perform this IP telephony control function is called *Cisco Unified Communications Manager*.)

The migration from using the already-installed telephone cabling, to these new IP phones that needed UTP cables that supported Ethernet, caused some problems in some offices. In particular:

- The older non-IP phones used a category of UTP cabling that often did not support 100-Mbps or 1000-Mbps Ethernet.
- Most offices had a single UTP cable running from the wiring closet to each desk, but now two devices (the PC and the new IP phone) both needed a cable from the desktop to the wiring closet.
- Installing a new cable to every desk would be expensive, plus you would need more switch ports.

To solve this problem, Cisco embedded small three-port switches into each phone.

IP telephones have included a small LAN switch, on the underside of the phone, since the earliest IP telephone products. Figure 1-14 shows the basic cabling, with the wiring closet cable connecting to one physical port on the embedded switch, the PC connecting with a short patch cable to the other physical port, and the phone's internal CPU connecting to an internal switch port.

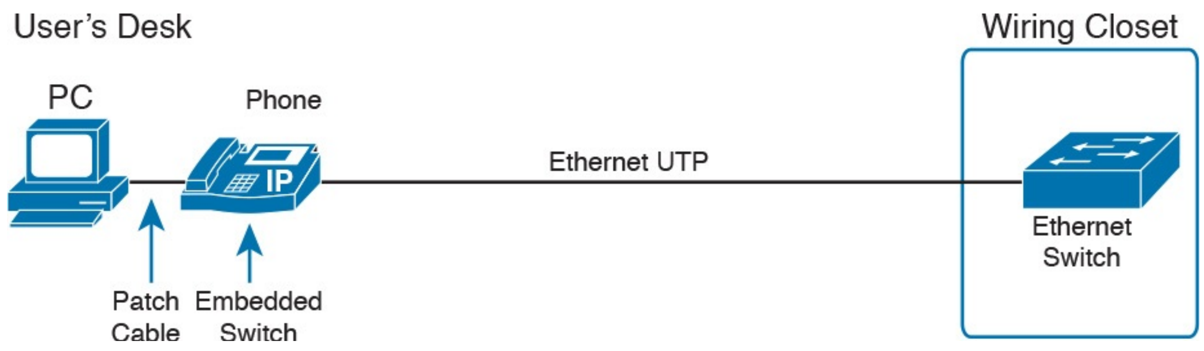


Figure 1-14 *Cabling with an IP Phone, a Single Cable, and an Integrated Switch*

Sites that use IP telephony, which includes most every company today, now have two devices off each access port. In addition, Cisco best practices for IP telephony design tell us to put the phones in one VLAN, and the PCs in a different VLAN. To make that happen, the switch port acts a little like an access link (for the PC's traffic), and a little like a trunk (for the phone's traffic). The configuration defines two VLANs on that port, as follows:

Key Topic

Data VLAN: Same idea and configuration as the access VLAN on an access port, but defined as the VLAN on that link for forwarding the traffic for the device connected to the phone on the desk (typically the user's PC).

Voice VLAN: The VLAN defined on the link for forwarding the phone's traffic. Traffic in this VLAN is typically tagged with an 802.1Q header.

Figure 1-15 illustrates this design with two VLANs on access ports that support IP telephones.

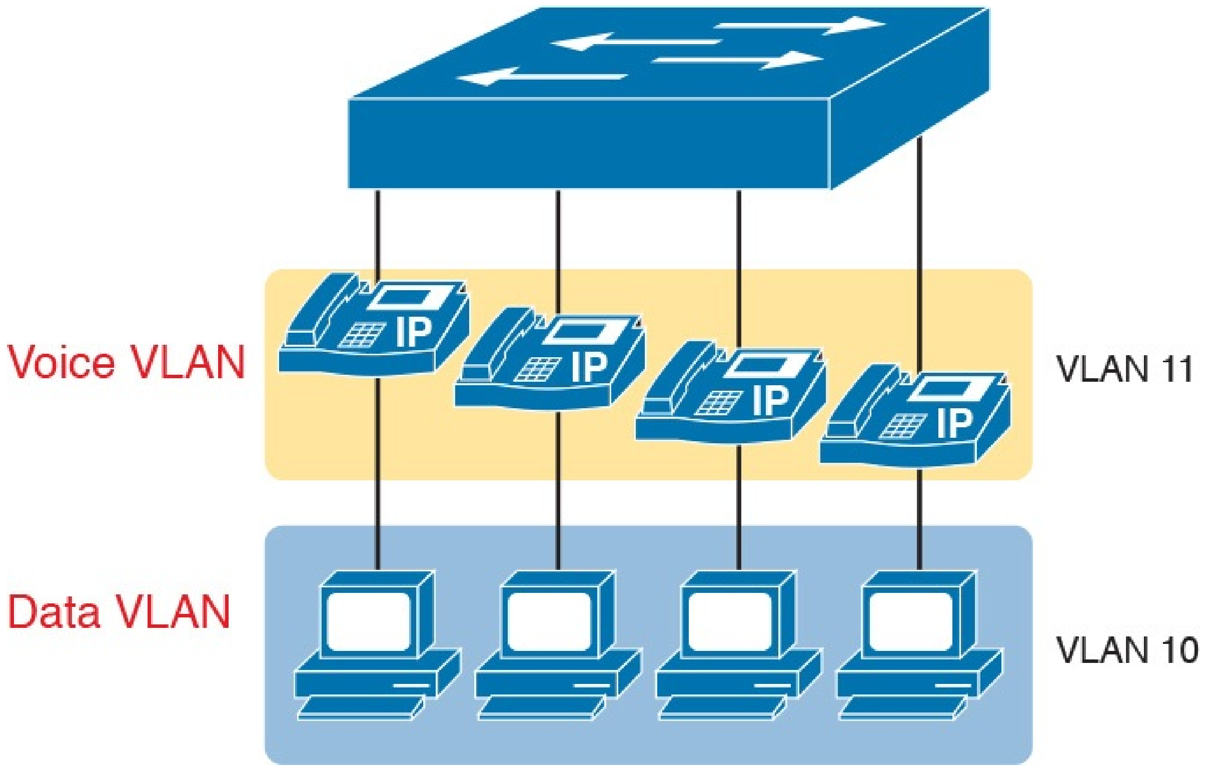


Figure 1-15 *A LAN Design, with Data in VLAN 10 and Phones in VLAN 11*

Data and Voice VLAN Configuration and Verification

Configuring a switch port to support IP phones, once you know the planned voice and data VLAN IDs, is easy. Making sense of the **show** commands once they are configured can be a challenge. The port acts like an access port in many ways. However, with most configuration options, the voice frames flow with an 802.1Q header, so that the link supports frames in both VLANs on the link. But that makes for some different **show** command output.

Example 1-5 shows an example. In this case, all four switch ports F0/1–F0/4 begin with default configuration. The configuration adds the new data and voice VLANs. The example then configures all four ports as access ports, and defines the access VLAN, which is also called the data VLAN when discussing IP telephony. Finally, the configuration includes the **switchport voice vlan 11** command, which defines the voice VLAN used on the port. The example matches Figure 1-15, using ports F0/1–F0/4.

Example 1-5 *Configuring the Voice and Data VLAN on Ports Connected to Phones*

Click here to view code image

```
SW1# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
```

```
SW1(config)# vlan 10
SW1(config-vlan)# vlan 11
SW1(config-vlan)# interface range FastEthernet0/1 - 4
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 10
SW1(config-if)# switchport voice vlan 11
SW1(config-if)#^Z
SW1#
```

Note

CDP, discussed in the ICND1 book's Chapter 33, "Device Management Protocols," must be enabled on an interface for a voice access port to work with Cisco IP Phones. CDP is enabled by default, so its configuration is not shown here.

The following list details the configuration steps for easier review and study:

Config Checklist

Step 1. Use the **vlan** *vlan-id* command in global configuration mode to create the data and voice VLANs if they do not already exist on the switch.

Step 2. Configure the data VLAN like an access VLAN, as usual:

A. Use the **interface** *type number* command in global configuration mode to move into interface configuration mode.

B. Use the **switchport access vlan** *id-number* command in interface configuration mode to define the data VLAN.

C. Use the **switchport mode access** command in interface configuration mode to make this port always operate in access mode (that is, to not trunk).

Step 3. Use the **switchport voice vlan** *id-number* command in interface configuration mode to set the voice VLAN ID.

Verifying the status of a switch port configured like Example 1-5 shows some different output compared to the pure access port and pure trunk port configurations seen earlier in this chapter. For example, the **show interfaces switchport** command shows details about the operation of an interface, including many details about access ports. Example 1-6 shows those details for port F0/4 after the configuration in Example 1-5 was added.

Example 1-6 *Verifying the Data VLAN (Access VLAN) and Voice VLAN*

Click here to view code image

```
SW1# show interfaces FastEthernet 0/4 switchport
Name: Fa0/4
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 10 (VLAN0010)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 11 (VLAN0011)
! The rest of the output is omitted for brevity
```

Working through the first three highlighted lines in the output, all those details should look familiar for any access port. The **switchport mode access** configuration command statically configures the administrative mode to be an access port, so the port of course operates as an access port. Also, as shown in the third highlighted line, the **switchport access vlan 10** configuration command defined the access mode VLAN as highlighted here.

The fourth highlighted line shows the one small new piece of information: the voice VLAN ID, as set with the **switchport voice vlan 11** command in this case. This small line of output is the only piece of information in the output that differs from the earlier access port examples in this chapter.

These ports act more like access ports than trunk ports. In fact, the **show interfaces** *type number* **switchport** command boldly proclaims, “Operational Mode: static access.” However, one other **show** command reveals just a little more about the underlying operation with 802.1Q tagging for the voice frames.

As mentioned earlier, the **show interfaces trunk** command—that is, the command that does not include a specific interface in the middle of the command—lists the operational trunks on a switch. With IP telephony ports, the ports do not show up in the list of trunks either—providing evidence that these links are *not* treated as trunks. Example 1-7 shows just such an example.

However, the **show interfaces trunk** command with the interface listed in the middle of the command, as is also shown in Example 1-7, does list some additional information. Note that in this case, the **show interfaces F0/4 trunk** command lists the status as not-trunking, but with VLANs 10 and 11 allowed on the trunk. (Normally, on an access port, only the access VLAN is listed in the “VLANs allowed on the trunk” list in the output of this command.)

Example 1-7 *Allowed VLAN List and the List of Active VLANs*

Click here to view code image

```
SW1# show interfaces trunk
SW1# show interfaces F0/4 trunk

Port        Mode             Encapsulation  Status
vlan
Fa0/4       off              802.1q         not-
trunking  1

Port        Vlans allowed on trunk
Fa0/4       10-11

Port        Vlans allowed and active in management
domain
Fa0/4       10-11

Port        Vlans in spanning tree forwarding state and
not pruned
Fa0/4       10-11
```

Summary: IP Telephony Ports on Switches

It might seem like this short topic about IP telephony and switch configuration includes a lot of small twists and turns and trivia, and it does. The most important items to remember are as follow:

Key Topic

- Configure these ports like a normal access port to begin: Configure it as a static access port and assign it an access VLAN.
- Add one more command to define the voice VLAN (**switchport voice vlan** *vlan-id*).
- Look for the mention of the voice VLAN ID, but no other new facts, in the output of the **show interfaces** *type number* **switchport** command.
- Look for both the voice and data (access) VLAN IDs in the output of the **show interfaces** *type number* **trunk** command.
- Do not expect to see the port listed in the list of operational trunks as listed by the **show interfaces trunk** command.

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book, DVD, or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 1-4 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Review Element	Review Date(s)	Resource Used
Review key topics		Book, DVD/website
Review key terms		Book, DVD/website
Answer DIKTA questions		Book, PCPT
Do labs		Blog
Review memory tables		DVD/website
Review config checklists		Book, DVD/website
Review command tables		Book

Table 1-4 Chapter Review Tracking

Review All the Key Topics

Key Topic

Key Topic Element	Description	Page Number
Figure 1-2	Basic VLAN concept	17
List	Reasons for using VLANs	17
Figure 1-5	Diagram of VLAN trunking	19
Figure 1-6	802.1Q header	20
Figure 1-9	Routing between VLANs with router-on-a-stick	23
Figure 1-10	Routing between VLANs with Layer 3 switch	24
Table 1-2	Options of the **switchport mode** command	30
Table 1-3	Expected trunking results based on the configuration of the **switchport mode** command	34
List	Definitions of data VLAN and voice VLAN	36
List	Summary of data and voice VLAN concepts, configuration, and verification	39

Table 1-5 Key Topics for Chapter 1

Key Terms You Should Know

802.1Q

trunk

trunking administrative mode

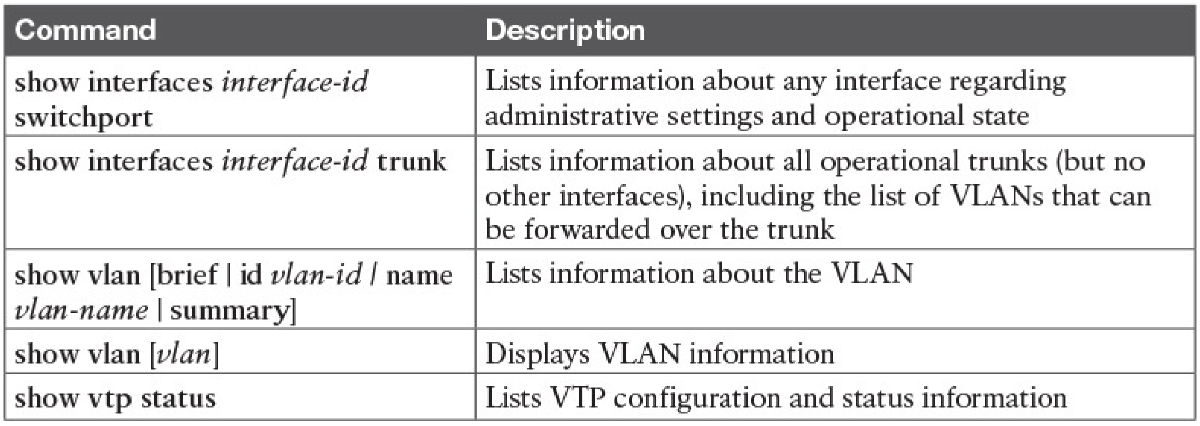
trunking operational mode
VLAN
VTP
VTP transparent mode
Layer 3 switch
access interface
trunk interface
data VLAN
voice VLAN

Command References

Tables 1-6 and 1-7 list configuration and verification commands used in this chapter, respectively. As an easy review exercise, cover the left column in a table, read the right column, and try to recall the command without looking. Then repeat the exercise, covering the right column, and try to recall what the command does.

Command	Description
vlan *vlan-id*	Global config command that both creates the VLAN and puts the CLI into VLAN configuration mode
name *vlan-name*	VLAN subcommand that names the VLAN
[no] shutdown	VLAN mode subcommand that enables (**no shutdown**) or disables (**shutdown**) the VLAN
[no] shutdown vlan *vlan-id*	Global config command that has the same effect as the **[no] shutdown** VLAN mode subcommands
vtp mode {server | client | transparent | off}	Global config command that defines the VTP mode
switchport mode {access | dynamic {auto | desirable} | trunk}	Interface subcommand that configures the trunking administrative mode on the interface
switchport access vlan *vlan-id*	Interface subcommand that statically configures the interface into that one VLAN
switchport trunk encapsulation {dot1q | isl | negotiate}	Interface subcommand that defines which type of trunking to use, assuming that trunking is configured or negotiated
switchport trunk native vlan *vlan-id*	Interface subcommand that defines the native VLAN for a trunk port
switchport nonegotiate	Interface subcommand that disables the negotiation of VLAN trunking
switchport voice vlan *vlan-id*	Interface subcommand that defines the voice VLAN on a port, meaning that the switch uses 802.1Q tagging for frames in this VLAN
switchport trunk allowed vlan {add | all | except | remove} *vlan-list*	Interface subcommand that defines the list of allowed VLANs

Table 1-6 Chapter 1 Configuration Command Reference



Command	Description
show interfaces *interface-id* switchport	Lists information about any interface regarding administrative settings and operational state
show interfaces *interface-id* trunk	Lists information about all operational trunks (but no other interfaces), including the list of VLANs that can be forwarded over the trunk
show vlan [brief | id *vlan-id* / name *vlan-name* | summary]	Lists information about the VLAN
show vlan [*vlan*]	Displays VLAN information
show vtp status	Lists VTP configuration and status information

Table 1-7 Chapter 1 EXEC Command Reference

Chapter 2. Spanning Tree Protocol Concepts

This chapter covers the following exam topics:

1.0 LAN Switching Technologies

1.3 Configure, verify, and troubleshoot STP protocols

1.3.a STP mode (PVST+ and RPVST+)

1.3.b STP root bridge selection

1.4 Configure, verify, and troubleshoot STP-related optional features

1.4.a PortFast

1.4.b BPDU guard

1.5 Configure, verify, and troubleshoot (Layer 2/Layer 3) EtherChannel

1.5.a Static

1.5.b PAGP

1.5.c LACP

Spanning Tree Protocol (STP) allows Ethernet LANs to have the added benefits of installing redundant links in a LAN, while overcoming the known problems that occur when adding those extra links. Using redundant links in a LAN design allows the LAN to keep working even when some links fail or even when some entire switches fail. Proper LAN design should add enough redundancy so that no single point of failure crashes the LAN; STP allows the design to use redundancy without causing some other problems.

STP affects many aspects of how switch forwarding logic works. Because Cisco puts the STP exam topics into the ICND2 half of the CCNA Routing and Switching exam, all the detailed examples in the ICND1 Cert Guide avoid showing redundant links in the LANs. For this ICND2 book, most of the LAN examples include redundancy. Therefore, you need to be prepared to rethink what you learned about LANs from reading the ICND1 book while thinking about LANs that have redundant links, and how STP and related features make those LANs work.

This chapter organizes the material into three sections. The first section presents core STP concepts that apply to most types of STP. STP has been improved and changed over the years, with Rapid STP (RSTP) being one major improvement. The first section looks at STP concepts without the RSTP logic added, while the second major section details RSTP concepts. The final major section discusses a small number of features that optimize and secure STP: PortFast, BPDU Guard, and EtherChannels.

As for the exam topics for this chapter, note that they all use the same three verbs: configure, verify, and troubleshoot. This chapter does not get into that level of depth on any of the specific topics, but instead lays the foundation to understand these features so that you are prepared to delve into the configuration, verification, and troubleshooting details in Chapters 3 and 4.

“Do I Know This Already?” Quiz

Take the quiz (either here, or use the PCPT software) if you want to use the score to help you decide how much time to spend on this chapter. The answers are at the bottom of the page following the quiz, and the explanations are in DVD Appendix C and in the PCPT software.

Foundation Topics Section	Questions
Spanning Tree Protocol (IEEE 802.1D)	1–4
Rapid STP (IEEE 802.1w) Concepts	5, 6
Optional STP Features	7

Table 2-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

1. Which of the following IEEE 802.1D port states are stable states used when STP has completed convergence? (Choose two answers.)
   - **a.** Blocking
   - **b.** Forwarding
   - **c.** Listening
   - **d.** Learning
   - **e.** Discarding
2. Which of the following are transitory IEEE 802.1D port states used only during the process of STP convergence? (Choose two answers.)
   - **a.** Blocking
   - **b.** Forwarding
   - **c.** Listening
   - **d.** Learning
   - **e.** Discarding
3. Which of the following bridge IDs wins election as root, assuming that the switches with these bridge IDs are in the same network?
   - **a.** 32769:0200.1111.1111
   - **b.** 32769:0200.2222.2222
   - **c.** 4097:0200.1111.1111
   - **d.** 4097:0200.2222.2222

e. 40961:0200.1111.1111

4. Which of the following facts determines how often a nonroot bridge or switch sends an 802.1D STP Hello BPDU message?

   a. The Hello timer as configured on that switch.

   b. The Hello timer as configured on the root switch.

   c. It is always every 2 seconds.

   d. The switch reacts to BPDUs received from the root switch by sending another BPDU 2 seconds after receiving the root BPDU.

5. Which of the following RSTP port states have the same name and purpose as a port state in traditional 802.1D STP? (Choose two answers.)

   a. Blocking

   b. Forwarding

   c. Listening

   d. Learning

   e. Discarding

6. RSTP adds some concepts to STP that enable ports to be used for a role if another port on the same switch fails. Which of the following statements correctly describe a port role that is waiting to take over for another port role? (Choose two answers.)

   a. An alternate port waits to become a root port.

   b. A backup port waits to become a root port.

   c. An alternate port waits to become a designated port.

   d. A backup port waits to become a designated port.

7. What STP feature causes an interface to be placed in the forwarding state as soon as the interface is physically active?

   a. STP

   b. EtherChannel

   c. Root Guard

   d. PortFast

Answers to the "Do I Know This Already?" quiz:

1 A, B **2** C, D **3** C **4** B **5** B, D **6** A, D **7** D

Foundation Topics

Spanning Tree Protocol (IEEE 802.1D)

Without Spanning Tree Protocol (STP), a LAN with redundant links would cause Ethernet frames to loop for an indefinite period of time. With STP enabled, some switches block ports so that these ports do not forward frames. STP intelligently chooses which ports block, with two goals in mind:

- All devices in a VLAN can send frames to all other devices. In other words, STP does not block too many ports, cutting off some parts of the LAN from other parts.
- Frames have a short life and do not loop around the network indefinitely.

STP strikes a balance, allowing frames to be delivered to each device, without causing the problems that occur when frames loop through the network over and over again.

STP prevents looping frames by adding an additional check on each interface before a switch uses it to send or receive user traffic. That check: If the port is in STP forwarding state in that VLAN, use it as normal; if it is in STP blocking state, however, block all user traffic and do not send or receive user traffic on that interface in that VLAN.

Note that these STP states do not change the other information you already know about switch interfaces. The interface's state of connected/notconnect does not change. The interface's operational state as either an access or trunk port does not change. STP adds this additional STP state, with the blocking state basically disabling the interface.

In many ways, those last two paragraphs sum up what STP does. However, the details of how STP does its work can take a fair amount of study and practice. This first major section of the chapter begins by explaining the need for STP and the basic ideas of what STP does to solve the problem of looping frames. The majority of this section then looks at how STP goes about choosing which switch ports to block to accomplish STP's goals.

The Need for Spanning Tree

STP prevents three common problems in Ethernet LANs. All three problems occur as a side effect of one fact: without STP, some Ethernet frames would loop around the network for a long time (hours, days, literally forever if the LAN devices and links never failed). By default, Cisco switches run STP, but you can disable STP. Do not disable it unless you know exactly what you are doing!

Just one looping frame causes what is called a *broadcast storm*. Broadcast storms happen when any kind of Ethernet frames—broadcast frames, multicast frames, or unknown-destination unicast frames—loop around a

LAN indefinitely. Broadcast storms can saturate all the links with copies of that one single frame, crowding out good frames, as well as significantly impacting end-user device performance by making the PCs process too many broadcast frames.

To help you understand how this occurs, Figure 2-1 shows a sample network in which Bob sends a broadcast frame. The dashed lines show how the switches forward the frame when STP does not exist.

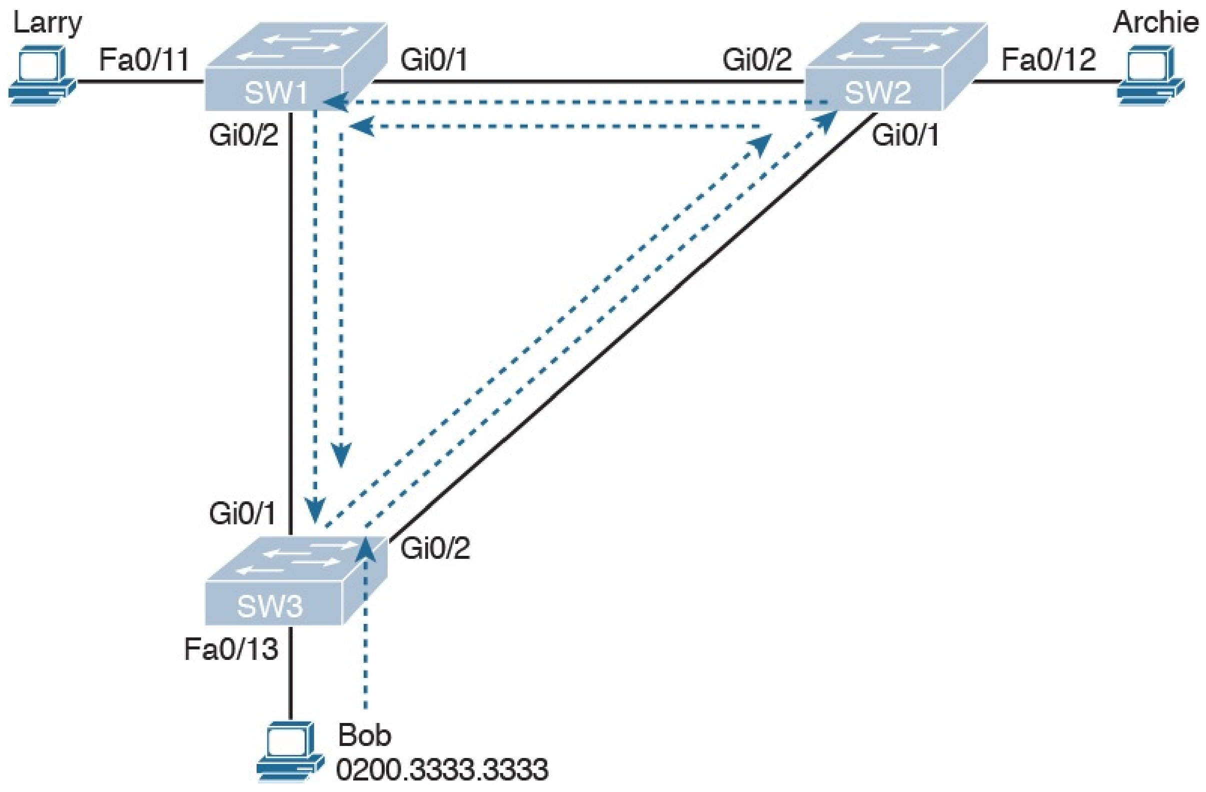


Figure 2-1 *Broadcast Storm*

Note

Bob's original broadcast would also be forwarded around the other direction as well, with SW3 sending a copy of the original frame out its Gi0/1 port. To reduce clutter, Figure 2-1 does not show that frame.

Remember that LAN switch? That logic tells switches to flood broadcasts out all interfaces in the same VLAN except the interface in which the frame arrived. In Figure 2-1, that means SW3 forwards Bob's frame to SW2, SW2 forwards the frame to SW1, SW1 forwards the frame back to SW3, and SW3 forwards it back to SW2 again.

When broadcast storms happen, frames like the one in Figure 2-1 keep looping until something changes—someone shuts down an interface, reloads a switch, or does something else to break the loop. Also note that the same event happens in the opposite direction. When Bob sends the original frame, SW3 also forwards a copy to SW1, SW1 forwards it to SW2, and so on.

The storm also causes a much more subtle problem called *MAC table instability*. MAC table instability means that the switches' MAC address tables keep changing, because frames with the same source MAC arrive on different ports. To see why, follow this example, in which SW3 begins Figure 2-1 with a MAC table entry for Bob, at the bottom of the figure, associated with port Fa0/13:

```
0200.3333.3333   Fa0/13   VLAN 1
```

However, now think about the switch-learning process that occurs when the looping frame goes to SW2, then SW1, and then back into SW3's Gi0/1 interface. SW3 thinks, "Hmm... the source MAC address is 0200.3333.3333, and it came in my Gi0/1 interface. Update my MAC table!" This results in the following entry on SW3, with interface Gi0/1 instead of Fa0/13:

```
0200.3333.3333   Gi0/1   VLAN 1
```

At this point, SW3 itself cannot correctly deliver frames to Bob's MAC address. At that instant, if a frame arrives at SW3 destined for Bob—a different frame than the looping frame that causes the problems—SW3 incorrectly forwards the frame out Gi0/1 to SW1, creating even more congestion.

The looping frames in a broadcast storm also cause a third problem: multiple copies of the frame arrive at the destination. Consider a case in which Bob sends a frame to Larry but none of the switches know Larry's MAC address. Switches flood frames sent to unknown destination unicast MAC addresses. When Bob sends the frame destined for Larry's MAC address, SW3 sends a copy to both SW1 and SW2. SW1 and SW2 also flood the frame, causing copies of the frame to loop. SW1 also sends a copy of each frame out Fa0/11 to Larry. As a result, Larry gets multiple copies of the frame, which may result in an application failure, if not more pervasive networking problems.

Table 2-2 summarizes the main three classes of problems that occur when STP is not used in a LAN that has redundancy.

Problem	Description
Broadcast storms	The forwarding of a frame repeatedly on the same links, consuming significant parts of the links' capacities
MAC table instability	The continual updating of a switch's MAC address table with incorrect entries, in reaction to looping frames, resulting in frames being sent to the wrong locations
Multiple frame transmission	A side effect of looping frames in which multiple copies of one frame are delivered to the intended host, confusing the host

Table 2-2 Three Classes of Problems Caused by Not Using STP in Redundant LANs

What IEEE 802.1D Spanning Tree Does

STP prevents loops by placing each switch port in either a forwarding state or a blocking state. Interfaces in the forwarding state act as normal, forwarding and receiving frames. However, interfaces in a blocking state do not process any frames except STP messages (and some other overhead messages). Interfaces that block do not forward user frames, do not learn MAC addresses of received frames, and do not process received user frames.

Figure 2-2 shows a simple STP tree that solves the problem shown in Figure 2-1 by placing one port on SW3 in the blocking state.

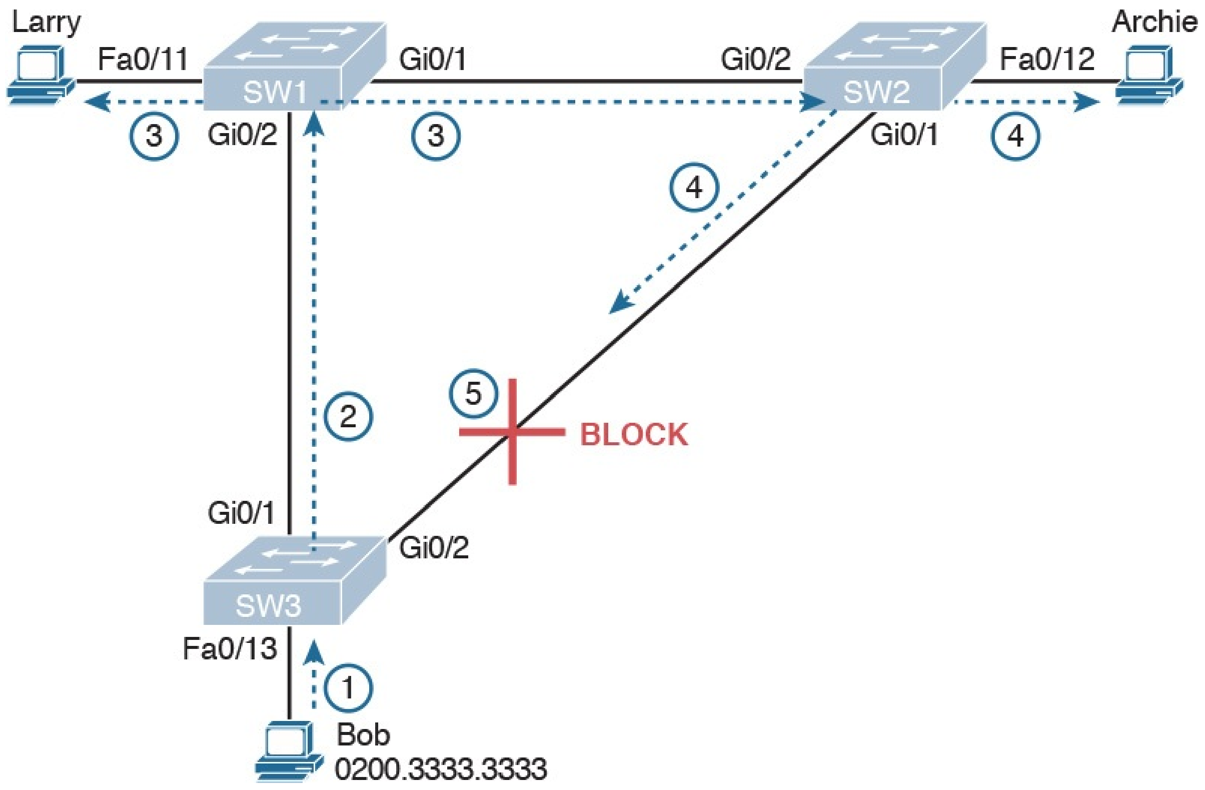


Figure 2-2 *What STP Does: Blocks a Port to Break the Loop*

Now when Bob sends a broadcast frame, the frame does not loop. As shown in the steps in the figure:

Step 1. Bob sends the frame to SW3.

Step 2. SW3 forwards the frame only to SW1, but not out Gi0/2 to SW2, because SW3's Gi0/2 interface is in a blocking state.

Step 3. SW1 floods the frame out both Fa0/11 and Gi0/1.

Step 4. SW2 floods the frame out Fa0/12 and Gi0/1.

Step 5. SW3 physically receives the frame, but it ignores the frame received from SW2 because SW3's Gi0/2 interface is in a blocking state.

With the STP topology in Figure 2-2, the switches simply do not use the link between SW2 and SW3 for traffic in this VLAN, which is the minor negative side effect of STP. However, if either of the other two links fails, STP converges so that SW3 forwards instead of blocks on its Gi0/2 interface.

Note

The term *STP convergence* refers to the process by which the switches collectively realize that something has changed in the LAN topology and determine whether they need to change which ports block and which ports forward.

That completes the description of what STP does, placing each port into either a forwarding or blocking state. The more interesting question, and the one that takes a lot more work to understand, is the question of how and why STP makes its choices. How does STP manage to make switches block or forward on each interface? And how does it converge to change state from blocking to forwarding to take advantage of redundant links in response to network outages? The following sections answer these questions.

How Spanning Tree Works

The STP algorithm creates a spanning tree of interfaces that forward frames. The tree structure of forwarding interfaces creates a single path to and from each Ethernet link, just like you can trace a single path in a living, growing tree from the base of the tree to each leaf.

Note

STP was created before LAN switches even existed. In those days, Ethernet bridges used STP. Today, switches play the same role as bridges, implementing STP. However, many STP terms still refer to bridge. For the purposes of STP and this chapter, consider the terms *bridge* and *switch* synonymous.

The process used by STP, sometimes called the *spanning-tree algorithm* (STA), chooses the interfaces that should be placed into a forwarding state. For any interfaces not chosen to be in a forwarding state, STP places the interfaces in blocking state. In other words, STP simply picks which interfaces should forward, and any interfaces left over go to a blocking state.

STP uses three criteria to choose whether to put an interface in forwarding state:

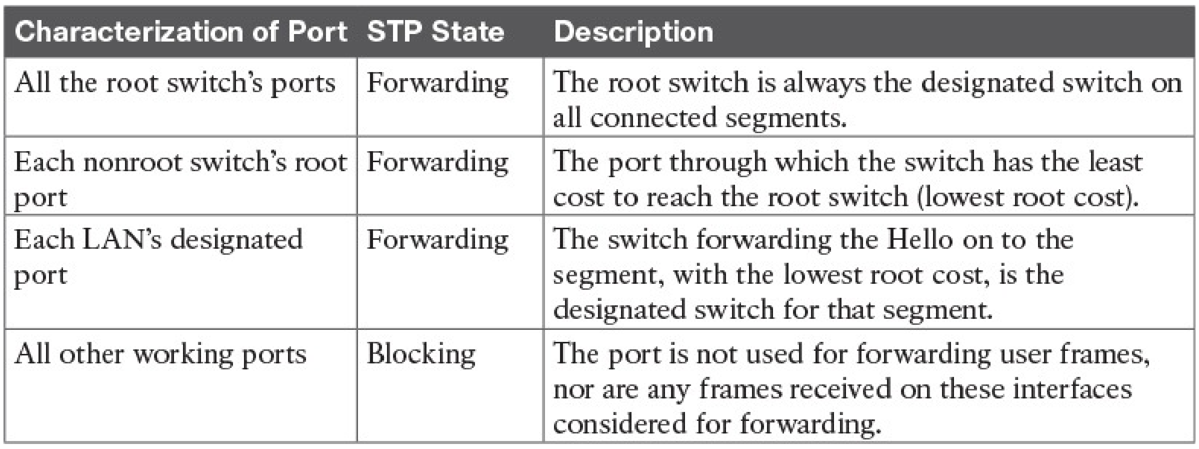
- STP elects a root switch. STP puts all working interfaces on the root switch in forwarding state.
- Each nonroot switch considers one of its ports to have the least administrative cost between itself and the root switch. The cost is called that switch's *root cost*. STP places its port that is part of the least root cost path, called that switch's *root port* (RP), in forwarding state.
- Many switches can attach to the same Ethernet segment, but in modern networks, normally two switches connect to each link. The switch with the lowest root cost, as compared with the other switches attached to the same link, is placed in forwarding state. That switch is the designated switch, and that switch's interface, attached to that segment, is called the *designated port* (DP).

Note

The real reason the root switches place all working interfaces in a forwarding state is that all its interfaces will become DPs, but it is easier to just remember that all the root switches' working interfaces will forward frames.

All other interfaces are placed in blocking state. Table 2-3 summarizes the reasons STP places a port in forwarding or blocking state.

Key Topic



Characterization of Port	STP State	Description
All the root switch's ports	Forwarding	The root switch is always the designated switch on all connected segments.
Each nonroot switch's root port	Forwarding	The port through which the switch has the least cost to reach the root switch (lowest root cost).
Each LAN's designated port	Forwarding	The switch forwarding the Hello on to the segment, with the lowest root cost, is the designated switch for that segment.
All other working ports	Blocking	The port is not used for forwarding user frames, nor are any frames received on these interfaces considered for forwarding.

Table 2-3 STP: Reasons for Forwarding or Blocking

Note

STP only considers working interfaces (those in a connected state). Failed interfaces (for example, interfaces with no cable installed) or administratively shutdown interfaces are instead placed into an STP disabled state. So, this section uses the term *working ports* to refer to interfaces that could forward frames if STP placed the interface into a forwarding state.

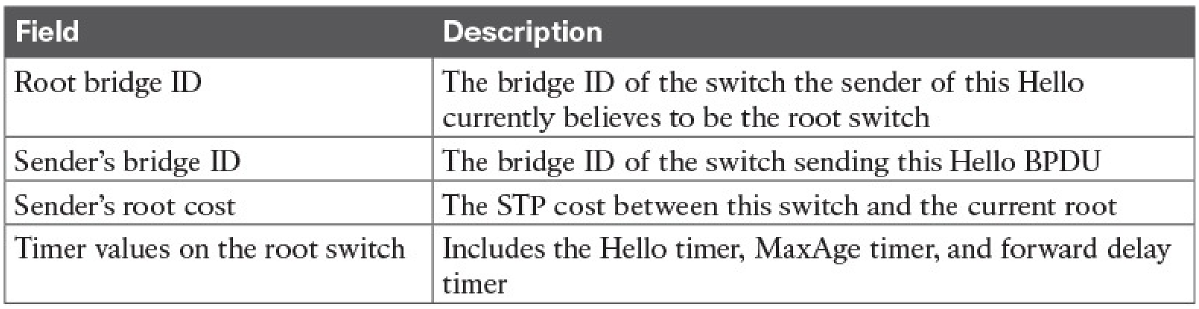
The STP Bridge ID and Hello BPDU

The STA begins with an election of one switch to be the root switch. To better understand this election process, you need to understand the STP messages sent between switches as well as the concept and format of the identifier used to uniquely identify each switch.

The STP *bridge ID* (BID) is an 8-byte value unique to each switch. The bridge ID consists of a 2-byte priority field and a 6-byte system ID, with the system ID being based on a universal (burned-in) MAC address in each switch. Using a burned-in MAC address ensures that each switch's bridge ID will be unique.

STP defines messages called *bridge protocol data units* (BPDU), which switches use to exchange information with each other. The most common BPDU, called a Hello BPDU, lists many details, including the sending switch's BID. By listing its own unique BID, switches can tell which switch sent which Hello BPDU. Table 2-4 lists some of the key information in the Hello BPDU.

Key Topic



Field	Description
Root bridge ID	The bridge ID of the switch the sender of this Hello currently believes to be the root switch
Sender's bridge ID	The bridge ID of the switch sending this Hello BPDU
Sender's root cost	The STP cost between this switch and the current root
Timer values on the root switch	Includes the Hello timer, MaxAge timer, and forward delay timer

Table 2-4 Fields in the STP Hello BPDU

For the time being, just keep the first three items from Table 2-4 in mind as the following sections work through the three steps in how STP chooses the interfaces to place into a forwarding state. Next, the text examines the three main steps in the STP process.

Electing the Root Switch

Switches elect a root switch based on the BIDs in the BPDUs. The root switch is the switch with the lowest numeric value for the BID. Because the two-part BID starts with the priority value, essentially the switch with the lowest priority becomes the root. For example, if one switch has priority 4096, and another switch has priority 8192, the switch with priority 4096 wins, regardless of what MAC address was used to create the BID for each switch.

If a tie occurs based on the priority portion of the BID, the switch with the lowest MAC address portion of the BID is the root. No other tiebreaker should be needed because switches use one of their own universal (burned-in) MAC addresses as the second part of their BIDs. So if the priorities tie, and one switch uses a MAC address of 0200.0000.0000 as part of the BID and the other uses 0911.1111.1111, the first switch (MAC 0200.0000.0000) becomes the root switch.

STP elects a root switch in a manner not unlike a political election. The process begins with all switches claiming to be the root by sending Hello BPDUs listing their own BID as the root BID. If a switch hears a Hello that lists a better (lower) BID, that switch stops advertising itself as root and starts forwarding the superior Hello. The Hello sent by the better switch lists the better switch's BID as the root. It works like a political race in which a less-popular candidate gives up and leaves the race, throwing his support behind the more popular candidate. Eventually, everyone agrees which switch has the best (lowest) BID, and everyone supports the elected switch—which is where the political race analogy falls apart.

Note

A better Hello, meaning that the listed root's BID is better (numerically lower), is called a *superior Hello*; a worse Hello, meaning that the listed root’s BID is not as good (numerically higher), is called an *inferior Hello*.

Figure 2-3 shows the beginning of the root election process. In this case, SW1 has advertised itself as root, as have SW2 and SW3. However, SW2 now believes that SW1 is a better root, so SW2 is now forwarding the Hello originating at SW1. So, at this point, the figure shows SW1 is saying Hello, claiming to be root; SW2 agrees, and is forwarding SW1’s Hello that lists SW1 as root; but, SW3 is still claiming to be best, sending its own Hello BPDUs, listing SW3’s BID as the root.

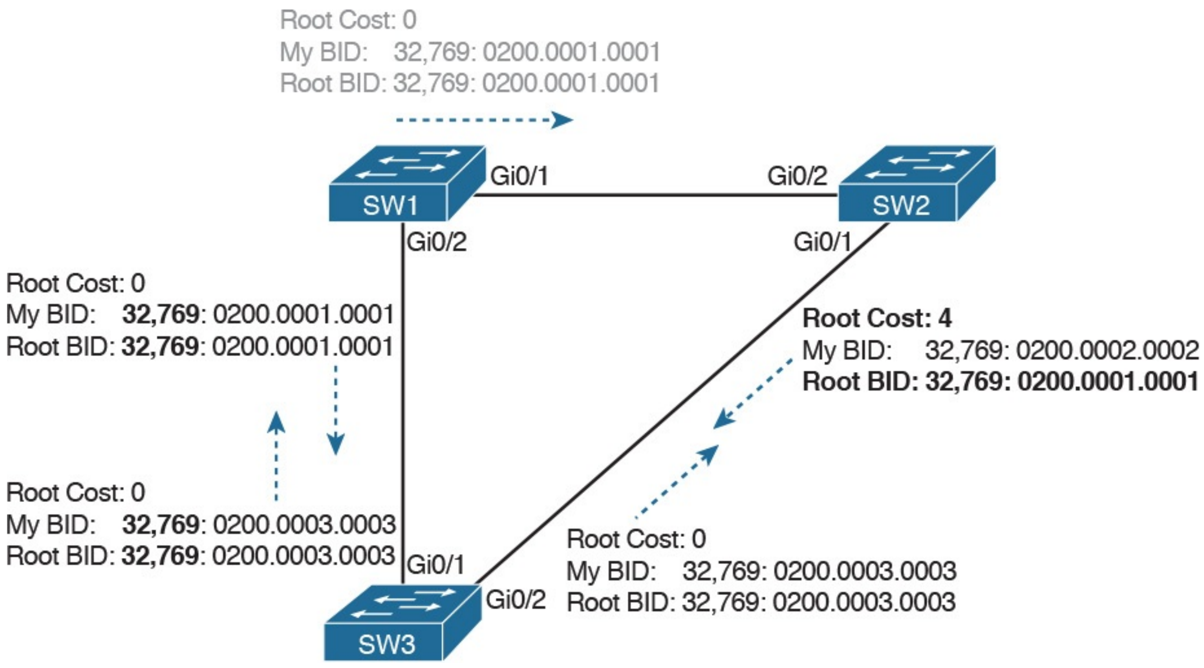

Root Cost: 0
My BID: 32,769: 0200.0001.0001
Root BID: 32,769: 0200.0001.0001
SW1
Gi0/1
Gi0/2
SW2
Gi0/2
Gi0/1
Root Cost: 0
My BID: 32,769: 0200.0001.0001
Root BID: 32,769: 0200.0001.0001
Root Cost: 4
My BID: 32,769: 0200.0002.0002
Root BID: 32,769: 0200.0001.0001
Root Cost: 0
My BID: 32,769: 0200.0003.0003
Root BID: 32,769: 0200.0003.0003
Root Cost: 0
My BID: 32,769: 0200.0003.0003
Root BID: 32,769: 0200.0003.0003
Gi0/1
Gi0/2
SW3

Figure 2-3 *Beginnings of the Root Election Process*

Two candidates still exist in Figure 2-3: SW1 and SW3. So who wins? Well, from the BID, the lower-priority switch wins; if a tie occurs, the lower MAC address wins. As shown in the figure, SW1 has a lower BID (32769:0200.0001.0001) than SW3 (32769:0200.0003.0003), so SW1 wins, and SW3 now also believes that SW1 is the better switch. Figure 2-4 shows the resulting Hello messages sent by the switches.

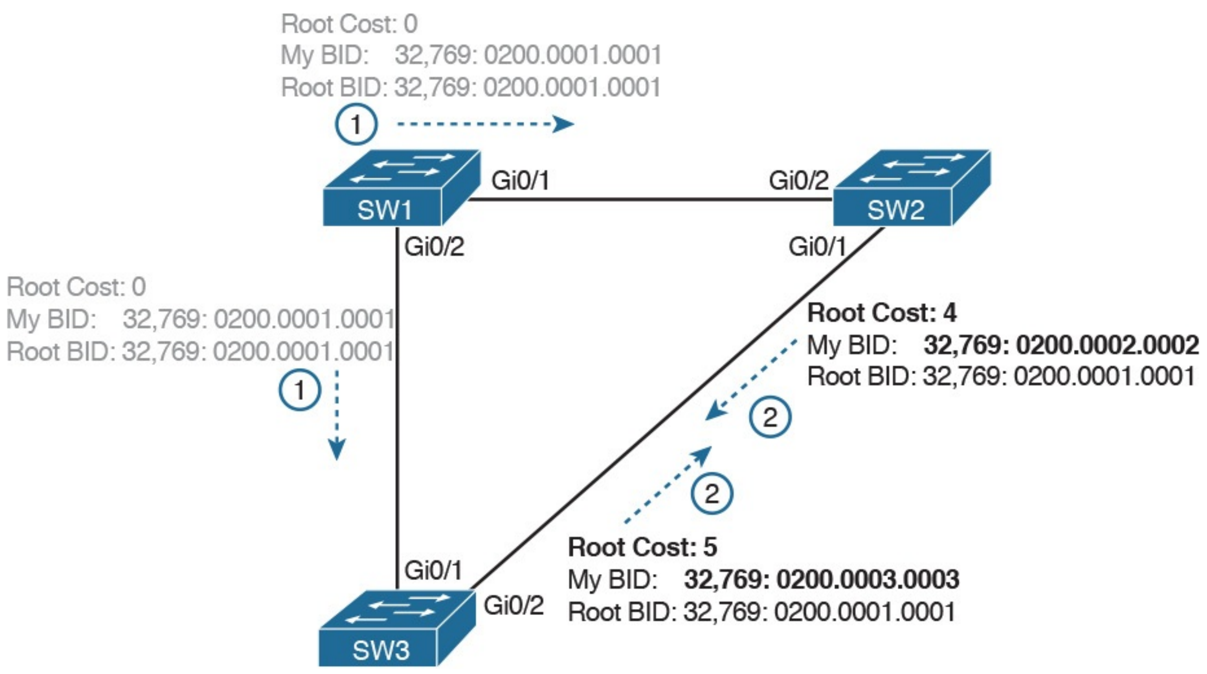

Root Cost: 0
My BID: 32,769: 0200.0001.0001
Root BID: 32,769: 0200.0001.0001
1
Gi0/1
Gi0/2
SW1
SW2
Gi0/2
Gi0/1
Root Cost: 0
My BID: 32,769: 0200.0001.0001
Root BID: 32,769: 0200.0001.0001
1
Root Cost: 4
My BID: 32,769: 0200.0002.0002
Root BID: 32,769: 0200.0001.0001
2
2
Root Cost: 5
My BID: 32,769: 0200.0003.0003
Root BID: 32,769: 0200.0001.0001
Gi0/1
Gi0/2
SW3

Figure 2-4 *SW1 Wins the Election*

After the election is complete, only the root switch continues to originate STP Hello BPDU messages. The other switches receive the Hellos, update the sender's BID field (and root cost field), and forward the Hellos out other interfaces. The figure reflects this fact, with SW1 sending Hellos at Step 1, and SW2 and SW3 independently forwarding the Hello out their other interfaces at Step 2.

Summarizing, the root election happens through each switch claiming to be root, with the best switch being elected based on the numerically lowest BID. Breaking down the BID into its components, the comparisons can be made as

Key Topic

- The lowest priority
- If that ties, the lowest switch MAC address

Choosing Each Switch's Root Port

The second part of the STP process occurs when each nonroot switch chooses its one and only *root port*. A switch's RP is its interface through which it has the least STP cost to reach the root switch (least root cost).

The idea of a switch's cost to reach the root switch can be easily seen for humans. Just look at a network diagram that shows the root switch, lists the STP cost associated with each switch port, and identifies the nonroot switch in question. Switches use a different process than looking at a network diagram, of course, but using a diagram can make it easier to learn the idea. Figure 2-5 shows just such a figure, with the same three switches shown in the last several figures. SW1 has already won the election as root, and the figure considers the cost from SW3's perspective.

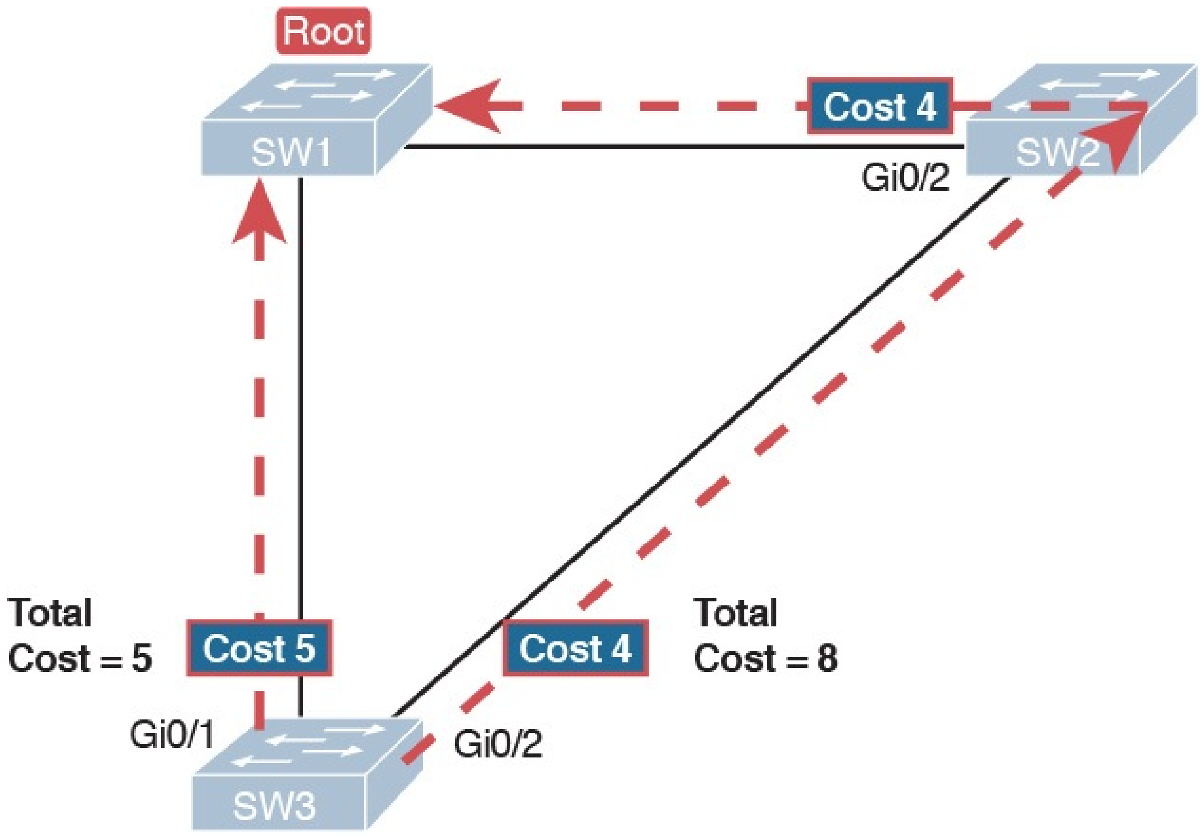


Figure 2-5 *How a Human Might Calculate STP Cost from SW3 to the Root (SW1)*

SW3 has two possible physical paths to send frames to the root switch: the direct path to the left, and the indirect path to the right through switch SW2. The cost is the sum of the costs of all the *switch ports the frame would exit* if it flowed over that path. (The calculation ignores the inbound ports.) As you can see, the cost over the direct path out SW3's G0/1 port has a total cost of 5, and the other path has a total cost of 8. SW3 picks its G0/1 port as root port because it is the port that is part of the least-cost path to send frames to the root switch.

Switches come to the same conclusion, but using a different process. Instead, they add their local interface STP cost to the root cost listed in each received Hello BPDU. The STP port cost is simply an integer value assigned to each interface, per VLAN, for the purpose of providing an objective measurement that allows STP to choose which interfaces to add to the STP topology. The switches also look at their neighbor's root cost, as announced in Hello BPDUs received from each neighbor.

Figure 2-6 shows an example of how switches calculate their best root cost and then choose their root port, using the same topology and STP costs as shown in Figure 2-5. STP on SW3 calculates its cost to reach the root over the two possible paths by adding the advertised cost (in Hello messages) to the interface costs listed in the figure.

Key Topic

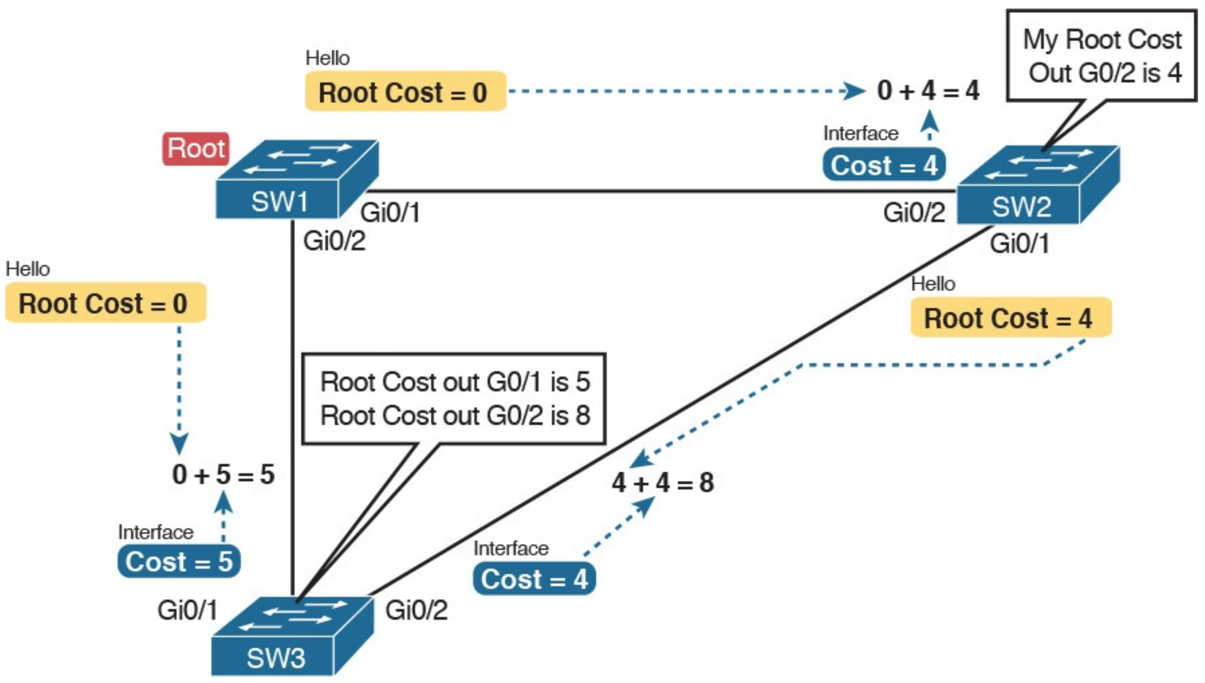


Figure 2-6 *How STP Actually Calculates the Cost from SW3 to the Root*

Focus on the process for a moment. The root switch sends Hellos, with a listed root cost of 0. The idea is that the root's cost to reach itself is 0.

Next, look on the left of the figure. SW3 takes the received cost (0) from the Hello sent by SW1, and adds the interface cost (5) of the interface on which that Hello was received. SW3 calculates that the cost to reach the root switch, out that port (G0/1), is 5.

On the right side, SW2 has realized its best cost to reach the root is cost 4. So, when SW2 forwards the Hello toward SW3, SW2 lists a root cost 4. SW3's STP port cost on port G0/2 is 4, so SW3 determines a total cost to reach root out its G0/2 port of 8.

As a result of the process depicted in Figure 2-6, SW3 chooses Gi0/1 as its RP, because the cost to reach the root switch through that port (5) is lower than the other alternative (Gi0/2, cost 8). Similarly, SW2 chooses Gi0/2 as its RP, with a cost of 4 (SW1's advertised cost of 0 plus SW2's Gi0/2 interface cost of 4). Each switch places its root port into a forwarding state.

In more complex topologies, the choice of root port will not be so obvious. Chapter 4, "LAN Troubleshooting," discusses these more complex examples, including the tiebreakers to use if the root costs tie.

Choosing the Designated Port on Each LAN Segment

STP’s final step to choose the STP topology is to choose the designated port on each LAN segment. The designated port (DP) on each LAN segment is the switch port that advertises the lowest-cost Hello onto a LAN segment. When a nonroot switch forwards a Hello, the nonroot switch sets the root cost field in the Hello to that switch’s cost to reach the root. In effect, the switch with the lower cost to reach the root, among all switches connected to a segment, becomes the DP on that segment.

For example, earlier Figure 2-4 shows in bold text the parts of the Hello messages from both SW2 and SW3 that determine the choice of DP on that segment. Note that both SW2 and SW3 list their respective cost to reach the root switch (cost 4 on SW2 and cost 5 on SW3). SW2 lists the lower cost, so SW2’s Gi0/1 port is the designated port on that LAN segment.

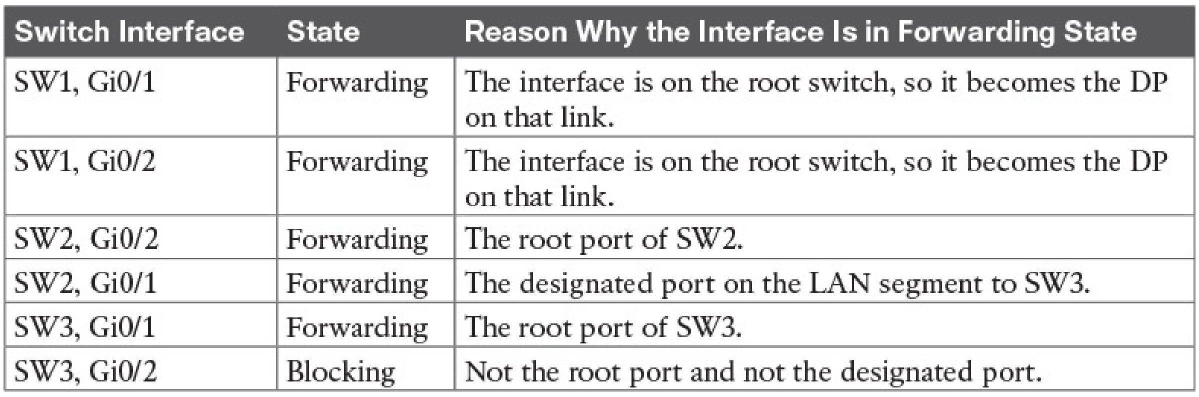
All DPs are placed into a forwarding state; so in this case, SW2’s Gi0/1 interface will be in a forwarding state.

If the advertised costs tie, the switches break the tie by choosing the switch with the lower BID. In this case, SW2 would also have won, with a BID of 32769:0200.0002.0002 versus SW3’s 32769:0200.0003.0003.

Note

Two additional tiebreakers are needed in some cases, although these would be unlikely today. A single switch can connect two or more interfaces to the same collision domain by connecting to a hub. In that case, the one switch hears its own BPDUs. So, if a switch ties with itself, two additional tiebreakers are used: the lowest interface STP priority and, if that ties, the lowest internal interface number.

The only interface that does not have a reason to be in a forwarding state on the three switches in the examples shown in Figures 2-3 through 2-6 is SW3’s Gi0/2 port. So, the STP process is now complete. Table 2-5 outlines the state of each port and shows why it is in that state.



Switch Interface	State	Reason Why the Interface Is in Forwarding State
SW1, Gi0/1	Forwarding	The interface is on the root switch, so it becomes the DP on that link.
SW1, Gi0/2	Forwarding	The interface is on the root switch, so it becomes the DP on that link.
SW2, Gi0/2	Forwarding	The root port of SW2.
SW2, Gi0/1	Forwarding	The designated port on the LAN segment to SW3.
SW3, Gi0/1	Forwarding	The root port of SW3.
SW3, Gi0/2	Blocking	Not the root port and not the designated port.

Table 2-5 State of Each Interface

Influencing and Changing the STP Topology

Switches do not just use STP once and never again. The switches continually watch for changes. Those changes can be because a link or switch fails or it can be a new link that can now be used. The configuration can change in a way that changes the STP topology. This section briefly discusses the kinds of things that change the STP topology, either through configuration or through changes in the status of devices and links in the LAN.

Making Configuration Changes to Influence the STP Topology

The network engineers can choose to change the STP settings to then change the choices STP makes in a given LAN. Two main tools available to the engineer are to configure the bridge ID and to change STP port costs.

Switches have a way to create a default BID, by taking a default priority value, and adding a universal MAC address that comes with the switch hardware. However, engineers typically want to choose which switch becomes the root. Chapter 3, "Spanning Tree Protocol Implementation," shows how to configure a Cisco switch to override its default BID setting to make a switch become root.

Port costs also have default values, per port, per VLAN. You can configure these port costs, or you can use the default values. Table 2-6 lists the default port costs suggested by IEEE. IOS on Cisco switches has long used the default settings as defined in the 1998 version of the 802.1D standard. The newer standard, useful when using links faster than 10 Gbps, can be used by adding a single configuration command to each switch (**spanning-tree pathcost method long**).

Key Topic

Ethernet Speed	IEEE Cost: 1998 (and Before)	IEEE Cost: 2004 (and After)
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2000
100 Gbps	N/A	200
1 Tbps	N/A	20

Table 2-6 Default Port Costs According to IEEE

With STP enabled, all working switch interfaces will settle into an STP forwarding or blocking state, even access ports. For switch interfaces connected to hosts or routers, which do not use STP, the switch still forwards Hellos on to those interfaces. By virtue of being the only device sending a Hello onto that LAN segment, the switch is sending the least-cost Hello on to that LAN segment, making the switch become the designated port on that LAN segment. So, STP puts working access interfaces into a forwarding state as a result of the designated port part of the STP process.

Reacting to State Changes That Affect the STP Topology

Once the engineer has finished all STP configuration, the STP topology should settle into a stable state and not change, at least until the network topology changes. This section examines the ongoing operation of STP while the network is stable, and then it covers how STP converges to a new topology when something changes.

The root switch sends a new Hello BPDU every 2 seconds by default. Each nonroot switch forwards the Hello on all DPs, but only after changing items listed in the Hello. The switch sets the root cost to that local switch's calculated root cost. The switch also sets the "sender's bridge ID" field to its own bridge ID. (The root's bridge ID field is not changed.)

By forwarding the received (and changed) Hellos out all DPs, all switches continue to receive Hellos every 2 seconds. The following steps summarize the steady-state operation when nothing is currently changing in the STP topology:

Key Topic

Step 1. The root creates and sends a Hello BPDU, with a root cost of 0, out all its working interfaces (those in a forwarding state).

Step 2. The nonroot switches receive the Hello on their root ports. After changing the Hello to list their own BID as the sender's BID, and listing that switch's root cost, the switch forwards the Hello out all designated ports.

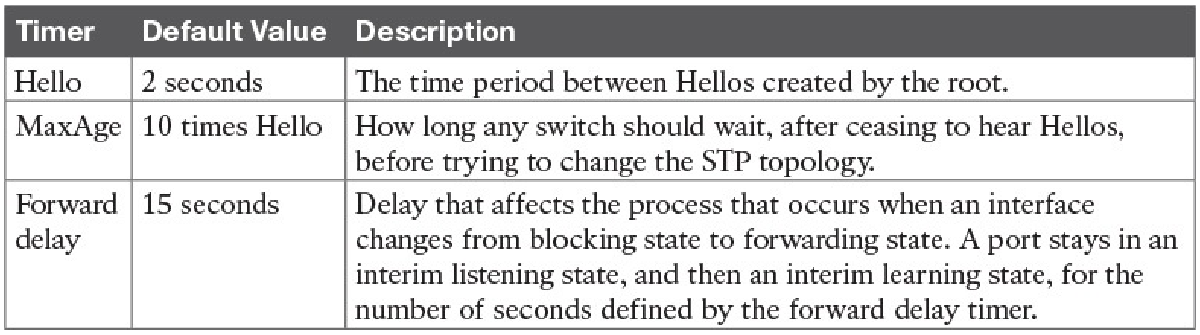
Step 3. Steps 1 and 2 repeat until something changes.

Each switch relies on these periodically received Hellos from the root as a way to know that its path to the root is still working. When a switch ceases to receive the Hellos, or receives a Hello that lists different details, something has failed, so the switch reacts and starts the process of changing the spanning-tree topology.

How Switches React to Changes with STP

For various reasons, the convergence process requires the use of three timers. Note that all switches use the timers as dictated by the root switch, which the root lists in its periodic Hello BPDU messages. Table 2-7 describes the timers.

Key Topic



Timer	Default Value	Description
Hello	2 seconds	The time period between Hellos created by the root.
MaxAge	10 times Hello	How long any switch should wait, after ceasing to hear Hellos, before trying to change the STP topology.
Forward delay	15 seconds	Delay that affects the process that occurs when an interface changes from blocking state to forwarding state. A port stays in an interim listening state, and then an interim learning state, for the number of seconds defined by the forward delay timer.

Table 2-7 STP Timers

If a switch does not get an expected Hello BPDU within the Hello time, the switch continues as normal. However, if the Hellos do not show up again within MaxAge time, the switch reacts by taking steps to change the STP topology. With default settings, MaxAge is 20 seconds (10 times the default Hello timer of 2 seconds). So, a switch would go 20 seconds without hearing a Hello before reacting.

After MaxAge expires, the switch essentially makes all its STP choices again, based on any Hellos it receives from other switches. It reevaluates which switch should be the root switch. If the local switch is not the root, it chooses its RP. And it determines whether it is DP on each of its other links. The best way to describe STP convergence is to show an example using the same familiar topology. Figure 2-7 shows the same familiar figure, with SW3's Gi0/2 in a blocking state, but SW1's Gi0/2 interface has just failed.

SW3 reacts to the change because SW3 fails to receive its expected Hellos on its Gi0/1 interface. However, SW2 does not need to react because SW2 continues to receive its periodic Hellos in its Gi0/2 interface. In this case, SW3 reacts either when MaxAge time passes without hearing the Hellos, or as soon as SW3 notices that interface Gi0/1 has failed. (If the interface fails, the switch can assume that the Hellos will not be arriving in that interface anymore.)

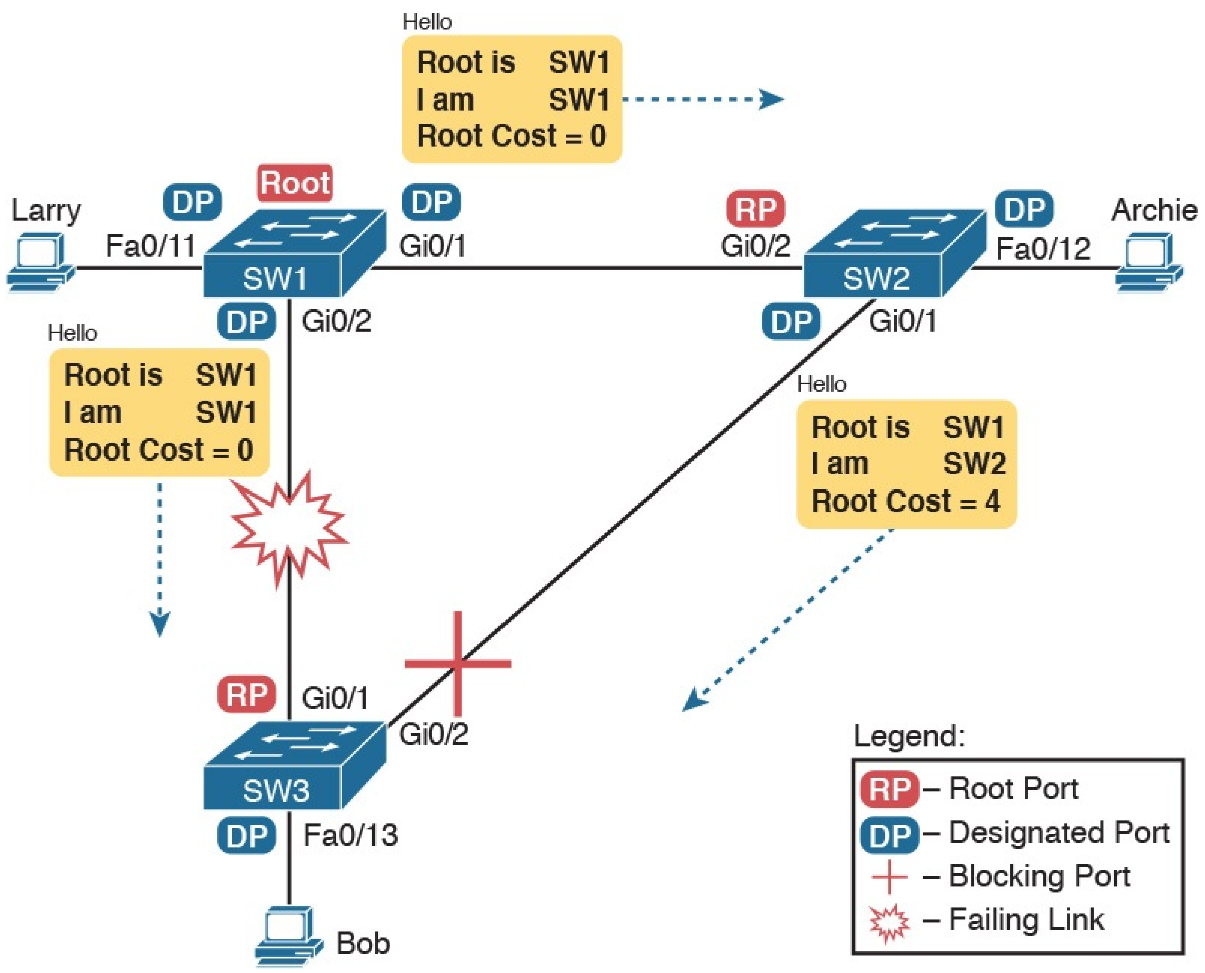


Figure 2-7 *Initial STP State Before SW1-SW3 Link Fails*

Now that SW3 can act, it begins by reevaluating the choice of root switch. SW3 still receives the Hellos from SW2, as forwarded from the root (SW1). SW1 still has a lower BID than SW3; otherwise, SW1 would not have already been the root. So, SW3 decides that SW1 is still the best switch and that SW3 is not the root.

Next, SW3 reevaluates its choice of RP. At this point, SW3 is receiving Hellos on only one interface: Gi0/2. Whatever the calculated root cost, Gi0/2 becomes SW3's new RP. (The cost would be 8, assuming the STP costs had no changes since Figures 2-5 and 2-6.)

SW3 then reevaluates its role as DP on any other interfaces. In this example, no real work needs to be done. SW3 was already DP on interface Fa0/13, and it continues to be the DP because no other switches connect to that port.

Changing Interface States with STP

STP uses the idea of roles and states. *Roles*, like root port and designated port, relate to how STP analyzes the LAN topology. *States*, like forwarding and blocking, tell a switch whether to send or receive frames. When STP converges, a switch chooses new port roles, and the port roles determine the state (forwarding or blocking).

Switches can simply move immediately from forwarding to blocking state, but they must take extra time to transition from blocking state to forwarding state. For instance, when a switch formerly used port G0/1 as its RP (a role), that port was in a forwarding state. After convergence, G0/1 might be neither an RP nor DP; the switch can immediately move that port to a blocking state.

When a port that formerly blocked needs to transition to forwarding, the switch first puts the port through two intermediate interface states. These temporary states help prevent temporary loops:

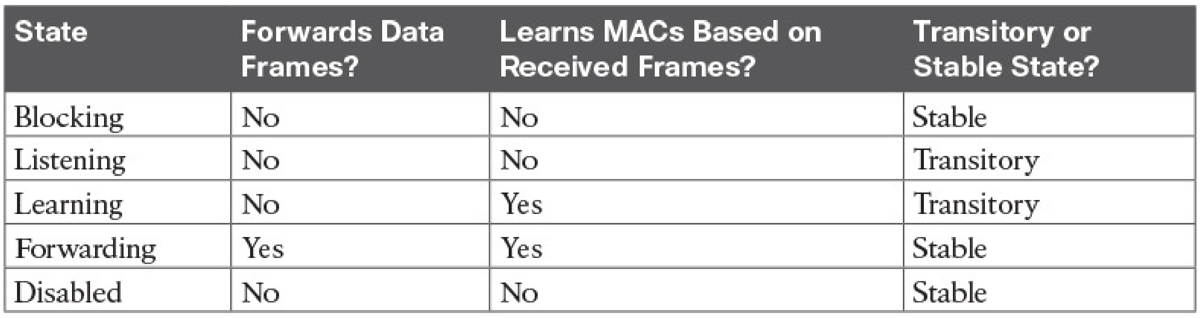
Key Topic

- **Listening:** Like the blocking state, the interface does not forward frames. The switch removes old stale (unused) MAC table entries for which no frames are received from each MAC address during this period. These stale MAC table entries could be the cause of the temporary loops.
- **Learning:** Interfaces in this state still do not forward frames, but the switch begins to learn the MAC addresses of frames received on the interface.

STP moves an interface from blocking to listening, then to learning, and then to forwarding state. STP leaves the interface in each interim state for a time equal to the forward delay timer, which defaults to 15 seconds. As a result, a convergence event that causes an interface to change from blocking to forwarding requires 30 seconds to transition from blocking to forwarding. In addition, a switch might have to wait MaxAge seconds before even choosing to move an interface from blocking to forwarding state.

For example, follow what happens with an initial STP topology as shown in Figures 2-3 through 2-6, with the SW1-to-SW3 link failing as shown in Figure 2-7. If SW1 simply quit sending Hello messages to SW3, but the link between the two did not fail, SW3 would wait MaxAge seconds before reacting (20 seconds is the default). SW3 would actually quickly choose its ports' STP roles, but then wait 15 seconds each in listening and learning states on interface Gi0/2, resulting in a 50-second convergence delay.

Table 2-8 summarizes spanning tree's various interface states for easier review.



State	Forwards Data Frames?	Learns MACs Based on Received Frames?	Transitory or Stable State?
Blocking	No	No	Stable
Listening	No	No	Transitory
Learning	No	Yes	Transitory
Forwarding	Yes	Yes	Stable
Disabled	No	No	Stable

Table 2-8 IEEE 802.1D Spanning-Tree States

Rapid STP (IEEE 802.1w) Concepts

The original STP worked well given the assumptions about networks and networking devices in that era. However, as with any computing or networking standard, as time passes, hardware and software capabilities improve, so new protocols emerge to take advantage of those new capabilities. For STP, one of the most significant improvements over time has been the introduction of Rapid Spanning Tree Protocol (RSTP), introduced as standard IEEE 802.1w.

Before getting into the details of RSTP, it helps to make sense of the standards numbers a bit. 802.1w was actually an amendment to the 802.1D standard. 802.1D was published anew in 1998 (and a few times before that). After the 1998 version of 802.1D, the IEEE published the 802.1w amendment in 2001. Later, when the IEEE 802.1 committee updated the 802.1D standard in 2004, the IEEE pulled the 802.1w amendment details into the 802.1D-2004 standard.

So, why do we care? Sometimes people use the term *STP* to refer to the original pre-RSTP rules for STP. Some use STP to mean anything in the 802.1D standard, which now includes RSTP. So for real life, make sure you know what people mean when they use STP: do they mean STP to include RSTP concepts, or not? For this book, throughout the book, if the distinction between STP and RSTP matters, the book will use STP for the original STP rules and RSTP for the new ones introduced by 802.1w.

Note

The IEEE sells its standards, but through the “Get IEEE 802” program, you can get free PDFs of the current 802 standards. To read about RSTP 802.1w, you will need to download the 802.1D standard, and then look for the sections about RSTP.

Now on to the details about RSTP in this chapter. There are similarities between RSTP and STP, so this section next compares and contrasts the two. Following that, the rest of this section discusses the concepts unique to RSTP that are not found in STP: alternate root ports, different port states, backup ports, and the port roles used by RSTP.

Comparing STP and RSTP

RSTP (802.1w) works just like STP (the original 802.1D) in several ways:

Key Topic

- It elects the root switch using the same parameters and tiebreakers.
- It elects the root port on nonroot switches with the same rules.
- It elects designated ports on each LAN segment with the same rules.
- It places each port in either forwarding or blocking state, although RSTP calls the blocking state the discarding state.

In fact, RSTP works so much like STP that they can both be used in the same network. RSTP and STP switches can be deployed in the same network, with RSTP features working in switches that support it, and traditional 802.1D STP features working in the switches that support only STP.

With all these similarities, you might be wondering why the IEEE bothered to create RSTP in the first place. The overriding reason is convergence. STP takes a relatively long time to converge (50 seconds with the default settings when all the wait times must be followed). RSTP improves network convergence when topology changes occur, usually converging within a few seconds (or in slow conditions, in about 10 seconds).

IEEE 802.1w RSTP changes and adds to IEEE 802.1D STP in ways that avoid waiting on STP timers, resulting in quick transitions from forwarding to blocking state and vice versa. Specifically, RSTP, compared to STP, defines more cases in which the switch can avoid waiting for a timer to expire, such as the following:

- Adds a new mechanism to replace the root port, without any waiting to reach a forwarding state (in some conditions)
- Adds a new mechanism to replace a designated port, without any waiting to reach a forwarding state (in some conditions)
- Lowers waiting times for cases in which RSTP must wait

For instance, when a link remains up, but Hello BPDUs simply stop arriving regularly on a port, STP requires a switch to wait for MaxAge seconds. STP defines the MaxAge timers based on ten times the Hello timer, or 20 seconds, by default. RSTP shortens this timer, defining MaxAge as three times the Hello timer.

The best way to get a sense for these mechanisms is to see how the RSTP alternate port and the backup port both work. RSTP uses the term *alternate port* to refer to a switch's other ports that could be used as root port if the root port ever fails. The *backup port* concept provides a backup port on the local switch for a designated port, but only applies to some topologies that frankly do not happen often with a modern network design. However, both are instructive about how RSTP works. Table 2-9 lists these RSTP port roles.

Key Topic

Function	Port Role
Nonroot switch's best path to the root	Root port
Replaces the root port when the root port fails	Alternate port
Switch port designated to forward onto a collision domain	Designated port
Replaces a designated port when a designated port fails	Backup port
Port that is administratively disabled	Disabled port

Table 2-9 Port Roles in 802.1w RSTP

RSTP and the Alternate (Root) Port Role

With STP, each nonroot switch places one port in the STP root port (RP) role. RSTP follows that same convention, with the same exact rules for choosing the RP. RSTP then takes another step, naming other possible RPs, identifying them as *alternate ports*.

To be an alternate port, both the RP and the alternate port must receive Hellos that identify the same root switch. For instance, in Figure 2-8, SW1 is the root. SW3 will receive Hello BPDUs on two ports: G0/1 and G0/2. Both Hellos list SW1's bridge ID (BID) as the root switch, so whichever port is not the root port meets the criteria to be an alternate port. SW3 picks G0/1 as its root port in this case, and then makes G0/2 an alternate port.

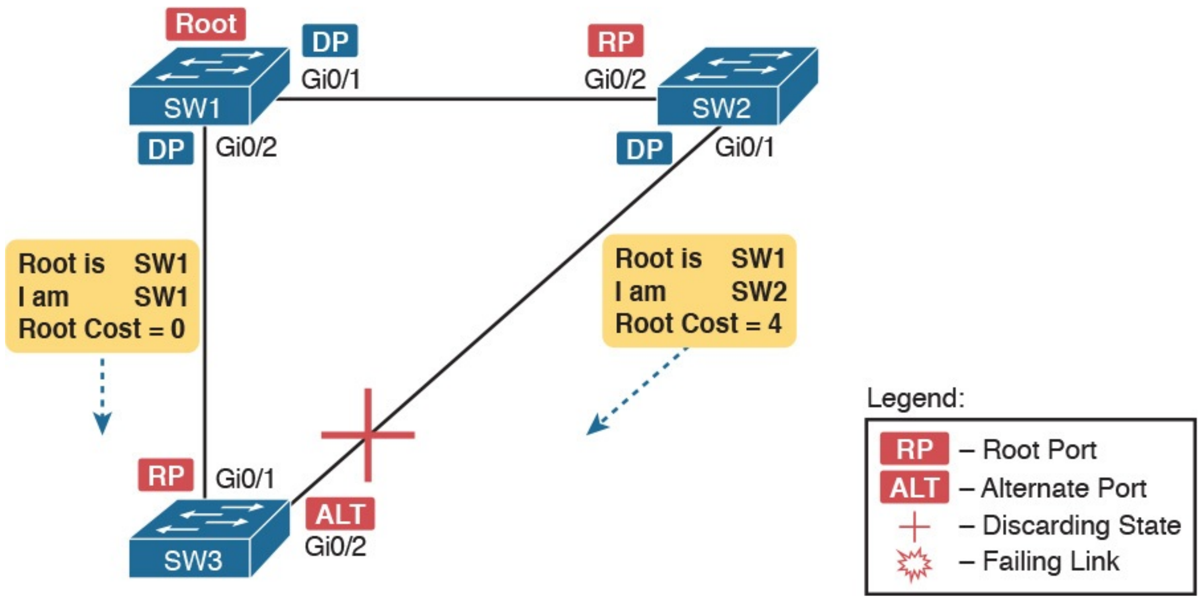


Figure 2-8 *Example of SW3 Making G0/2 Become an Alternate Port*

An alternate port basically works like the second-best option for root port. The alternate port can take over for the former root port, often very rapidly, without requiring a wait in other interim RSTP states. For instance, when the root port fails, or when Hellos stop arriving on the original root port, the switch changes the former root port's role and state: (a) the role from root port to a disabled port, and (b) the state from forwarding to discarding (the equivalent of STP's blocking state). Then, without waiting on any timers, the switch changes roles and state for the alternate port: its role changes to be the root port, with a forwarding state.

Notably, the new root port also does not need to spend time in other states, such as learning state, instead moving immediately to forwarding state.

Figure 2-9 shows an example of RSTP convergence. SW3's root port before the failure shown in this figure is SW3's G0/1, the link connected directly to SW1 (the root switch). Then SW3's link to SW1 fails as shown in Step 1 of the figure.

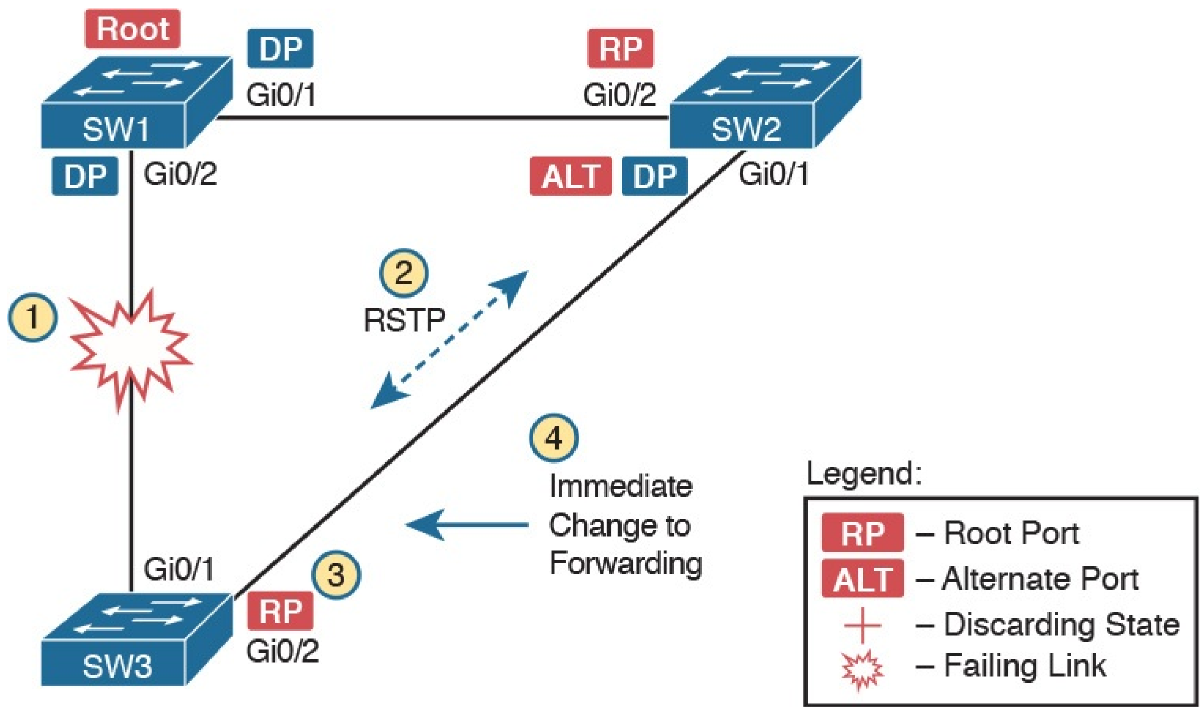


Figure 2-9 *Convergence Events with SW3 G0/1 Failure*

Following the steps in Figure 2-9:

Step 1. The link between SW1 and SW3 fails, so that SW3's current root port (Gi0/1) fails.

Step 2. SW3 and SW2 exchange RSTP messages to confirm that SW3 will now transition its former alternate port (Gi0/2) to be the root port. This action causes SW2 to flush the required MAC table entries.

Step 3. SW3 transitions G0/1 to the disabled role and G0/2 to the root port role.

Step 4. SW3 transitions G0/2 to a forwarding state immediately, without using learning state, because this is one case in which RSTP knows the transition will not create a loop.

As soon as SW3 realizes its G0/1 interface has failed, the process shown in the figure takes very little time. None of the processes rely on timers, so as soon as the work can be done, the convergence completes. (This particular convergence example takes about 1 second in a lab.)

RSTP States and Processes

The depth of the previous example does not point out all details of RSTP, of course; however, the example does show enough details to discuss RSTP states and internal processes.

Both STP and RSTP use *port states*, but with some differences. First, RSTP keeps both the learning and forwarding states as compared with STP, for the same purposes. However, RSTP does not even define a listening state, finding it unnecessary. Finally, RSTP renames the blocking state to the discarding state, and redefines its use slightly.

RSTP uses the discarding state for what 802.1D defines as two states: disabled state and blocking state. Blocking should be somewhat obvious by now: The interface can work physically, but STP/RSTP chooses to not forward traffic to avoid loops. STP's disabled state simply meant that the interface was administratively disabled. RSTP just combines those into a single discarding state. Table 2-10 shows the list of STP and RSTP states for comparison purposes.

Function	802.1D State	802.1w State
Port is administratively disabled	Disabled	Discarding
Stable state that ignores incoming data frames and is not used to forward data frames	Blocking	Discarding
Interim state without MAC learning and without forwarding	Listening	Not used
Interim state with MAC learning and without forwarding	Learning	Learning
Stable state that allows MAC learning and forwarding of data frames	Forwarding	Forwarding

Table 2-10 Port States Compared: 802.1D STP and 802.1w RSTP

RSTP also changes some processes and message content (compared to STP) to speed convergence. For example, STP waits for a time (forward delay) in both listening and learning states. The reason for this delay in STP is that, at the same time, the switches have all been told to time out their MAC table entries. When the topology changes, the existing MAC table entries may actually cause a loop. With STP, the switches all tell each other (with BPDU messages) that the topology has changed, and to time out any MAC table entries using the forward delay timer. This removes the entries, which is good, but it causes the need to wait in both listening and learning state for forward delay time (default 15 seconds each).

RSTP, to converge more quickly, avoids relying on timers. RSTP switches tell each other (using messages) that the topology has changed. Those messages also direct neighboring switches to flush the contents of their MAC tables in a way that removes all the potentially loop-causing entries, without a wait. As a result, RSTP creates more scenarios in which a formerly discarding port can immediately transition to a forwarding state, without waiting, and without using the learning state, as shown in the example in Figure 2-9.

RSTP and the Backup (Designated) Port Role

The RSTP backup port role acts as yet another new RSTP port role as compared to STP. As a reminder, the RSTP alternate port role creates a way for RSTP to quickly replace a switch's root port. Similarly, the RSTP backup port role creates a way for RSTP to quickly replace a switch's designated port on some LAN.

The need for a backup port can be a bit confusing at first, because the need for the backup port role only happens in designs that are a little unlikely today. The reason is that a design must use hubs, which then allows the possibility that one switch connects more than one port to the same collision domain.

Figure 2-10 shows an example. SW3 and SW4 both connect to the same hub. SW4's port F0/1 happens to win the election as designated port (DP). The other port on SW4 that connects to the same collision domain, F0/2, acts as a backup port.

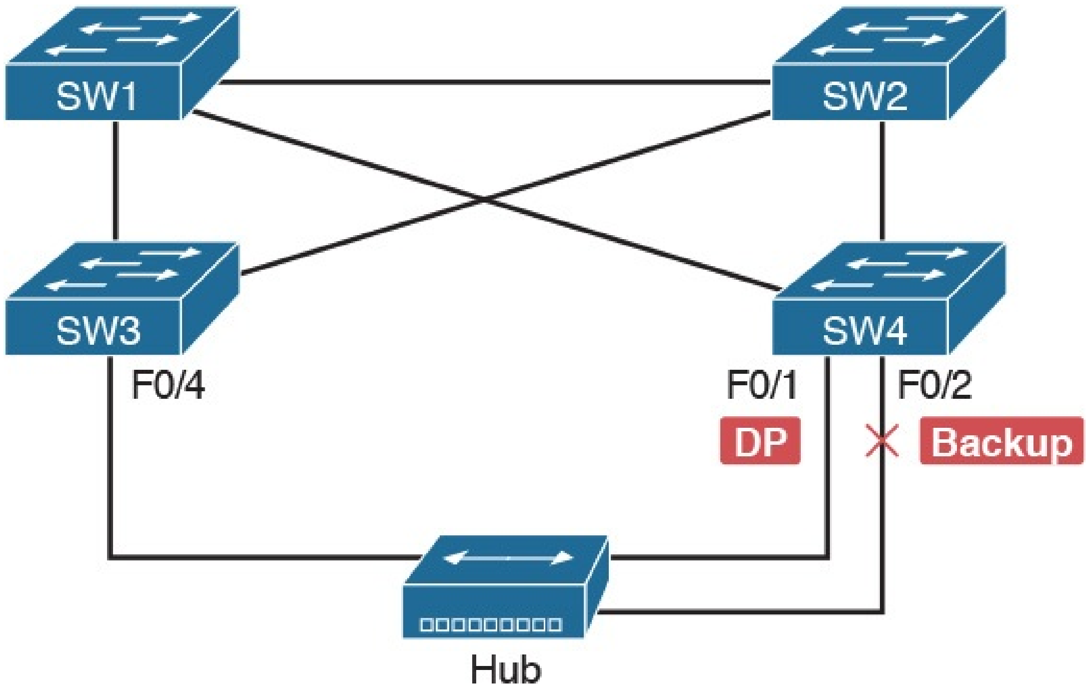


Figure 2-10 *RSTP Backup Port Example*

With a backup port, if the current designated port fails, SW4 can start using the backup port with rapid convergence. For instance, if SW4's F0/1 interface were to fail, SW4 could transition F0/2 to the designated port role, without any delay in moving from discarding state to a forwarding state.

RSTP Port Types

The final RSTP concept included here relates to some terms RSTP uses to refer to different types of ports and the links that connect to those ports.

To begin, consider the basic figure of Figure 2-11. It shows several links between two switches. RSTP considers these links to be point-to-point links and the ports connected to them to be point-to-point ports, because the link connects exactly two devices (points).

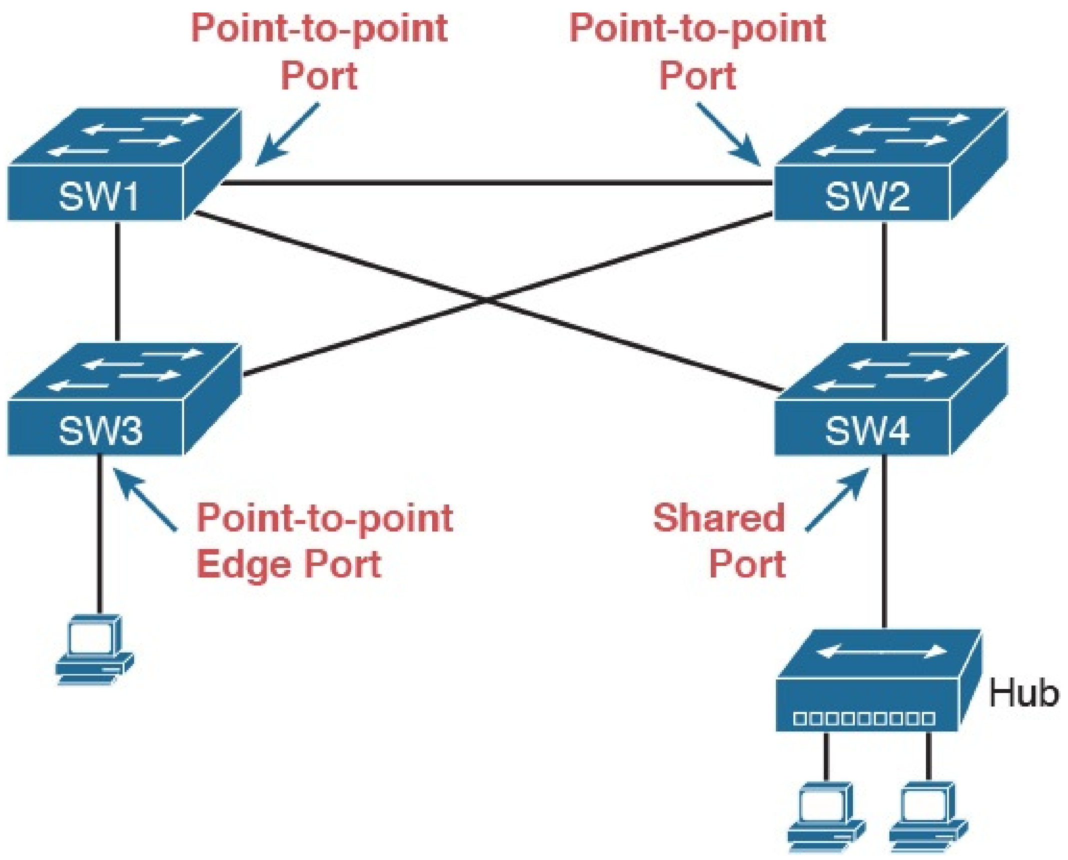


Figure 2-11 *RSTP Link Types*

RSTP further classifies point-to-point ports into two categories. Point-to-point ports that connect two switches are not at the edge of the network and are simply called *point-to-point ports*. Ports that instead connect to a single endpoint device at the edge of the network, like a PC or server, are called *point-to-point edge ports*, or simply *edge ports*. In Figure 2-11, SW3's switch port connected to a PC is an edge port.

Finally, RSTP defines the term *shared* to describe ports connected to a hub. The term *shared* comes from the fact that hubs create a shared Ethernet; hubs also force the attached switch port to use half-duplex logic. RSTP assumes that all half-duplex ports may be connected to hubs, treating ports that use half duplex as shared ports. RSTP converges more slowly on shared ports as compared to all point-to-point ports.

Optional STP Features

At this point, you have learned plenty of details that will be useful to next configure and verify STP operations, as discussed in Chapter 3. However, before moving to that chapter, the final section of the chapter briefly introduces a few related topics that make STP work even better or be more secure: EtherChannel, PortFast, and BPDU Guard.

EtherChannel

One of the best ways to lower STP's convergence time is to avoid convergence altogether. EtherChannel provides a way to prevent STP convergence from being needed when only a single port or cable failure occurs.

EtherChannel combines multiple parallel segments of equal speed (up to eight) between the same pair of switches, bundled into an EtherChannel. The switches treat the EtherChannel as a single interface with regard to STP. As a result, if one of the links fails, but at least one of the links is up, STP convergence does not have to occur. For example, Figure 2-12 shows the familiar three-switch network, but now with two Gigabit Ethernet connections between each pair of switches.

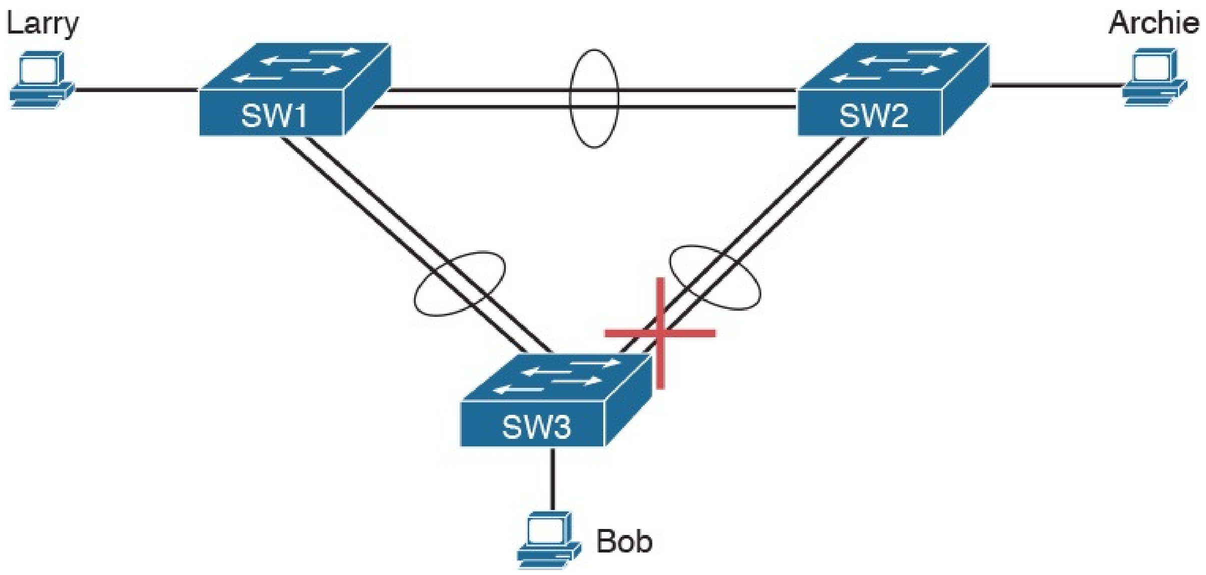


Figure 2-12 *Two-Segment EtherChannels Between Switches*

With each pair of Ethernet links configured as an EtherChannel, STP treats each EtherChannel as a single link. In other words, both links to the same switch must fail for a switch to need to cause STP convergence. Without EtherChannel, if you have multiple parallel links between two switches, STP blocks all the links except one. With EtherChannel, all the parallel links can be up and working at the same time, while reducing the number of times STP must converge, which in turn makes the network more available.

When a switch makes a forwarding decision to send a frame out an EtherChannel, the switch then has to take an extra step in logic: Out which physical interface does it send the frame? The switch has load-balancing logic that lets it pick an interface for each frame, with a goal of spreading the traffic load across all active links in the channel. As a result, a LAN design that uses EtherChannels makes much better use of the available bandwidth between switches, while also reducing the number of times that STP must converge.

Note that EtherChannels may be Layer 2 EtherChannels (as described here) or Layer 3 EtherChannels (as discussed in Chapter 19, “IPv4 Routing in the LAN”). Layer 2 EtherChannels combine links that switches use as switch ports, with the switches using Layer 2 switching logic to forward and receive Ethernet frames over the EtherChannels. Layer 3 EtherChannels also combine links, but the switches use Layer 3 routing logic to forward packets over the EtherChannels. All references to EtherChannel in Part I of this book refer to Layer 2 EtherChannels unless otherwise noted.

PortFast

PortFast allows a switch to immediately transition from blocking to forwarding, bypassing listening and learning states. However, the only ports on which you can safely enable PortFast are ports on which you know that no bridges, switches, or other STP-speaking devices are connected. Otherwise, using PortFast risks creating loops, the very thing that the listening and learning states are intended to avoid.

PortFast is most appropriate for connections to end-user devices. If you turn on PortFast on ports connected to end-user devices, when an end-user PC boots, the switch port can move to an STP forwarding state and forward traffic as soon as the PC NIC is active. Without PortFast, each port must wait while the switch confirms that the port is a DP, and then wait while the interface sits in the temporary listening and learning states before settling into the forwarding state.

PortFast is a popular feature for edge ports; in fact, RSTP incorporates PortFast concepts. You may recall the mention of RSTP port types, particularly point-to-point edge port types, around Figure 2-11. RSTP, by design of the protocol, converges quickly on these point-to-point edge type ports by bypassing the learning state, which is the same idea Cisco originally introduced with PortFast. In practice, Cisco switches enable RSTP point-to-point edge ports by enabling PortFast on the port.

BPDU Guard

STP opens up the LAN to several different types of possible security exposures. For example:

- An attacker could connect a switch to one of these ports, one with a low STP priority value, and become the root switch. The new STP topology could have worse performance than the desired topology.
- The attacker could plug into multiple ports, into multiple switches, become root, and actually forward much of the traffic in the LAN. Without the networking staff realizing it, the attacker could use a LAN

analyzer to copy large numbers of data frames sent through the LAN.

- Users could innocently harm the LAN when they buy and connect an inexpensive consumer LAN switch (one that does not use STP). Such a switch, without any STP function, would not choose to block any ports and could cause a loop.

The *Cisco BPDU Guard* feature helps defeat these kinds of problems by disabling a port if any BPDUs are received on the port. So, this feature is particularly useful on ports that should be used only as an access port and never connected to another switch.

In addition, the BPDU Guard feature helps prevent problems with PortFast. PortFast should be enabled only on access ports that connect to user devices, not to other LAN switches. Using BPDU Guard on these same ports makes sense because if another switch connects to such a port, the local switch can disable the port before a loop is created.

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book, DVD, or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 2-11 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Review Element	Review Date(s)	Resource Used
Review key topics		Book, DVD/website
Review key terms		Book, DVD/website
Answer DIKTA questions		Book, PCPT
Review memory tables		Book, App

Table 2-11 Chapter Review Tracking

Review All the Key Topics

Key Topic

Key Topic Element	Description	Page Number
Table 2-2	Lists the three main problems that occur when not using STP in a LAN with redundant links	47
Table 2-3	Lists the reasons why a switch chooses to place an interface into forwarding or blocking state	49
Table 2-4	Lists the most important fields in Hello BPDU messages	50
List	Logic for the root switch election	52
Figure 2-6	Shows how switches calculate their root cost	53
Table 2-6	Lists the original and current default STP port costs for various interface speeds	55
Step list	A summary description of steady-state STP operations	56
Table 2-7	STP timers	56
List	Definitions of what occurs in the listening and learning states	58
Table 2-8	Summary of 802.1D states	58
List	Key similarities between 802.1D STP and 802.1w RSTP	59
Table 2-9	List of 802.1w port roles	60
Table 2-10	Comparisons of port states with 802.1D and 802.1w	62

Table 2-12 Key Topics for Chapter 2

Key Terms You Should Know

blocking state
BPDU Guard
bridge ID
bridge protocol data unit (BPDU)
designated port
EtherChannel
forward delay
forwarding state
Hello BPDU
IEEE 802.1D
learning state
listening state
MaxAge
PortFast
root port
root switch
root cost
Spanning Tree Protocol (STP)
rapid STP (RSTP)

IEEE 802.1w
alternate port
backup port
disabled port
discarding state

Chapter 3. Spanning Tree Protocol Implementation

This chapter covers the following exam topics:

1.0 LAN Switching Technologies

1.3 Configure, verify, and troubleshoot STP protocols

1.3.a STP mode (PVST+ and RPVST+)

1.3.b STP root bridge selection

1.4 Configure, verify, and troubleshoot STP-related optional features

1.4.a PortFast

1.4.b BPDU guard

1.5 Configure, verify, and troubleshoot (Layer 2/Layer 3) EtherChannel

1.5.a Static

1.5.b PAGP

1.5.c LACP

Cisco IOS–based LAN switches enable Spanning Tree Protocol (STP) by default on all interfaces in every VLAN. However, network engineers who work with medium-size to large-size Ethernet LANs usually want to configure at least some STP settings. First and foremost, Cisco IOS switches traditionally default to use STP rather than Rapid STP (RSTP), and the simple upgrade to RSTP improves convergence. For most LANs with more than a few switches, the network engineer will likely want to influence the choices made by STP, whether using traditional STP or RSTP—choices such as which switch becomes root, with predictability about which switch ports will block/discard when all ports are physically working. The configuration can also be set so that when links or switches fail, the engineer can predict the STP topology in those cases, as well.

This chapter discusses configuration and verification of STP. The first major section weaves a story of how to change different settings, per VLAN, with the **show** commands that reveal the current STP status affected by each configuration command. Those settings impact both STP and RSTP, but the examples use switches that use traditional 802.1D STP rather than RSTP. The second major section shows how to configure the same optional STP features mentioned in Chapter 2: PortFast, BPDU Guard, and EtherChannel (specifically Layer 2 EtherChannel). The final major section of this chapter looks at the simple (one command) configuration to enable RSTP, and the differences and similarities in **show** command output that occur when using RSTP versus STP.

“Do I Know This Already?” Quiz

Take the quiz (either here, or use the PCPT software) if you want to use the score to help you decide how much time to spend on this chapter. The answers are at the bottom of the page following the quiz, and the explanations are in DVD Appendix C and in the PCPT software.

Foundation Topics Section	Questions
Implementing STP	1–3
Implementing Optional STP Features	4
Implementing RSTP	5, 6

Table 3-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

1. On a 2960 switch, which of the following commands change the value of the bridge ID? (Choose two answers.)

   a. spanning-tree bridge-id *value*

   b. spanning-tree vlan *vlan-number* **root** {**primary** | **secondary**}

   c. spanning-tree vlan *vlan-number* **priority** *value*

   d. set spanning-tree priority *value*

2. Examine the following extract from the **show spanning-tree** command on a Cisco switch:

Click here to view code image

```
Bridge ID  Priority    32771  (priority 32768 sys-id-
ext 3)
           Address     0019.e86a.6f80
```

   Which of the following answers is true about the switch on which this command output was gathered?

   a. The information is about the STP instance for VLAN 1.

   b. The information is about the STP instance for VLAN 3.

   c. The command output confirms that this switch cannot possibly be the root switch.

   d. The command output confirms that this switch is currently the root switch.

3. A switch’s G0/1 interface, a trunk that supports VLANs 1–10, has autonegotiated a speed of 100 Mbps. The switch currently has all default settings for STP. Which of the following actions results in the

switch using an STP cost of 19 for that interface in VLAN 3? (Choose two answers.)

a. spanning-tree cost 19

b. spanning-tree port-cost 19

c. spanning-tree vlan 3 port-cost 19

d. Adding no configuration

4. An engineer configures a switch to put interfaces G0/1 and G0/2 into the same Layer 2 EtherChannel. Which of the following terms is used in the configuration commands?

a. EtherChannel

b. PortChannel

c. Ethernet-Channel

d. Channel-group

5. Examine the following first seven lines of output from the **show spanning-tree** command on a Cisco switch:

Click here to view code image

```
SW1# show spanning-tree vlan 5

VLAN0005
  Spanning tree enabled protocol rstp
  Root ID     Priority    32773
              Address     1833.9d7b.0e80
              Cost        15
              Port        25 (GigabitEthernet0/1)
              Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
```

Which of the following answers is true about the switch on which this command output was gathered? (Choose two answers.)

a. The root switch's MAC address is 1833.9d7b.0e80 and the local switch is the root.

b. The local switch's MAC address is 1833.9d7b.0e80 and it is not the root.

c. This switch uses STP and not RSTP.

d. This switch uses RSTP.

6. The following output shows the last lines of output of a **show spanning-tree** command extracted from a Cisco switch running IOS:

Click here to view code image

```
SW1# show spanning-tree vlan 10
! lines omitted

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----
------------------
Fa0/1               Desg FWD 100       128.1    P2p
Edge
Fa0/2               Desg FWD 19        128.2    Shr
Gi0/1               Desg FWD 4         128.25   P2p
Gi0/2               Root FWD 4         128.26   P2p
```

The answers all mention an interface and the state listed in the Type column of the output, along with a reason why that port should be listed as that type of STP port. Which answers list what could be a correct reason for the interface to be listed as that type of STP port? (Choose two answers.)

a. Fa0/1 is P2p Edge because of the **spanning-tree rstp edge** interface subcommand.

b. Fa0/2 is Shr because Fa0/2 uses half duplex.

c. Gi0/1 is P2p because it is a VLAN trunk.

d. Gi0/2 is P2p because the switch had no reason to make it Shr or P2p Edge.

Answers to the "Do I Know This Already?" quiz:

1 B, C **2** B **3** A, D **4** D **5** A, D **6** B, D

Foundation Topics

Implementing STP

Cisco IOS switches usually use STP (IEEE 802.1D) by default rather than RSTP, and with effective default settings. You can buy some Cisco switches and connect them with Ethernet cables in a redundant topology, and STP will ensure that frames do not loop. And you never even have to think about changing any settings!

Although STP works without any configuration, most medium-size to large-size campus LANs benefit from some STP configuration. With all defaults, the switches choose the root based on the lowest burned-in MAC address on the switches because they all default to use the same STP priority. As a better option, configure the switches so that the root is predictable.

For instance, Figure 3-1 shows a typical LAN design model, with two distribution layer switches (D1 and D2). The design may have dozens of access layer switches that connect to end users; the figure shows just three access switches (A1, A2, and A3). For a variety of reasons, most network engineers make the distribution layer switches be the root. For instance, the configuration could make D1 be the root by having a lower priority, with D2 configured with the next lower priority, so it becomes root if D1 fails.

Key Topic

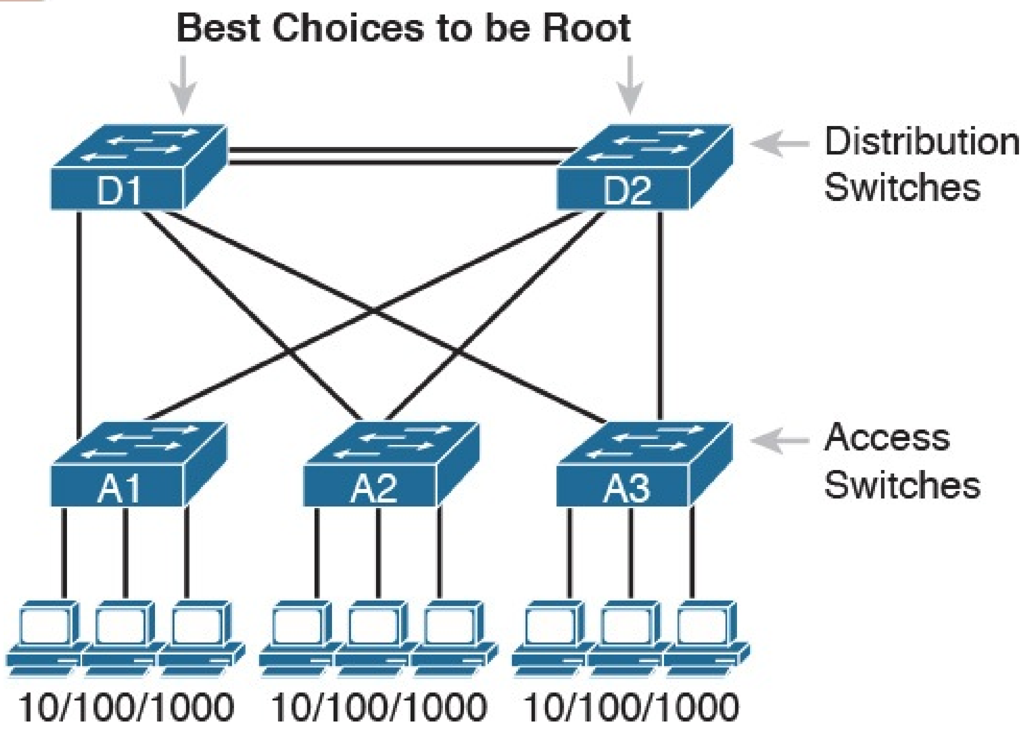


Figure 3-1 *Typical Configuration Choice: Making Distribution Switch Be Root*

This first section of the chapter examines a variety of topics that somehow relate to STP configuration. It begins with a look at STP configuration options, as a way to link the concepts of Chapter 2 to the configuration choices in this chapter. Following that, this section introduces some **show** commands for the purpose of verifying the default STP settings before changing any configuration.

Setting the STP Mode

Chapter 2 described how 802.1D STP works in one VLAN. Now that this chapter turns our attention to STP configuration in Cisco switches, one of the first questions is this: Which kind of STP do you intend to use in a LAN? And to answer that question, you need to know a little more background.

The IEEE first standardized STP as the IEEE 802.1D standard, first published back in 1990. To put some perspective on that date, Cisco sold no LAN switches at the time, and virtual LANs did not exist yet. Instead of multiple VLANs in a LAN, there was just one broadcast domain, and one instance of STP. However, the addition of VLANs and the introduction of LAN switches into the market have created a need to add to and extend STP.

Today, Cisco IOS–based LAN switches allow you to use one of three STP configuration modes that reflect that history. The first two sections of this chapter use the mode called Per-VLAN Spanning Tree Plus (PVST+, or sometimes PVSTP), a Cisco-proprietary improvement of 802.1D STP. The *per-VLAN* part of the name gives away the main feature: PVST+ creates a different STP topology per VLAN, whereas 802.1D actually did not. PVST+ also introduced PortFast. Cisco switches often use PVST+ as the default STP mode per a default global command of **spanning-tree mode pvst**.

Over time, Cisco added RSTP support as well, with two STP modes that happen to use RSTP. One mode basically takes PVST+ and upgrades it to use RSTP logic as well, with a mode called *Rapid PVST+*, enabled with the global command **spanning-tree mode rapid-pvst**. Cisco IOS–based switches support a third mode, called Multiple Spanning Tree (MST) (or Multiple Instance of Spanning Tree), enabled with the **spanning-tree mode mst** command. (This book does not discuss MST beyond this brief mention; the CCNP Switch exam typically includes MST details.)

Connecting STP Concepts to STP Configuration Options

If you think back to the details of STP operation in Chapter 2, STP uses two types of numbers for most of its decisions: the BID and STP port costs. Focusing on those two types of numbers, consider this summary of what STP does behind the scenes:

- Uses the BID to elect the root switch, electing the switch with the numerically lowest BID
- Uses the total STP cost in each path to the root, when each nonroot switch chooses its own root port (RP)
- Uses each switch’s root cost, which is in turn based on STP port costs, when switches decide which switch port becomes the designated port (DP) on each LAN segment

Unsurprisingly, Cisco switches let you configure part of a switch’s BID and the STP port cost, which in turn influences the choices each switch makes with STP.

Per-VLAN Configuration Settings

Beyond supporting the configuration of the BID and STP port costs, Cisco switches support configuring both settings per VLAN. By default, Cisco switches use IEEE 802.1D, not RSTP (802.1w), with a Cisco-proprietary feature called Per-VLAN Spanning Tree Plus (PVST+). PVST+ (often abbreviated as simply PVST today) creates a different instance of STP for each VLAN. So, before looking at the tunable STP parameters, you need to have a basic understanding of PVST+, because the configuration settings can differ for each instance of STP.

PVST+ gives engineers a load-balancing tool with STP. By changing some STP configuration parameters differently for different VLANs, the engineer could cause switches to pick different RPs and DPs in different VLANs. As a result, some traffic in some VLANs can be forwarded over one trunk, and traffic for other VLANs can be forwarded over a different trunk.

Figure 3-2 shows the basic idea, with SW3 forwarding odd-numbered VLAN traffic over the left trunk (Gi0/1) and even-numbered VLANs over the right trunk (Gi0/2).

Key Topic

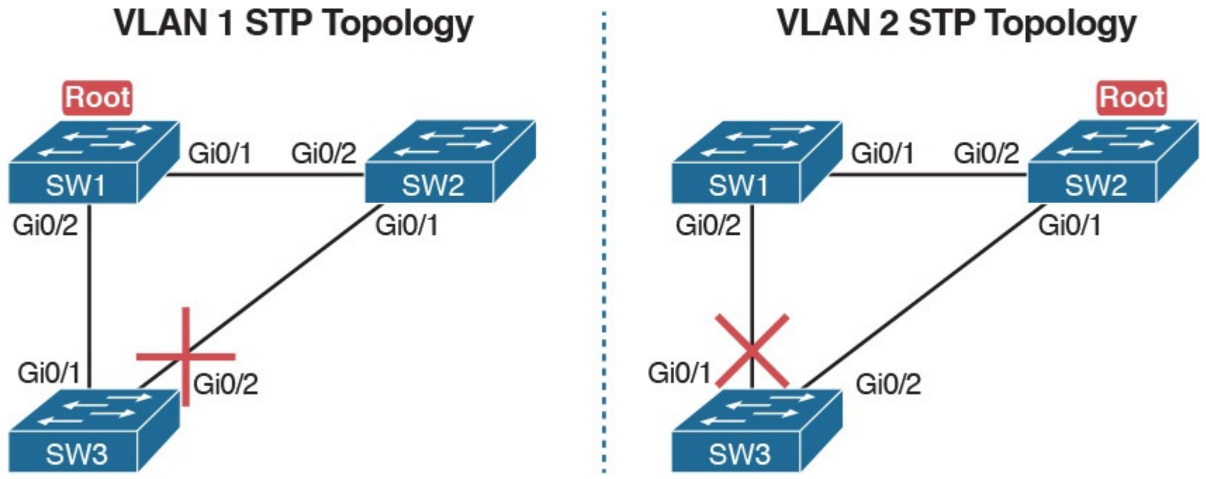


Figure 3-2 *Load Balancing with PVST+*

The next few pages look specifically at how to change the BID and STP port cost settings, per VLAN, when using the default PVST+ mode.

The Bridge ID and System ID Extension

Originally, a switch's BID was formed by combining the switch's 2-byte priority and its 6-byte MAC address. Later, the IEEE changed the rules, splitting the original priority field into two separate fields, as shown in Figure 3-3: a 4-bit priority field and a 12-bit subfield called the *system ID extension* (which represents the VLAN ID).

Key Topic

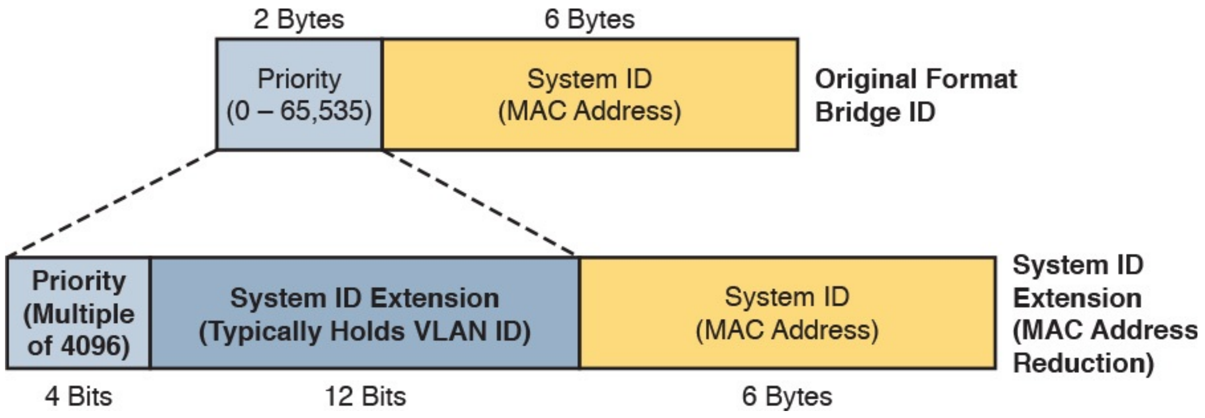

2 Bytes
6 Bytes
Priority (0 – 65,535)
System ID (MAC Address)
Original Format Bridge ID
Priority (Multiple of 4096)
System ID Extension (Typically Holds VLAN ID)
System ID (MAC Address)
System ID Extension (MAC Address Reduction)
4 Bits
12 Bits
6 Bytes

Figure 3-3 *STP System ID Extension*

Cisco switches let you configure the BID, but only the priority part. The switch fills in its universal (burned-in) MAC address as the system ID. It also plugs in the VLAN ID of a VLAN in the 12-bit system ID extension field. The only part configurable by the network engineer is the 4-bit priority field.

Configuring the number to put in the priority field, however, is one of the strangest things to configure on a Cisco router or switch. As shown at the top of Figure 3-3, the priority field was originally a 16-bit number, which represented a decimal number from 0 to 65,535. Because of that history, the current configuration command (**spanning-tree vlan** *vlan-id* **priority** *x*) requires a decimal number between 0 and 65,535. But not just any number in that range will suffice—it must be a multiple of 4096: 0, 4096, 8192, 12288, and so on, up through 61,440.

The switch still sets the first 4 bits of the BID based on the configured value. As it turns out, of the 16 allowed multiples of 4096, from 0 through 61,440, each has a different binary value in their first 4 bits: 0000, 0001, 0010, and so on, up through 1111. The switch sets the true 4-bit priority based on the first 4 bits of the configured value.

Although the history and configuration might make the BID priority idea seem a bit convoluted, having an extra 12-bit field in the BID works well in practice because it can be used to identify the VLAN ID. VLAN IDs range from 1 to 4094, requiring 12 bits. Cisco switches place the VLAN ID into the system ID extension field, so each switch has a unique BID per VLAN.

For example, a switch configured with VLANs 1 through 4, with a default base priority of 32,768, has a default STP priority of 32,769 in VLAN 1, 32,770 in VLAN 2, 32,771 in VLAN 3, and so on. So, you can view the 16-bit priority as a base priority (as configured in the **spanning-tree vlan** *vlan-id* **priority** *x* command) plus the VLAN ID.

Note

Cisco switches must use the system ID extension version of the bridge ID; it cannot be disabled.

Per-VLAN Port Costs

Each switch interface defaults its per-VLAN STP cost based on the IEEE recommendations listed in Table 2-6 in Chapter 2. On interfaces that support multiple speeds, Cisco switches base the cost on the current actual speed. So, if an interface negotiates to use a lower speed, the default STP cost reflects that lower speed. If the interface negotiates to use a different speed, the switch dynamically changes the STP port cost as well.

Alternatively, you can configure a switch's STP port cost with the **spanning-tree** [**vlan** *vlan-id*] **cost** *cost* interface subcommand. You see this command most often on trunks because setting the cost on trunks has an impact on the switch's root cost, whereas setting STP costs on access ports does not.

For the command itself, it can include the VLAN ID, or not. The command only needs a **vlan** parameter on trunk ports to set the cost per VLAN. On a trunk, if the command omits the VLAN parameter, it sets the STP cost for all VLANs whose cost is not set by a **spanning-tree vlan** *x* **cost** command for that VLAN.

STP Configuration Option Summary

Table 3-2 summarizes the default settings for both the BID and the port costs and lists the optional configuration commands covered in this chapter.

Key Topic

Setting	Default	Command(s) to Change Default
BID priority	Base: 32,768	spanning-tree vlan *vlan-id* root {primary | secondary} spanning-tree vlan *vlan-id* priority *priority*
Interface cost	100 for 10 Mbps 19 for 100 Mbps 4 for 1 Gbps 2 for 10 Gbps	spanning-tree vlan *vlan-id* cost *cost*
PortFast	Not enabled	spanning-tree portfast
BPDU Guard	Not enabled	spanning-tree bpduguard enable

Table 3-2 STP Defaults and Configuration Options

Next, the configuration section shows how to examine the operation of STP in a simple network, along with how to change these optional settings.

Verifying STP Operation

Before taking a look at how to change the configuration, first consider a few STP verification commands. Looking at these commands first will help reinforce the default STP settings. In particular, the examples in this section use the network shown in Figure 3-4.

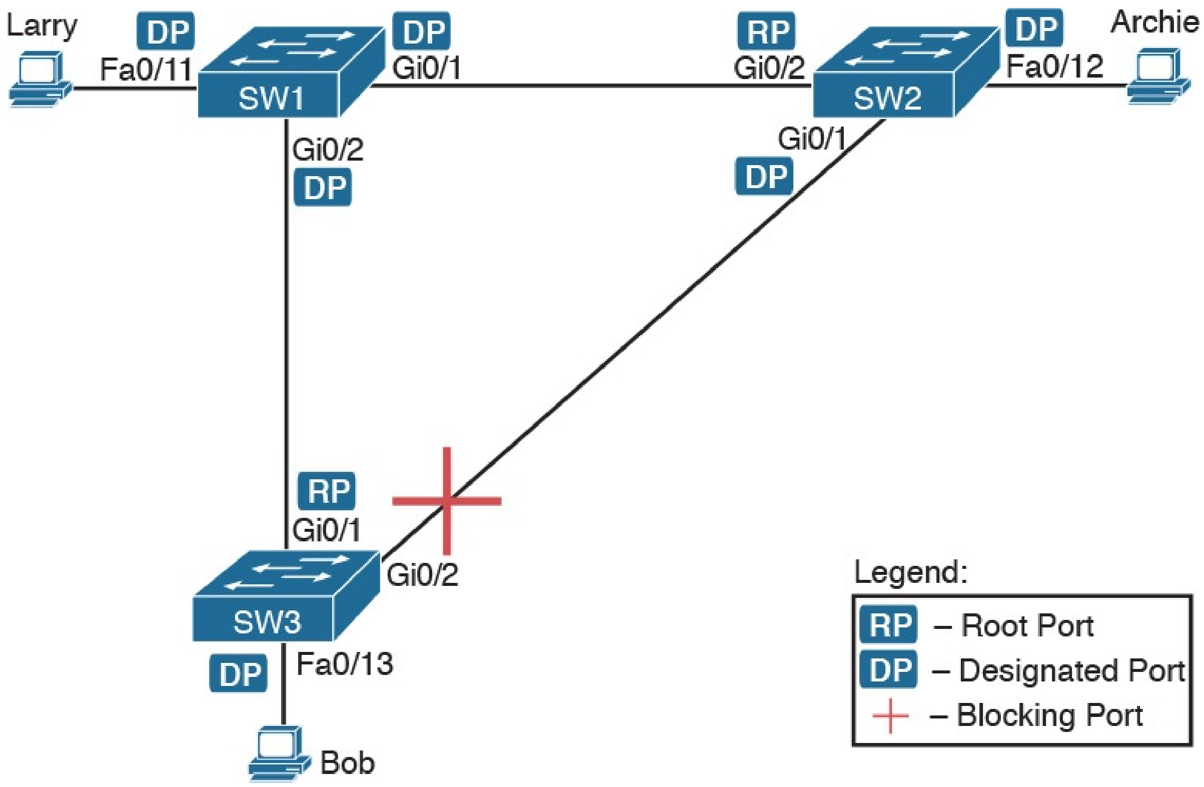


Figure 3-4 *Sample LAN for STP Configuration and Verification Examples*

Example 3-1 begins the discussion with a useful command for STP: the **show spanning-tree vlan 10** command. This command identifies the root switch and lists settings on the local switch. Example 3-1 lists the output of this command on both SW1 and SW2, as explained following the example.

Example 3-1 *STP Status with Default STP Parameters on SW1 and SW2*

Click here to view code image

```
SW1# show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
             Address     1833.9d7b.0e80
             This bridge is the root
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-
ext 10)
                Address     1833.9d7b.0e80
                Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
                Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------
-------------------------
Fa0/11              Desg FWD 19        128.11   P2p
Edge
Gi0/1               Desg FWD 4         128.25   P2p
Gi0/2               Desg FWD 4         128.26   P2p
```

```
SW2# show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
             Address     1833.9d7b.0e80
             Cost        4
             Port        26 (GigabitEthernet0/2)
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-
ext 10)
             Address     1833.9d7b.1380
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------
-------------------------
Fa0/12              Desg FWD 19        128.12   P2p
Gi0/1               Desg FWD 4         128.25   P2p
Gi0/2               Root FWD 4         128.26   P2p
```

Example 3-1 begins with the output of the **show spanning-tree vlan 10** command on SW1. This command first lists three major groups of messages: one group of messages about the root switch, followed by another group about the local switch, and ending with interface role and status information. In this case, SW1 lists its own BID as the root, with even a specific statement that “This bridge is the root,” confirming that SW1 is now the root of the VLAN 10 STP topology.

Next, compare the highlighted lines of the same command on SW2 in the lower half of the example. SW2 lists SW1's BID details as the root; in other words, SW2 agrees that SW1 has won the root election. SW2 does not list the phrase "This bridge is the root." SW2 then lists its own (different) BID details in the lines after the details about the root's BID.

The output also confirms a few default values. First, each switch lists the priority part of the BID as a separate number: 32778. This value comes from the default priority of 32768, plus VLAN 10, for a total of 32778. The output also shows the interface cost for some Fast Ethernet and Gigabit Ethernet interfaces, defaulting to 19 and 4, respectively.

Finally, the bottom of the output from the **show spanning-tree** command lists each interface in the VLAN, including trunks, with the STP port role and port state listed. For instance, on switch SW1, the output lists three interfaces, with a role of Desg for designated port (DP) and a state of FWD for forwarding. SW2 lists three interfaces, two DPs, and one root port, so all three are in an FWD or forwarding state.

Example 3-1 shows a lot of good STP information, but two other commands, shown in Example 3-2, work better for listing BID information in a shorter form. The first, **show spanning-tree root**, lists the root's BID for each VLAN. This command also lists other details, like the local switch's root cost and root port. The other command, **show spanning-tree vlan 10 bridge**, breaks out the BID into its component parts. In this example, it shows SW2's priority as the default of 32768, the VLAN ID of 10, and the MAC address.

Example 3-2 *Listing Root Switch and Local Switch BIDs on Switch SW2*

Click here to view code image

```
SW2# show spanning-tree root

                                        Root    Hello
Max Fwd
Vlan                   Root
ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- -
-- ---  ------------
VLAN0001         32769
1833.9d5d.c900        23     2    20   15   Gi0/1
VLAN0010         32778
1833.9d7b.0e80         4     2    20   15   Gi0/2
VLAN0020         32788 1833.9d7b.0e80
        4     2    20   15   Gi0/2
VLAN0030         32798
```

```
1833.9d7b.0e80          4    2   20  15  Gi0/2
VLAN0040          32808
1833.9d7b.0e80          4    2   20  15  Gi0/2

SW2# show spanning-tree vlan 10 bridge

                                                   Hellc
Vlan                          Bridge
ID                Time  Age  Dly  Protocol
---------------- --------------------------------- ----
-  ---  ---  --------
VLAN0010         32778 (32768,  10)
1833.9d7b.1380    2    20   15  ieee
```

Note that both the commands in Example 3-2 have a VLAN option: **show spanning-tree** [**vlan** *x*] **root** and **show spanning-tree** [**vlan** *x*] **bridge**. Without the VLAN listed, each command lists one line per VLAN; with the VLAN, the output lists the same information, but just for that one VLAN.

Configuring STP Port Costs

Changing the STP port costs requires a simple interface subcommand: **spanning-tree** [**vlan** *x*] **cost** *x*. To show how it works, consider the following example, which changes what happens in the network shown in Figure 3-4.

Back in Figure 3-4, with default settings, SW1 became root, and SW3 blocked on its G0/2 interface. A brief scan of the figure, based on the default STP cost of 4 for Gigabit interfaces, shows that SW3 should have found a cost 4 path and a cost 8 path to reach the root, as shown in Figure 3-5.

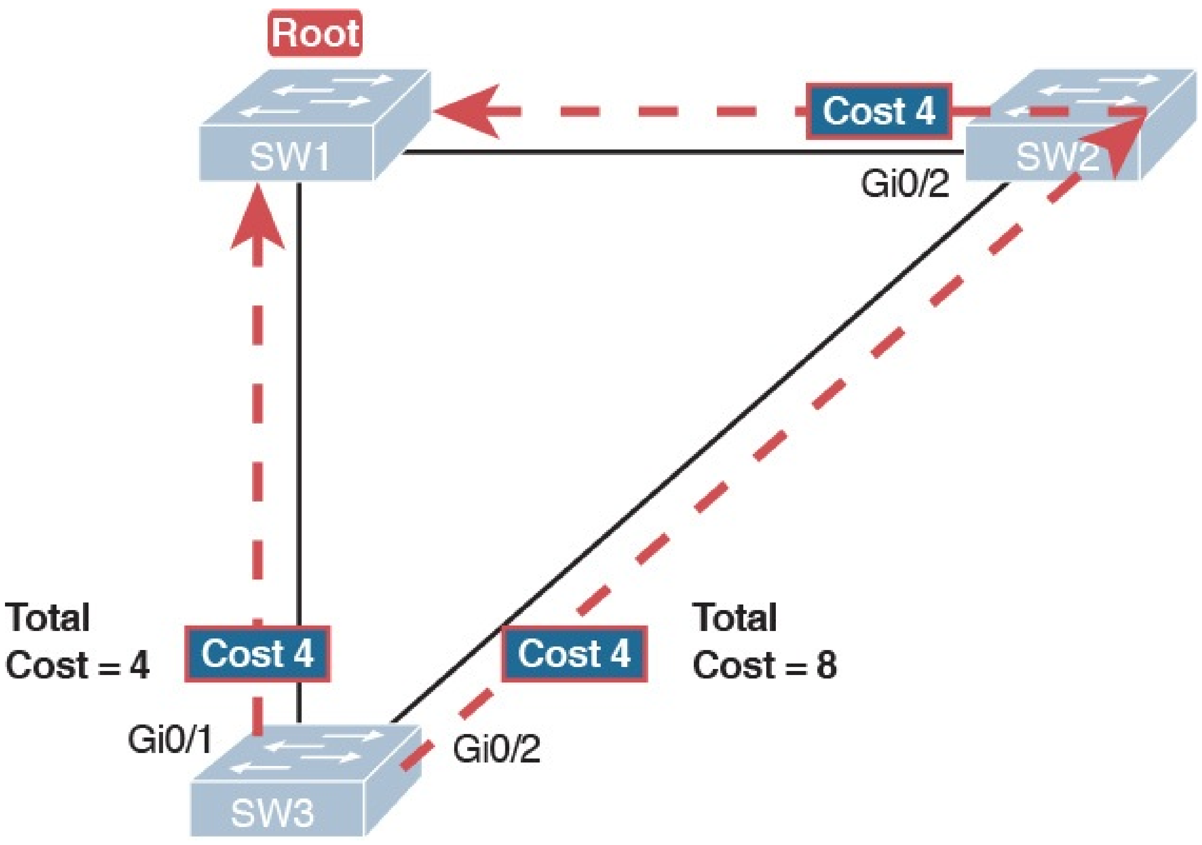


Figure 3-5 *Analysis of SW3's Current Root Cost of 4 with Defaults*

To show the effects of changing the port cost, the next example shows a change to SW3's configuration, setting its G0/1 port cost higher so that the better path to the root goes out SW3's G0/2 port instead. Example 3-3 also shows several other interesting effects.

Example 3-3 *Manipulating STP Port Cost and Watching the Transition to Forwarding State*

Click here to view code image

```
SW3# debug spanning-tree events
Spanning Tree event debugging is on
SW3# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW3(config)# interface gigabitethernet0/1
SW3(config-if)# spanning-tree vlan 10 cost 30
SW3(config-if)# ^Z
SW3#
*Mar 11 06:28:00.860: STP: VLAN0010 new root port
Gi0/2, cost 8
*Mar 11 06:28:00.860: STP: VLAN0010 Gi0/2 -> listening
*Mar 11 06:28:00.860: STP: VLAN0010 sent Topology
```

```
Change Notice on Gi0/2
*Mar 11 06:28:00.860: STP[10]: Generating TC trap for
port GigabitEthernet0/1
*Mar 11 06:28:00.860: STP: VLAN0010 Gi0/1 -> blocking
*Mar 11 06:28:15.867: STP: VLAN0010 Gi0/2 -> learning
*Mar 11 06:28:30.874: STP[10]: Generating TC trap for
port GigabitEthernet0/2
*Mar 11 06:28:30.874: STP: VLAN0010 sent Topology
Change Notice on Gi0/2
*Mar 11 06:28:30.874: STP: VLAN0010 Gi0/2 -> forwarding
```

This example starts with the **debug spanning-tree events** command on SW3. This command tells the switch to issue debug log messages whenever STP performs changes to an interface's role or state. These messages show up in the example as a result of the configuration.

Next, the example shows the configuration to change SW3's port cost, in VLAN 10, to 30, with the **spanning-tree vlan 10 cost 30** interface subcommand. Based on the figure, the root cost through SW3's G0/1 will now be 30 instead of 4. As a result, SW3's best cost to reach the root is cost 8, with SW3's G0/2 as its root port.

The debug messages tell us what STP on SW3 is thinking behind the scenes, with timestamps. Note that the first five debug messages, displayed immediately after the user exited configuration mode in this case, all happen at the same time (down to the same millisecond). Notably, G0/1, which had been forwarding, immediately moves to a blocking state. Interface G0/2, which had been blocking, does not go to a forwarding state, instead moving to a listening state (at least, according to this message).

Now look for the debug message that lists G0/2 transitioning to learning state, and then the next one that shows it finally reaching forwarding state. How long between the messages? In each case, the message's timestamps show that 15 seconds passed. In this experiment, the switches used a default setting of forward delay (15 seconds). So, these debug messages confirm the steps that STP takes to transition an interface from blocking to forwarding state.

If you did not happen to enable a debug when configuring the cost, using **show** commands later can confirm the same choice by SW3, to now use its G0/2 port as its RP. Example 3-4 shows the new STP port cost setting on SW3, along with the new root port and root cost, using the **show spanning-tree vlan 10** command. Note that G0/2 is now listed as the root port. The top of the output lists SW3's root cost as 8, matching the analysis shown in Figure 3-5.

Example 3-4 *New STP Status and Settings on SW3*

Click here to view code image

```
SW3# show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
             Address     1833.9d7b.0e80
             Cost        8
             Port        26 (GigabitEthernet0/2)
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-
ext 10)
             Address     f47f.35cb.d780
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -------
-------------------------
Fa0/23              Desg FWD 19        128.23   P2p
Gi0/1               Altn BLK 30        128.25   P2p
Gi0/2               Root FWD 4         128.26   P2p
```

Configuring Priority to Influence the Root Election

The other big STP configuration option is to influence the root election by changing the priority of a switch. The priority can be set explicitly with the **spanning-tree vlan** *vlan-id* **priority** *value* global configuration command, which sets the base priority of the switch. (This is the command that requires a parameter of a multiple of 4096.)

However, Cisco gives us a better configuration option than configuring a specific priority value. In most designs, the network engineers pick two switches to be root: one to be root if all switches are up, and another to take over if the first switch fails. Switch IOS supports this idea with the **spanning-tree vlan** *vlan-id* **root primary** and **spanning-tree vlan** *vlan-id* **root secondary** commands.

The **spanning-tree vlan** *vlan-id* **root primary** command tells the switch to set its priority low enough to become root right now. The switch looks at the current root in that VLAN, and at the root's priority. Then the local switch chooses a priority value that causes the local switch to take over as root.

Remembering that Cisco switches use a default base priority of 32,768, this command chooses the base priority as follows:

Key Topic

- If the current root has a base priority higher than 24,576, the local switch uses a base priority of 24,576.
- If the current root's base priority is 24,576 or lower, the local switch sets its base priority to the highest multiple of 4096 that still results in the local switch becoming root.

For the switch intended to take over as the root if the first switch fails, use the **spanning-tree vlan** *vlan-id* **root secondary** command. This command is much like the **spanning-tree vlan** *vlan-id* **root primary** command, but with a priority value worse than the primary switch but better than all the other switches. This command sets the switch's base priority to 28,672 regardless of the current root's current priority value.

For example, in Figures 3-4 and 3-5, SW1 was the root switch, and as shown in various commands, all three switches defaulted to use a base priority of 32,768. Example 3-5 shows a configuration that makes SW2 the primary root, and SW1 the secondary, just to show the role move from one to the other. These commands result in SW2 having a base priority of 24,576, and SW1 having a base priority of 28,672.

Example 3-5 *Making SW2 Become Root Primary, and SW1 Root Secondary*

Click here to view code image

```
! First, on SW2:
SW2# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW2(config)# spanning-tree vlan 10 root primary
SW2(config)# ^Z
```

```
! Next, SW1 is configured to back-up SW1
SW1# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW1(config)# spanning-tree vlan 10 root secondary
SW1(config)# ^Z
SW1#

! The next command shows the local switch's BID (SW1)
```

```
SW1# show spanning-tree vlan 10 bridge

                                                                   Hello
Vlan                                 Bridge
ID                  Time   Age   Dly   Protocol
---------------- ---------------------------------- ----
-  ---   ---   --------
VLAN0010            28682  (28672,   10)
1833.9d7b.0e80        2     20    15   ieee

! The next command shows the root's BID (SW2)
SW1# show spanning-tree vlan 10 root

                                                  Root     Hello
Max Fwd
Vlan                           Root ID            Cost
   Time   Age Dly  Root Port
---------------- -------------------- --------- ----- -
-- ---   ------------
VLAN0010            24586
1833.9d7b.1380             4     2    20   15   Gi0/1
```

The output of the two **show** commands clearly points out the resulting priority values on each switch. First, the **show spanning-tree bridge** command lists the local switch's BID information, while the **show spanning-tree root** command lists the root's BID, plus the local switch's root cost and root port (assuming it is not the root switch). So, SW1 lists its own BID, with priority 28,682 (base 28,672, with VLAN 10) with the **show spanning-tree bridge** command. Still on SW1, the output lists the root's priority as 24,586 in VLAN 10, implied as base 24,576 plus 10 for VLAN 10, with the **show spanning-tree root** command.

Note that alternatively you could have configured the priority settings specifically. SW1 could have used the **spanning-tree vlan 10 priority 28672** command, with SW2 using the **spanning-tree vlan 10 priority 24576** command. In this particular case, both options would result in the same STP operation.

Implementing Optional STP Features

This just-completed first major section of the chapter showed examples that used PVST+ only, assuming a default global command of **spanning-tree mode pvst**. At the same time, all the configuration commands shown in that first section, commands that influence STP operation, would influence both traditional STP and RSTP operation.

This section, the second of three major sections in this chapter, now moves on to discuss some useful but optional features that make both STP and RSTP work even better.

Configuring PortFast and BPDU Guard

You can easily configure the PortFast and BPDU Guard features on any interface, but with two different configuration options. One option works best when you want to enable these features only on a few ports, and the other works best when you want to enable these features on most every access port.

First, to enable the features on just one port at a time, use the **spanning-tree portfast** and the **spanning-tree bpduguard enable** interface subcommands. Example 3-6 shows an example of the process, with SW3's F0/4 interface enabling both features. (Also, note the long warning message IOS lists when enabling PortFast; using PortFast on a port connected to other switches can indeed cause serious problems.)

Example 3-6 *Enabling PortFast and BPDU Guard on One Interface*

Click here to view code image

```
SW3# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW3(config)# interface fastEthernet 0/4
SW3(config-if)# spanning-tree portfast
%Warning: portfast should only be enabled on ports
connected to a single
 host. Connecting hubs, concentrators, switches,
bridges, etc... to this
 interface  when portfast is enabled, can cause
temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on FastEthernet0/4 but
will only
 have effect when the interface is in a non-trunking
mode.

SW3(config-if)# spanning-tree bpduguard ?
  disable  Disable BPDU guard for this interface
  enable   Enable BPDU guard for this interface

SW3(config-if)# spanning-tree bpduguard enable
SW3(config-if)# ^Z
SW3#
```

Example 3-7 shows some brief information about the interface configuration of both PortFast and BPDU Guard. Of course, the **show running-config** command (not shown) would confirm the configuration commands from Example 3-6. The **show spanning-tree interface fastethernet0/4 portfast** command in Example 3-7 lists the PortFast status of the interface; note that the status value of *enabled* is displayed only if PortFast is configured and the interface is up. The **show spanning-tree interface detail** command then shows a line near the end of the output that states that PortFast and BPDU Guard are enabled. Note that this command would not list those two highlighted lines of output if these two features were not enabled.

Example 3-7 *Verifying PortFast and BPDU Guard Configuration*

Click here to view code image

```
SW3# show spanning-tree interface fastethernet0/4
portfast
VLAN0104            enabled

SW11# show spanning-tree interface F0/4 detail
 Port 4 (FastEthernet0/4) of VLAN0001 is designated
forwarding
   Port path cost 19, Port priority 128, Port
Identifier 128.4.
   Designated root has priority 32769, address
bcc4.938b.a180
   Designated bridge has priority 32769, address
bcc4.938b.e500
   Designated port id is 128.4, designated path cost 19
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   The port is in the portfast mode
   Link type is point-to-point by default
   Bpdu guard is enabled
   BPDU: sent 1721, received 0
```

PortFast and BPDU Guard are disabled by default on all interfaces, and to use them, each interface requires interface subcommands like those in Example 3-6. Alternately, for both features, you can enable the feature globally. Then, for interfaces for which the feature should be disabled, you can use another interface subcommand to disable the feature.

The ability to change the global default for these features reduces the number of interface subcommands required. For instance, on an access layer switch with 48 access ports and two uplinks, you probably want to enable both

PortFast and BPDU Guard on all 48 access ports. Rather than requiring the interface subcommands on all 48 of those ports, enable the features globally, and then disable them on the uplink ports.

Table 3-3 summarizes the commands to enable and disable both PortFast and BPDU Guard, both globally and per interface. For instance, the global command **spanning-tree portfast default** changes the default so that all interfaces use PortFast, unless a port also has the **spanning-tree portfast disable** interface subcommand configured.

Action	Globally	One Interface
Disable PortFast	no spanning-tree portfast default	spanning-tree portfast disable
Enable PortFast	spanning-tree portfast default	spanning-tree portfast
Disable BPDU Guard	no spanning-tree portfast bpduguard default	spanning-tree bpduguard disable
Enable BPDU Guard	spanning-tree portfast bpduguard default	spanning-tree bpduguard enable

Table 3-3 Enabling and Disabling PortFast and BPDU Gsuard, Globally and Per Interface

Example 3-8 shows another new command, **show spanning-tree summary**. This command shows the current global settings for several STP parameters, including the PortFast and BPDU Guard features. This output was gathered on a switch that had enabled both PortFast and BPDU Guard globally.

Example 3-8 *Displaying Status of Global Settings for PortFast and BPDU Guard*

Click here to view code image

```
SW1# show spanning-tree summary
Switch is in pvst mode
Root bridge for: none
EtherChannel misconfig guard is enabled
Extended system ID           is enabled
Portfast Default             is enabled
PortFast BPDU Guard Default  is enabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Configured Pathcost method used is short

Name                   Blocking Listening Learning
Forwarding STP Active
```

```
---------------------- -------- --------- -------- ----
------ ----------
VLAN0001                     3         0         0
---------------------- -------- --------- -------- ----
------ ----------
1
vlan                       3         0         0
```

Configuring EtherChannel

As introduced back in Chapter 2, two neighboring switches can treat multiple parallel links between each other as a single logical link called an *EtherChannel*. STP operates on the EtherChannel, instead of the individual physical links, so that STP either forwards or blocks on the entire logical EtherChannel for a given VLAN. As a result, a switch in a forwarding state can then load balance traffic over all the physical links in the EtherChannel. Without EtherChannel, only one of the parallel links between two switches would be allowed to forward traffic, with the rest of the links blocked by STP.

Note

All references to EtherChannel in this Chapter refer to Layer 2 EtherChannels, and not to Layer 3 EtherChannels (as discussed in Chapter 19, “IPv4 Routing in the LAN”).

EtherChannel may be one of the most challenging switch features to make work. First, the configuration has several options, so you have to remember the details of which options work together. Second, the switches also require a variety of other interface settings to match among all the links in the channel, so you have to know those settings as well.

This section focuses on the correct EtherChannel configuration. Chapter 4’s section “Troubleshooting Layer 2 EtherChannel” looks at many of the potential problems with EtherChannel, including all those other configuration settings that a switch checks before allowing the EtherChannel to work.

Configuring a Manual EtherChannel

The simplest way to configure an EtherChannel is to add the correct **channel-group** configuration command to each physical interface, on each switch, all with the **on** keyword. The **on** keyword tells the switches to place a physical interface into an EtherChannel.

Before getting into the configuration and verification, however, you need to start using three terms as synonyms: *EtherChannel*, *PortChannel*, and *Channel-group*. Oddly, IOS uses the **channel-group** configuration command, but then to display its status, IOS uses the **show etherchannel** command. Then, the output of this **show** command refers to neither an "EtherChannel" nor a "Channel-group," instead using the term "PortChannel." So, pay close attention to these three terms in the example.

To configure an EtherChannel manually, follow these steps:

Key Topic

Step 1. Add the **channel-group** *number* **mode on** command in interface configuration mode under each physical interface that should be in the channel to add it to the channel.

Step 2. Use the same number for all commands on the same switch, but the channel-group number on the neighboring switch can differ.

Example 3-9 shows a simple example, with two links between switches SW1 and SW2, as shown in Figure 3-6. The configuration shows SW1's two interfaces placed into channel-group 1, with two **show** commands to follow.

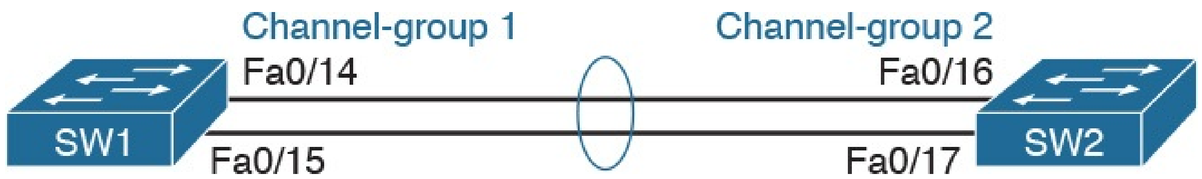


Figure 3-6 *Sample LAN Used in EtherChannel Example*

Example 3-9 *Configuring and Monitoring EtherChannel*

Click here to view code image

```
SW1# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW1(config)# interface fa 0/14
SW1(config-if)# channel-group 1 mode on
SW1(config)# interface fa 0/15
SW1(config-if)# channel-group 1 mode on
SW1(config-if)# ^Z

SW1# show spanning-tree vlan 3

VLAN0003
  Spanning tree enabled protocol ieee
  Root ID     Priority    28675
              Address     0019.e859.5380
              Cost        12
              Port        72 (Port-channel1)
              Hello Time   2 sec  Max Age 20
```

```
sec   Forward Delay 15 sec

  Bridge ID  Priority    28675  (priority 28672 sys-id-
ext 3)
             Address     0019.e86a.6f80
             Hello Time   2 sec  Max Age 20
sec   Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- ----------
----------------------
Po1              Root FWD 12        128.64   P2p
Peer(STP)

SW1# show etherchannel 1 summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate
aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+----------------------
-------------------------
1      Po1(SU)          -        Fa0/14(P)   Fa0/15(P)
```

Take a few moments to look at the output in the two **show** commands in the example, as well. First, the **show spanning-tree** command lists Po1, short for PortChannel1, as an interface. This interface exists because of the **channel-group** commands using the **1** parameter. STP no longer operates on physical interfaces F0/14 and F0/15, instead operating on the PortChannel1 interface, so only that interface is listed in the output.

Next, note the output of the **show etherchannel 1 summary** command. It lists as a heading "Port-channel," with Po1 below it. It also lists both F0/14 and F0/15 in the list of ports, with a (P) beside each. Per the legend, the *P* means that the ports are bundled in the port channel, which is a code that means these ports have passed all the configuration checks and are valid to be included in the channel.

Note

Cisco uses the term *EtherChannel* to refer to the concepts discussed in this section. To refer to the item configured in the switch, Cisco instead uses the term *port channel*, with the command keyword **port-channel**. For the purposes of understanding the technology, you may treat these terms as synonyms. However, it helps to pay close attention to the use of the terms *port channel* and *EtherChannel* as you work through the examples in this section, because IOS uses both.

Configuring Dynamic EtherChannels

Cisco switches support two different protocols that allow the switches to negotiate whether a particular link becomes part of an EtherChannel or not. Basically, the configuration enables the protocol for a particular channel-group number. At that point, the switch can use the protocol to send messages to/from the neighboring switch and discover whether their configuration settings pass all checks. If a given physical link passes, the link is added to the EtherChannel and used; if not, it is placed in a down state, and not used, until the configuration inconsistency can be resolved.

Cisco switches support the Cisco-proprietary Port Aggregation Protocol (PAgP) and the IEEE standard Link Aggregation Control Protocol (LACP), based on IEEE standard 802.3ad. Although differences exist between the two, to the depth discussed here, they both accomplish the same task: negotiate so that only links that pass the configuration checks are actually used in an EtherChannel.

To configure either protocol, a switch uses the **channel-group** configuration commands on each switch, but with a keyword that either means "use this protocol and begin negotiations" or "use this protocol and wait for the other switch to begin negotiations." As shown in Figure 3-7, the **desirable** and **auto** keywords enable PAgP, and the **active** and **passive** keywords enable LACP. With these options, at least one side has to begin the negotiations. In other words, with PAgP, at least one of the two sides must use **desirable**, and with LACP, at least one of the two sides must use **active**.

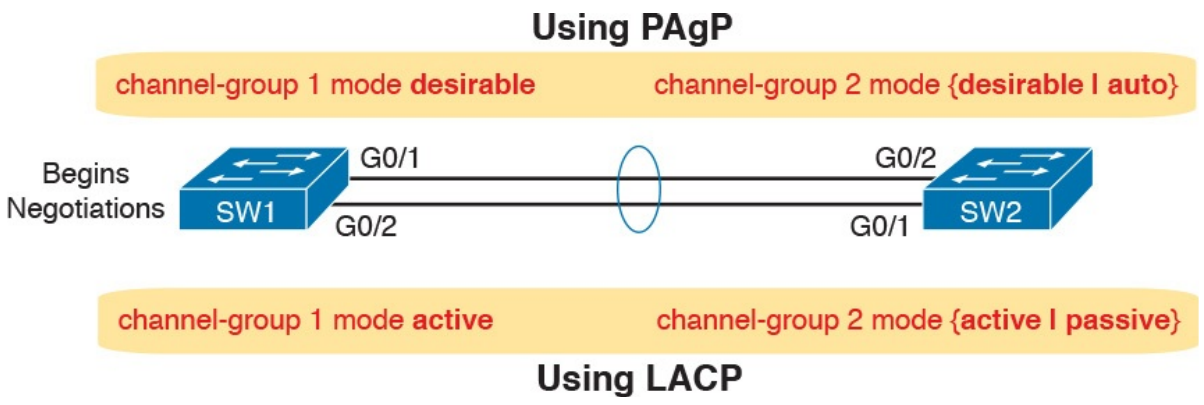

Using PAgP
channel-group 1 mode desirable
channel-group 2 mode {desirable | auto}
Begins Negotiations
SW1
G0/1
G0/2
G0/2
G0/1
SW2
channel-group 1 mode active
channel-group 2 mode {active | passive}
Using LACP

Figure 3-7 *Correct EtherChannel Configuration Combinations*

Note

Do not use the **on** parameter on one end, and either **auto** or **desirable** (or for LACP, **active** or **passive**) on the neighboring switch. The **on** option uses neither PAgP nor LACP, so a configuration that uses **on**, with PAgP or LACP options on the other end, would prevent the EtherChannel from working.

For example, in the design shown in Figure 3-7, imagine both physical interfaces on both switches were configured with the **channel-group 2 mode desirable** interface subcommand. As a result, the two switches would negotiate and create an EtherChannel. Example 3-10 shows the verification of that configuration, with the command **show etherchannel 2 port-channel**. This command confirms the protocol in use (PAgP, because the **desirable** keyword was configured), and the list of interfaces in the channel.

Example 3-10 *EtherChannel Verification: PAgP Desirable Mode*

Click here to view code image

```
SW1# show etherchannel 2 port-channel
                Port-channels in the group:
                ---------------------------

Port-channel: Po2
------------

Age of the Port-channel   = 0d:00h:04m:04s
Logical slot/port    = 16/1          Number of ports = 2
GC                   = 0x00020001      HotStandBy port =
null
Port state           = Port-channel Ag-Inuse
Protocol             =   PAgP
```

```
Port security        = Disabled

Ports in the Port-channel:

Index   Load   Port     EC state        No of bits
------+------+------+------------------+-----------
  0     00     Gi0/1    Desirable-Sl       0
  0     00     Gi0/2    Desirable-Sl       0

Time since last port
bundled:    0d:00h:03m:57s    Gi0/2
```

Implementing RSTP

All you have to do to migrate from STP to RSTP is to configure the **spanning-tree mode rapid-pvst** global command on all the switches. However, for exam preparation, it helps to work through the various **show** commands, particularly to prepare for Simlet questions. Those questions can ask you to interpret **show** command output without allowing you to look at the configuration, and the output of **show** commands when using STP versus RSTP is very similar.

This third and final major section of this chapter focuses on pointing out the similarities and differences between STP and RSTP as seen in Catalyst switch configuration and verification commands. This section explains the configuration and verification of RSTP, with emphasis on how to identify RSTP features.

Identifying the STP Mode on a Catalyst Switch

Cisco Catalyst switches operate in some STP mode as defined by the **spanning-tree mode** global configuration command. Based on this command's setting, the switch is using either 802.1D STP or 802.1w RSTP, as noted in Table 3-4.

Key Topic

Parameter on spanning-tree mode Command	Uses STP or RSTP?	Protocol Listed in Command Output	Description
pvst	STP	ieee	Default; Per-VLAN Spanning Tree instance
rapid-pvst	RSTP	rstp	Like PVST, but uses RSTP rules instead of STP for each STP instance
mst	RSTP	mst	Creates multiple RSTP instances but does not require one instance per each VLAN

Table 3-4 Cisco Catalyst STP Configuration Modes

To determine whether a Cisco Catalyst switch uses RSTP, you can look for two types of information. First, you can look at the configuration, as noted in the left column of Table 3-4. Also, some **show** commands list the STP protocol as a reference to the configuration of the **spanning-tree mode** global configuration command. A protocol of **rstp** or **mst** refers to one of the modes that uses RSTP, and a protocol of **ieee** refers to the mode that happens to use STP.

Before looking at an example of the output, review the topology in Figure 3-8. The remaining RSTP examples in this chapter use this topology. In the RSTP examples in this chapter, SW1 will become root, and SW3 will block on one port (G0/2), as shown.

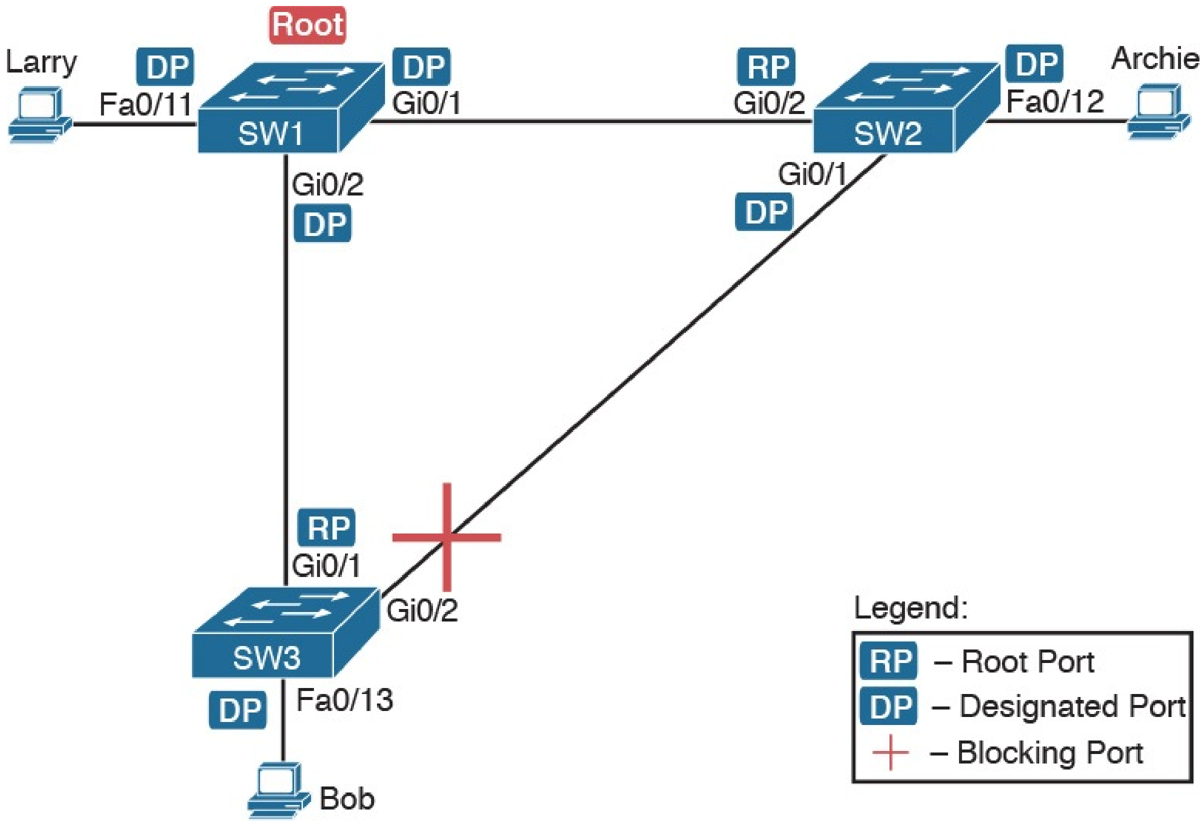


Figure 3-8 *Network Topology for STP and RSTP Examples*

The first example focuses on VLAN 10, with all switches using 802.1D STP and the default setting of **spanning-tree mode pvst**. This setting creates an instance of STP per VLAN (which is the per-VLAN part of the name) and uses 802.1D STP. Each switch places the port connected to the PC into VLAN 10 and enables both PortFast and BPDU Guard. Example 3-11 shows a sample configuration from switch SW3, with identical interface subcommands configured on SW1's F0/11 and SW2's F0/12 ports, respectively.

Example 3-11 *Sample Configuration from Switch SW3*

Click here to view code image

```
SW3# show running-config interface Fastethernet 0/13

Building configuration...

Current configuration : 117 bytes
!
interface FastEthernet0/13
 switchport access vlan 10
 spanning-tree portfast
 spanning-tree bpduguard enable
end
```

At this point, the three switches use 802.1D STP because all use the default PVST mode. Example 3-12 shows the evidence of STP's work, with only subtle and indirect clues that STP happens to be in use.

Example 3-12 *Output That Confirms the Use of 802.1D STP on Switch SW3*

Click here to view code image

```
SW3# show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
             Address     1833.9d7b.0e80
             Cost        4
             Port        25 (GigabitEthernet0/1)
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-
ext 10)
             Address     f47f.35cb.d780
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -------
-------------------------
Fa0/13              Desg FWD 19        128.13   P2p
```

```
Edge
Gi0/1               Root FWD 4         128.25    P2p
Gi0/2               Altn BLK 4         128.26    P2p


SW3# show spanning-tree vlan 10 bridge

                                                       Hello
Vlan                             Bridge
ID                Time   Age   Dly  Protocol
---------------- --------------------------------- ----
-  ---  ---  ---------
VLAN0010          32778 (32768,  10)
f47f.35cb.d780     2     20   15   ieee
```

The highlighted parts of the example note the references to the STP protocol as ieee, which implies that STP is in use. The term *ieee* is a reference to the original IEEE 802.1D STP standard.

To migrate this small network to use RSTP, configure the **spanning-tree mode rapid-pvst** command. This continues the use of per-VLAN spanning-tree instances, but it applies RSTP logic to each STP instance. Example 3-13 shows the output of the same two commands from Example 3-12 after configuring the **spanning-tree mode rapid-pvst** command on all three switches.

Example 3-13 *Output That Confirms the Use of 802.1w RSTP on Switch SW3*

Click here to view code image

```
SW3# show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol rstp
  Root ID    Priority    32778
             Address     1833.9d7b.0e80
             Cost        4
             Port        25 (GigabitEthernet0/1)
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-
ext 10)
             Address     f47f.35cb.d780
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
             Aging Time  300 sec
```

```
Interface           Role Sts Cost       Prio.Nbr Type
------------------- ---- --- ---------- -------- -------
-------------------------
Fa0/13              Desg FWD 19         128.13   P2p
Edge
Gi0/1               Root FWD 4          128.25   P2p
Gi0/2               Altn BLK 4          128.26   P2p


SW3# show spanning-tree vlan 10 bridge

                                                    Hellc
Vlan                             Bridge
ID                Time   Age   Dly  Protocol
---------------- --------------------------------- ----
-  ---   ---   --------
VLAN0010          32778 (32768,   10)
f47f.35cb.d780     2     20    15   rstp
```

Pay close attention to the differences between the 802.1D STP output in Example 3-12 and the 802.1w RSTP output in Example 3-13. Literally, the only difference is rstp instead of ieee in one place in the output of each of the two commands listed. In this case, rstp refers to the configuration of the **spanning-tree mode rapid-pvst** global config command, which implied the use of RSTP.

RSTP Port Roles

RSTP adds two port roles to STP: the alternate port and the backup port. Example 3-14 repeats an excerpt from the **show spanning-tree vlan 10** command on switch SW3 to show an example of the alternate port role. SW3 (as shown earlier in Figure 3-8) is not the root switch, with G0/1 as its root port and G0/2 as an alternate port.

Example 3-14 *Output Confirming SW3's Root Port and Alternate Port Roles*

Click here to view code image

```
SW3# show spanning-tree vlan 10
! Lines omitted for brevity
Interface           Role Sts Cost       Prio.Nbr Type
------------------- ---- --- ---------- -------- -------
-------------------------
Fa0/13              Desg FWD 19         128.13   P2p
Edge
```

```
Gi0/1               Root FWD 4         128.25   P2p
Gi0/2               Altn BLK 4         128.26   P2p
```

The good news is that the output clearly lists which port is the root port (Gi0/1) and which port is the alternate root port (Gi0/2). The only trick is to know that Altn is a shortened version of the word *alternate*.

Pay close attention to this short description of an oddity about the STP and RSTP output on Catalyst switches! Cisco Catalyst switches often show the alternate and backup ports in output even when using STP and not RSTP. The alternate and backup port concepts are RSTP concepts. The switches only converge faster using these concepts when using RSTP. But **show** command output, when using STP and not RSTP, happens to identify what would be the alternate and backup ports if RSTP were used.

Key Topic

Why might you care about such trivia? Seeing output that lists an RSTP alternate port does not confirm that the switch is using RSTP. So, do not make that assumption on the exam. To confirm that a switch uses RSTP, you must look at the configuration of the **spanning-tree mode** command, or look for the protocol as summarized back in Table 3-4.

For instance, just compare the output of Example 3-12 and Example 3-14. Example 3-12 shows output for this same SW3, with the same parameters, except that all switches used PVST mode, meaning all the switches used STP. Example 3-12's output (based on STP) lists SW3's G0/2 as Altn, meaning alternate, even though the alternate port concept is not an STP concept, but an RSTP concept.

RSTP Port States

RSTP added one new port state compared to STP, discarding, using it as a replacement for the STP port states of disabled and blocking. You might think that after you configure a switch to use RSTP rather than STP, instead of seeing ports in a blocking state, you would now see the discarding state. However, the Cisco Catalyst switch output basically ignores the new term *discarding*, continuing to use the old term *blocking* instead.

For example, scan back to the most recent RSTP example (Example 3-14), to the line for SW3's port G0/2. Then look for the column with heading STS, which refers to the status or state. The output shows G0/2 is listed as BLK, or blocking. In theory, because SW3 uses RSTP, the port state ought to be discarding, but the switch IOS continues to use the older notation of BLK for blocking.

Just as one more bit of evidence, the command **show spanning-tree vlan 10 interface gigabitethernet0/2 state** lists the STP or RSTP port state with the state fully spelled out. Example 3-15 shows this command, taken from SW3, for interface G0/2. Note the fully spelled-out *blocking* term instead of the RSTP term *discarding*.

Example 3-15 *SW3, an RSTP Switch, Continues to Use the Old Blocking Term*

Click here to view code image

```
SW3# show spanning-tree vlan 10 interface
gigabitEthernet 0/2 state
VLAN0010            blocking
```

RSTP Port Types

Cisco Catalyst switches determine the RSTP port type based on two port settings: the current duplex (full or half) and whether the PortFast feature is enabled. First, full duplex tells the switch to use port type point-to-point, with half duplex telling the switch to use port type shared. Enabling PortFast tells the switch to treat the port as an edge port. Table 3-5 summarizes the combinations.

Type	Current Duplex Status	Is Spanning-Tree PortFast Configured?
Point-to-point	Full	No
Point-to-point edge	Full	Yes
Shared	Half	No
Shared edge[1]	Half	Yes

[1] Cisco recommends against using this combination, to avoid causing loops.

Table 3-5 RSTP Port Types

You can easily find the RSTP port types in the output of several commands, including the same **show spanning-tree** command in Example 3-16. Example 3-16 lists output from switch SW2, with a hub added off SW2's F0/18 port (not shown in Figure 3-8). The hub was added so that the output in Example 3-16 lists a shared port (noted as Shr) to go along with the point-to-point ports (noted as P2p).

Example 3-16 *RSTP Port Types*

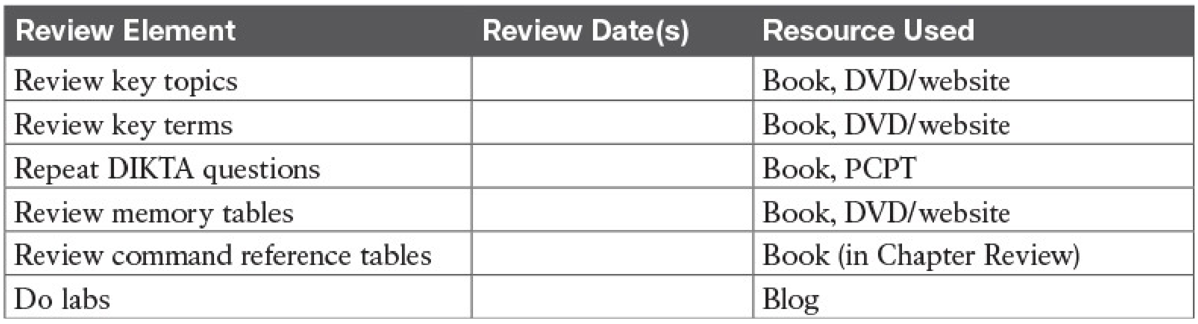
Click here to view code image

```
SW2# show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol rstp
  Root ID    Priority    32778
             Address     1833.9d7b.0e80
             Cost        4
             Port        26 (GigabitEthernet0/2)
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec

  Bridge ID  Priority    32778  (priority 32768 sys-id-
ext 10)
             Address     1833.9d7b.1380
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -------
-------------------------
Fa0/12              Desg FWD 19        128.12   P2p
Edge
Fa0/18              Desg FWD 19        128.18   Shr
Gi0/1               Desg FWD 4         128.25   P2p
Gi0/2               Root FWD 4         128.26   P2p
```

For exam prep, again note an odd fact about the highlighted output in Example 3-16: The port type details appear in the output when using both STP and RSTP. For example, refer to Example 3-12 again, which shows output from SW3 when using STP (when configured for PVST mode). The Type column also identifies point-to-point and edge interfaces.

Chapter Review

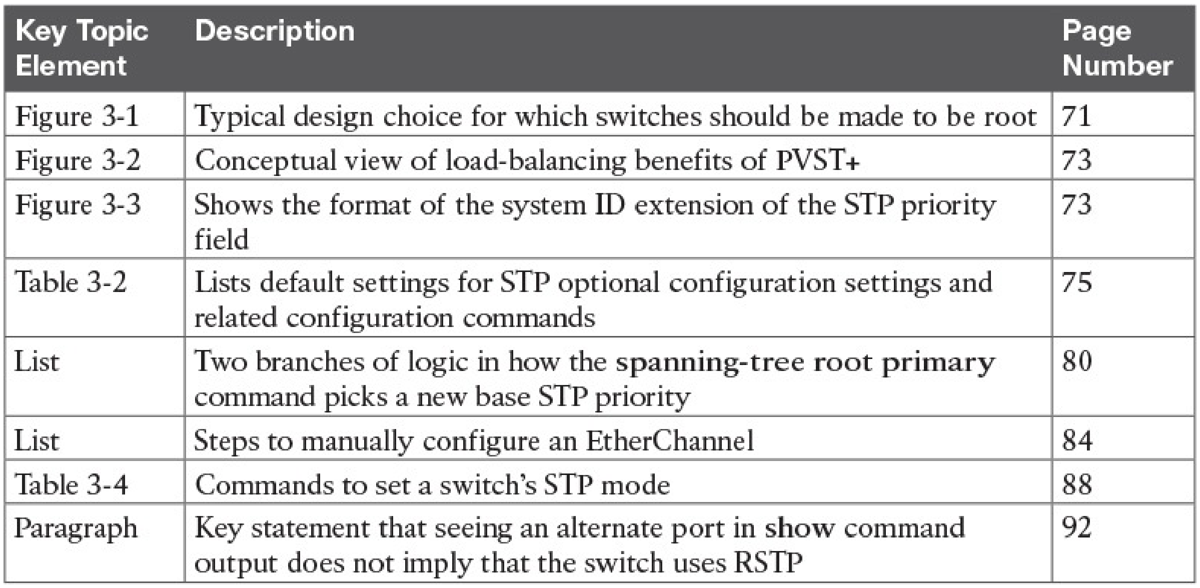
One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book, DVD, or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 3-6 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.



Review Element	Review Date(s)	Resource Used
Review key topics		Book, DVD/website
Review key terms		Book, DVD/website
Repeat DIKTA questions		Book, PCPT
Review memory tables		Book, DVD/website
Review command reference tables		Book (in Chapter Review)
Do labs		Blog

Table 3-6 Chapter Review Tracking

Review All the Key Topics



Key Topic Element	Description	Page Number
Figure 3-1	Typical design choice for which switches should be made to be root	71
Figure 3-2	Conceptual view of load-balancing benefits of PVST+	73
Figure 3-3	Shows the format of the system ID extension of the STP priority field	73
Table 3-2	Lists default settings for STP optional configuration settings and related configuration commands	75
List	Two branches of logic in how the **spanning-tree root primary** command picks a new base STP priority	80
List	Steps to manually configure an EtherChannel	84
Table 3-4	Commands to set a switch's STP mode	88
Paragraph	Key statement that seeing an alternate port in **show** command output does not imply that the switch uses RSTP	92

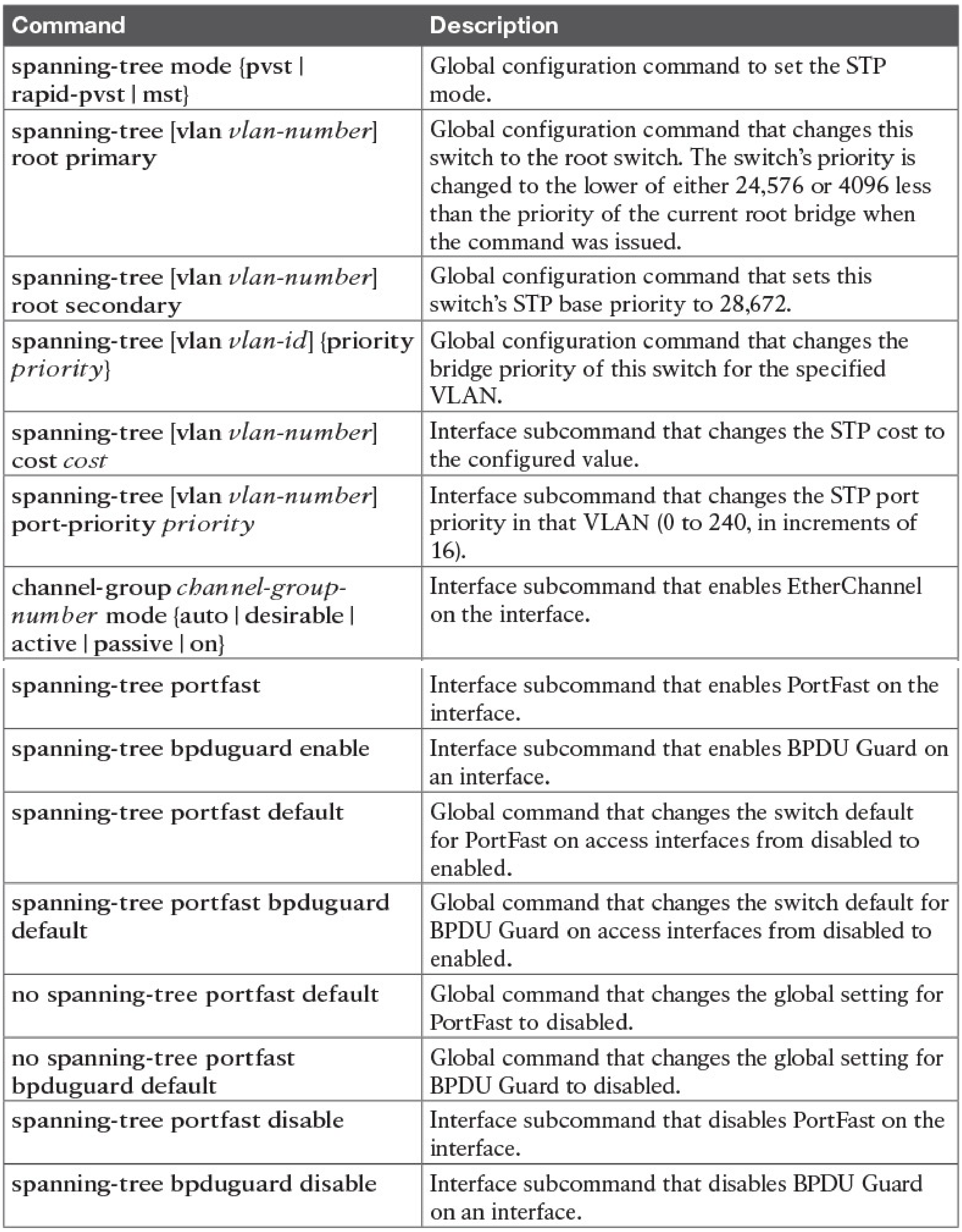
Table 3-7 Key Topics for Chapter 3

Key Terms You Should Know

Rapid PVST+
PVST+
system ID extension
PAgP
LACP
PortChannel
Channel-group

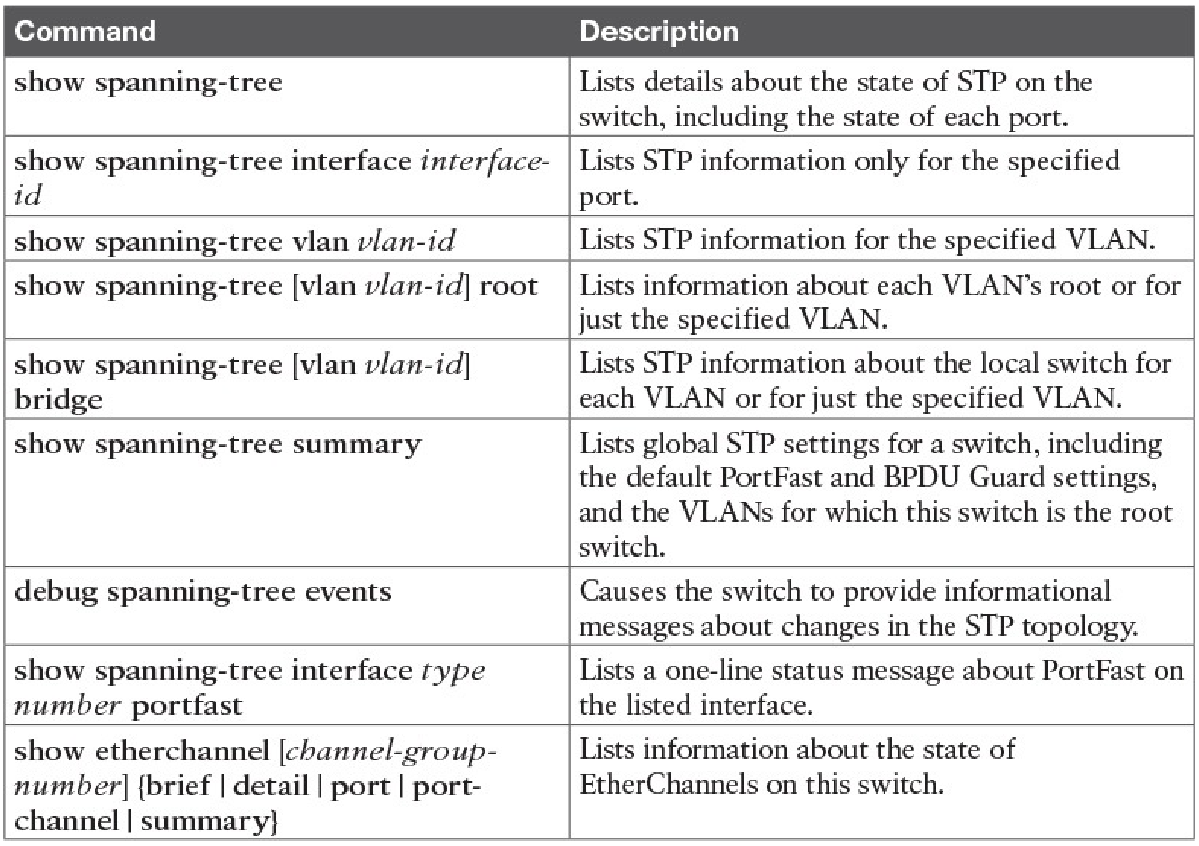
Command References

Tables 3-8 and 3-9 list configuration and verification commands used in this chapter. As an easy review exercise, cover the left column in a table, read the right column, and try to recall the command without looking. Then repeat the exercise, covering the right column, and try to recall what the command does.



Command	Description
spanning-tree mode {pvst | rapid-pvst | mst}	Global configuration command to set the STP mode.
spanning-tree [vlan *vlan-number*] root primary	Global configuration command that changes this switch to the root switch. The switch's priority is changed to the lower of either 24,576 or 4096 less than the priority of the current root bridge when the command was issued.
spanning-tree [vlan *vlan-number*] root secondary	Global configuration command that sets this switch's STP base priority to 28,672.
spanning-tree [vlan *vlan-id*] {priority *priority*}	Global configuration command that changes the bridge priority of this switch for the specified VLAN.
spanning-tree [vlan *vlan-number*] cost *cost*	Interface subcommand that changes the STP cost to the configured value.
spanning-tree [vlan *vlan-number*] port-priority *priority*	Interface subcommand that changes the STP port priority in that VLAN (0 to 240, in increments of 16).
channel-group *channel-group-number* mode {auto | desirable | active | passive | on}	Interface subcommand that enables EtherChannel on the interface.
spanning-tree portfast	Interface subcommand that enables PortFast on the interface.
spanning-tree bpduguard enable	Interface subcommand that enables BPDU Guard on an interface.
spanning-tree portfast default	Global command that changes the switch default for PortFast on access interfaces from disabled to enabled.
spanning-tree portfast bpduguard default	Global command that changes the switch default for BPDU Guard on access interfaces from disabled to enabled.
no spanning-tree portfast default	Global command that changes the global setting for PortFast to disabled.
no spanning-tree portfast bpduguard default	Global command that changes the global setting for BPDU Guard to disabled.
spanning-tree portfast disable	Interface subcommand that disables PortFast on the interface.
spanning-tree bpduguard disable	Interface subcommand that disables BPDU Guard on an interface.

Table 3-8 Chapter 3 Configuration Command Reference



Command	Description
show spanning-tree	Lists details about the state of STP on the switch, including the state of each port.
show spanning-tree interface *interface-id*	Lists STP information only for the specified port.
show spanning-tree vlan *vlan-id*	Lists STP information for the specified VLAN.
show spanning-tree [vlan *vlan-id*] root	Lists information about each VLAN's root or for just the specified VLAN.
show spanning-tree [vlan *vlan-id*] bridge	Lists STP information about the local switch for each VLAN or for just the specified VLAN.
show spanning-tree summary	Lists global STP settings for a switch, including the default PortFast and BPDU Guard settings, and the VLANs for which this switch is the root switch.
debug spanning-tree events	Causes the switch to provide informational messages about changes in the STP topology.
show spanning-tree interface *type number* portfast	Lists a one-line status message about PortFast on the listed interface.
show etherchannel [*channel-group-number*] {brief | detail | port | port-channel | summary}	Lists information about the state of EtherChannels on this switch.

Table 3-9 Chapter 3 EXEC Command Reference

Chapter 4. LAN Troubleshooting

This chapter covers the following exam topics:

1.0 LAN Switching Technologies

1.1 Configure, verify, and troubleshoot VLANs (normal/extended range) spanning multiple switches

1.1.a Access ports (data and voice)

1.1.b Default VLAN

1.2 Configure, verify, and troubleshoot interswitch connectivity

1.2.a Add and remove VLANs on a trunk

1.2.b DTP and VTP (v1&v2)

1.3 Configure, verify, and troubleshoot STP protocols

1.3.a STP mode (PVST+ and RPVST+)

1.3.b STP root bridge selection

1.5 Configure, verify, and troubleshoot (Layer 2/Layer 3) EtherChannel

1.5.a Static

1.5.b PAGP

1.5.c LACP

1.7 Describe common access layer threat mitigation techniques

1.7.c Non-default native VLAN

This chapter discusses the LAN topics discussed in depth in the first three chapters, plus a few prerequisite topics, from a troubleshooting perspective.

Troubleshooting for any networking topic requires a slightly different mindset as compared to thinking about configuration and verification. When thinking about configuration and verification, it helps to think about basic designs, learn how to configure the feature correctly, and learn how to verify the correct configuration is indeed working correctly. However, to learn how to troubleshoot, you need to think about symptoms when the design is incorrect, or if the configuration does not match the good design. What symptoms occur when you make one type of mistake or another? This chapter looks at the common types of mistakes, and works through how to look at the status with **show** commands to find those mistakes.

This chapter breaks the material into four major sections. The first section tackles the largest topic, STP troubleshooting. STP is not likely to fail as a protocol; instead, STP may not be operating as designed, so the task is to find how STP is currently working and discover how to then make the configuration implement the correct design. The second major section then moves on to Layer 2 EtherChannels, which have a variety of small potential problems that can prevent the dynamic formation of an EtherChannel.

The third major section of the chapter focuses on the data plane forwarding of Ethernet frames on LAN switches, in light of VLANs, trunks, STP, and EtherChannels. That same section reviews the Layer 2 forwarding logic of a switch in light of these features. The fourth and final major section then examines VLAN and trunking issues, and how those issues impact switch forwarding.

Note that a few of the subtopics listed within the exam topics at the beginning of this chapter are not discussed in this chapter. This chapter does not discuss VTP beyond its basic features (VTP is discussed in depth in Chapter 5) or Layer 3 EtherChannels (discussed in Chapter 19).

"Do I Know This Already?" Quiz

A few of the troubleshooting chapters in this book not only discuss troubleshooting of specific topics but also serve as a tool to summarize and review some important topics. This chapter is one of those chapters. As a result, it is useful to read these chapters regardless of your current knowledge level. Therefore, this chapter does not include a "Do I Know This Already?" quiz. However, if you feel particularly confident about troubleshooting IPv4 routing features covered in this book, feel free to move to the "Chapter Review" section near the end of this chapter to bypass the majority of the chapter.

Foundation Topics

Troubleshooting STP

STP questions tend to intimidate many test takers. STP uses many rules, with tiebreakers in case one rule ends with a tie. Without much experience with STP, people tend to distrust their own answers. Also, even those of us with networking jobs already probably do not troubleshoot STP very often, because STP works well. Often, troubleshooting STP is not about STP failing to do its job but rather about STP working differently than designed, with a different root switch, or different root ports (RP), and so on. Seldom does STP troubleshooting begin with a case in which STP has failed to prevent a loop.

This section reviews the rules for STP, while emphasizing some important troubleshooting points. In particular, this section takes a closer look at the tiebreakers that STP uses to make decisions. It also makes some practical suggestions about how to go about answering exam questions such as “which switch is the root switch?”

Determining the Root Switch

Determining the STP root switch is easy if you know all the switches’ BIDs: Just pick the lowest value. If the question lists the priority and MAC address separately, as is common in some **show** command output, pick the switch with the lowest priority, or in the case of a tie, pick the lower MAC address value.

And just to be extra clear, STP does not have nor need a tiebreaker for electing the root switch. The BID uses a switch universal MAC address as the last 48 bits of the BID. These MAC addresses are unique in the universe, so there should never be identical BIDs or the need for a tiebreaker.

For the exam, a question that asks about the root switch might not be so simple as listing a bunch of BIDs and asking you which one is “best.” A more likely question is a simulator (sim) question in which you have to do any **show** commands you like or a multiple choice question that lists the output from only one or two commands. Then you have to apply the STP algorithm to figure out the rest.

When faced with an exam question using a simulator, or just the output in an exhibit, use a simple strategy of ruling out switches, as follows:

Key Topic

Step 1. Begin with a list or diagram of switches, and consider all as possible root switches.

Step 2. Rule out any switches that have an RP (**show spanning-tree**, **show spanning-tree root**), because root switches do not have an RP.

Step 3. Always try **show spanning-tree**, because it identifies the local switch as root directly: “This switch is the root” on the fifth line of output.

Step 4. Always try **show spanning-tree root**, because it identifies the local switch as root indirectly: The RP column is empty if the local switch is the root.

Step 5. When using a sim, rather than try switches randomly, chase the RPs. For example, if starting with SW1, and SW1’s G0/1 is an RP, next try the switch on the other end of SW1’s G0/1 port.

Step 6. When using a sim, use **show spanning-tree vlan** *x* on a few switches and record the root switch, RP, and designated port (DP). This strategy can quickly show you most STP facts.

The one step in this list that most people ignore is the idea of ruling out switches that have an RP. Root switches do not have an RP, so any switch with an RP can be ruled out as not being the root switch for that VLAN. Example 4-1 shows two commands on switch SW2 in some LAN that confirms that SW2 has an RP and is therefore not the root switch.

Example 4-1 *Ruling Out Switches as Root Based on Having a Root Port*

Click here to view code image

```
SW2# show spanning-tree vlan 20 root

                                               Root Hello
Max Fwd
Vlan                    Root ID                Cost   Time
Age Dly  Root Port
---------------- -------------------- --------- ----- -
-- ---  ------------
VLAN0020         32788
1833.9d7b.0e80          4     2    20   15   Gi0/2

SW2# show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    32788
             Address     1833.9d7b.0e80
             Cost        4
             Port        26 (GigabitEthernet0/2)
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec

  Bridge ID  Priority    32788  (priority 32768 sys-id-
ext 20)
             Address     1833.9d7b.1380
             Hello Time   2 sec  Max Age 20
sec  Forward Delay 15 sec
             Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -------
-------------------------
Gi0/1               Desg FWD 4         128.25   P2p
Gi0/2               Root FWD 4         128.26   P2p
```

Both commands identify SW2’s G0/2 port as its RP, so if you follow the suggestions, the next switch to try in a sim question would be the switch on the other end of SW2's G0/2 interface.

Determining the Root Port on Nonroot Switches

Determining the RP of a switch when **show** command output is available is relatively easy. As shown recently in Example 4-1, both **show spanning-tree** and **show spanning-tree root** list the root port of the local switch, assuming it is not the root switch. The challenge comes more when an exam question makes you think through how the switches choose the RP based on the root cost of each path to the root switch, with some tiebreakers as necessary.

As a review, each nonroot switch has one, and only one, RP for a VLAN. To choose its RP, a switch listens for incoming Hello bridge protocol data units (BPDU). For each received Hello, the switch adds the cost listed in the hello BPDU to the cost of the incoming interface (the interface on which the Hello was received). That total is the root cost over that path. The lowest root cost wins, and the local switch uses its local port that is part of the least root cost path as its root port.

Although that description has a lot of twists and turns in the words, it is the same concept described for Chapter 2's Figure 2-8.

Most humans can analyze what STP chooses by using a network diagram and a slightly different algorithm. Instead of thinking about Hello messages and so on, approach the question as this: the sum of all outgoing port costs between the nonroot switch and the root. Repeating a familiar example, with a twist, Figure 4-1 shows the calculation of the root cost. Note that SW3's Gi0/1 port has yet again had its cost configured to a different value.

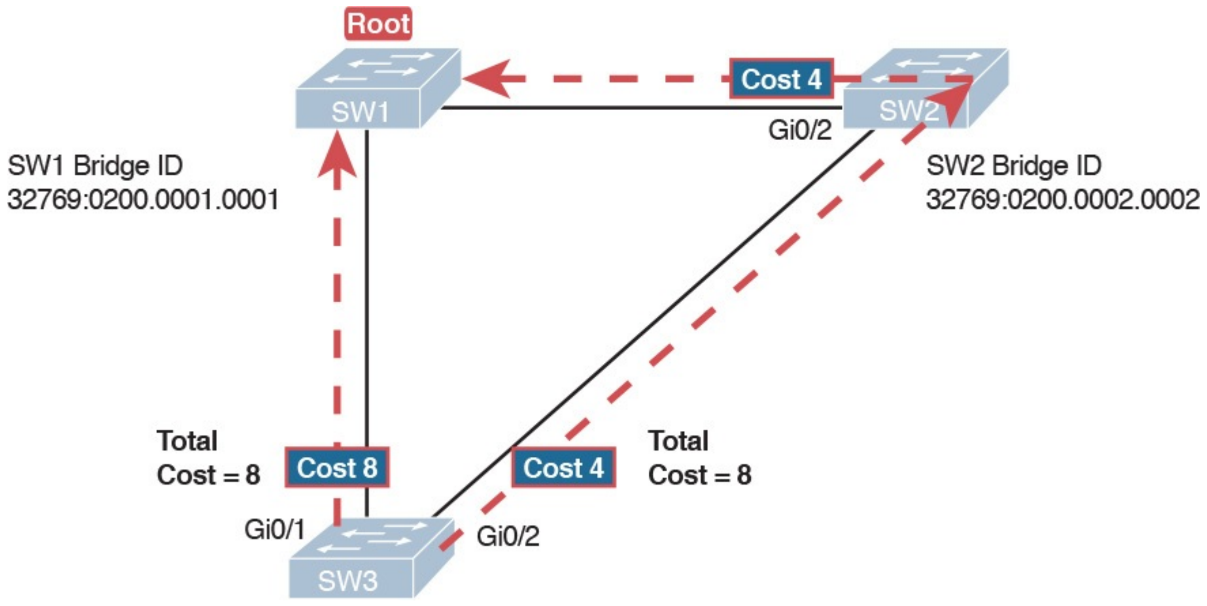


Figure 4-1 *SW3's Root Cost Calculation Ends in a Tie*

STP Tiebreakers When Choosing the Root Port

Figure 4-1 shows the easier process of adding the STP costs of the outgoing interfaces over each from SW3, a nonroot, to SW1, the root. It also shows a tie (on purpose), to talk about the tiebreakers.

When a switch chooses its root port, the first choice is to choose the local port that is part of the least root cost path. When those costs tie, the switch picks the port connected to the neighbor with the lowest BID. This tiebreaker usually breaks the tie, but not always. So, for completeness, the three tiebreakers are, in the order a switch uses them, as follows:

1. Choose based on the lowest neighbor bridge ID.
2. Choose based on the lowest neighbor port priority.
3. Choose based on the lowest neighbor internal port number.

(Note that the switch only considers the root paths that tie when thinking about these tiebreakers.)

For example, Figure 4-1 shows that SW3 is not root and that its two paths to reach the root tie with their root costs of 8. The first tiebreaker is the lowest neighbor's BID. SW1's BID value is lower than SW2's, so SW3 chooses its G0/1 interface as its RP in this case.

The last two RP tiebreakers come into play only when two switches connect to each other with multiple links, as shown in Figure 4-2. In that case, a switch receives Hellos on more than one port from the same neighboring switch, so the BIDs tie.

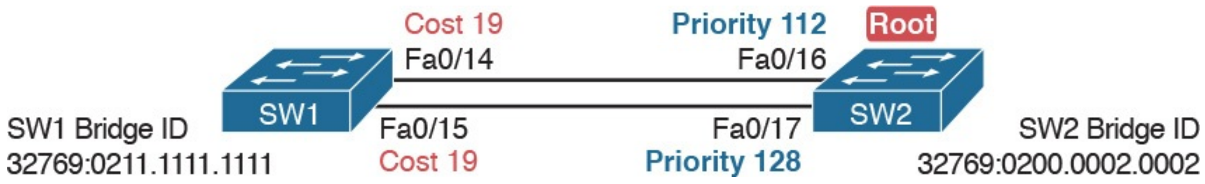


Figure 4-2 *Topology Required for the Last Two Tiebreakers for Root Port*

In this particular example, SW2 becomes root, and SW1 needs to choose its RP. SW1's port costs tie, at 19 each, so SW1's root cost over each path will tie at 19. SW2 sends Hellos over each link to SW1, so SW1 cannot break the tie based on SW1's neighbor BID because both list SW2's BID. So, SW1 has to turn to the other two tiebreakers.

Note

In real life, most engineers would put these two links into an EtherChannel.

The next tiebreaker is a configurable option: the neighboring switch's port priority on each neighboring switch interface. Cisco switch ports default to a setting of 128, with a range of values from 0 through 255, with lower being better (as usual). In this example, the network engineer has set SW2's F0/16 interface with the **spanning-tree vlan 10 port-priority 112** command. SW1 learns that the neighbor has a port priority of 112 on the top link and 128 on the bottom, so SW1 uses its top (F0/14) interface as the root port.

If the port priority ties, which it often does due to the default values, STP relies on an internal port numbering on the neighbor. Cisco switches assign an internal integer to identify each interface on the switch. The nonroot looks for the neighbor's lowest internal port number (as listed in the Hello messages) and chooses its RP based on the lower number.

Cisco switches use an obvious numbering, with Fa0/1 having the lowest number, then Fa0/2, then Fa0/3, and so on. So, in Figure 4-2, SW2's Fa0/16 would have a lower internal port number than Fa0/17; SW1 would learn those numbers in the Hello; and SW1 would use its Fa0/14 port as its RP.

Suggestions for Attacking Root Port Problems on the Exam

Exam questions that make you think about the RP can be easy if you know where to look and the output of a few key commands is available. However, the more conceptual the question, the more you have to calculate the root cost over each path, correlate that to different **show** commands, and put the ideas together. The following list makes a few suggestions about how to approach STP problems on the exam:

Key Topic

1. If available, look at the **show spanning-tree** and **show spanning-tree root** commands. Both commands list the root port and the root cost (see Example 4-1).

2. The **show spanning-tree** command lists cost in two places: the root cost at the top, in the section about the root switch; and the interface cost, at the bottom, in the per-interface section. Be careful, though; the cost at the bottom is the interface cost, not the root cost!

3. For problems where you have to calculate a switch's root cost:

   a. Memorize the default cost values: 100 for 10 Mbps, 19 for 100 Mbps, 4 for 1 Gbps, and 2 for 10 Gbps.

   b. Look for any evidence of the **spanning-tree cost** configuration command on an interface, because it overrides the default cost. Do not assume default costs are used.

   c. When you know a default cost is used, if you can, check the current actual speed as well. Cisco switches choose STP cost defaults based on the current speed, not the maximum speed.

Determining the Designated Port on Each LAN Segment

Each LAN segment has a single switch that acts as the designated port (DP) on that segment. On segments that connect a switch to a device that does not even use STP—for example, segments connecting a switch to a PC or a router—the switch always wins, because it is the only device sending a Hello onto the link. However, links with two switches require a little more work to discover which should be the DP. By definition:

Key Topic

Step 1. For switches connected to the same LAN segment, the switch with the lowest cost to reach the root, as advertised in the Hello they send onto the link, becomes the DP on that link.

Step 2. In case of a tie, among the switches that tied on cost, the switch with the lowest BID becomes the DP.

For example, consider Figure 4-3. This figure notes the root, RPs, and DPs and each switch's least cost to reach the root over its respective RP.

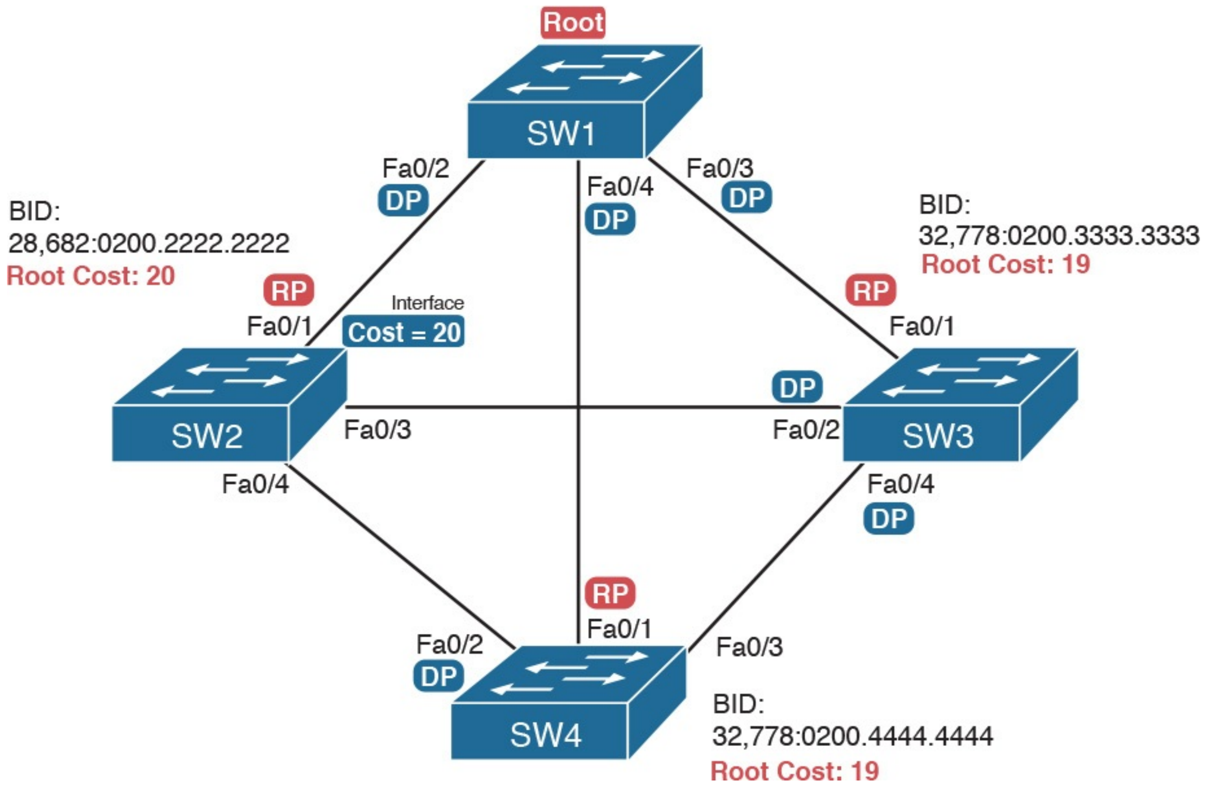


Figure 4-3 *Picking the DPs*

Focus on the segments that connect the nonroot switches for a moment:

SW2–SW4 segment: SW4 wins because of its root cost of 19, compared to SW2's root cost of 20.

SW2–SW3 segment: SW3 wins because of its root cost of 19, compared to SW2's root cost of 20.

SW3–SW4 segment: SW3 and SW4 tie on root cost, both with root cost 19. SW3 wins due to its better (lower) BID value.

Interestingly, SW2 loses and does not become DP on the links to SW3 and SW4 even though SW2 has the better (lower) BID value. The DP tiebreaker does use the lowest BID, but the first DP criteria is the lowest root cost, and SW2's root cost happens to be higher than SW3's and SW4's.

Note

A single switch can connect two or more interfaces to the same collision domain, and compete to become DP, if hubs are used. In such cases, two different switch ports on the same switch tie, the DP choice uses the same two final tiebreakers as used with the RP selection: the lowest interface STP priority, and if that ties, the lowest internal interface number.

Suggestions for Attacking Designated Port Problems on the Exam

As with exam questions asking about the RP, exam questions that make you think about the DP can be easy if you know where to look and the output of a few key commands is available. However, the more conceptual the question, the more you have to think about the criteria for choosing the DP: first the root cost of the competing switches, and then the better BID if they tie based on root cost.

The following list gives some tips to keep in mind when digging into a given DP issue. Some of this list repeats the suggestions for finding the RP, but to be complete, this list includes each idea as well.

Key Topic

1. If available, look at the **show spanning-tree** commands, at the list of interfaces at the end of the output. Then, look for the Role column, and look for Desg, to identify any DPs.
2. Identify the root cost of a switch directly by using the **show spanning-tree** command. But be careful! This command lists the cost in two places, and only the mention at the top, in the section about the root, lists the root cost.
3. For problems where you have to calculate a switch's root cost, do the following:
   a. Memorize the default cost values: 100 for 10 Mbps, 19 for 100

Mbps, 4 for 1 Gbps, and 2 for 10 Gbps.

b. Look for any evidence of the **spanning-tree cost** configuration command on an interface, because it overrides the default cost. Do not assume default costs are used.

c. When you know a default cost is used, if you can, check the current actual speed as well. Cisco switches choose STP cost defaults based on the current speed, not the maximum speed.

STP Convergence

STP puts each RP and DP into a forwarding state, and ports that are neither RP nor DP into a blocking state. Those states may remain as is for days, weeks, or months. But at some point, some switch or link will fail, a link may change speeds (changing the STP cost), or the STP configuration may change. Any of these events can cause switches to repeat their STP algorithm, which may in turn change their own RP and any ports that are DPs.

When STP converges based on some change, not all the ports have to change their state. For instance, a port that was forwarding, if it still needs to forward, just keeps on forwarding. Ports that were blocking that still need to block keep on blocking. But when a port needs to change state, something has to happen, based on the following rules:

Key Topic

- For interfaces that stay in the same STP state, nothing needs to change.
- For interfaces that need to move from a forwarding state to a blocking state, the switch immediately changes the state to blocking.
- For interfaces that need to move from a blocking state to a forwarding state, the switch first moves the interface to listening state, then learning state, each for the time specified by the forward delay timer (default 15 seconds). Only then is the interface placed into forwarding state.

Because the transition from blocking to forwarding does require some extra steps, you should be ready to respond to conceptual questions about the transition. To be ready, review the section "Reacting to State Changes That Affect the STP Topology" in Chapter 2.

Troubleshooting Layer 2 EtherChannel

EtherChannels can prove particularly challenging to troubleshoot for a couple of reasons. First, you have to be careful to match the correct configuration, and there are many more incorrect configuration combinations than there are correct combinations. Second, many interface settings must match on the physical links, both on the local switch and on the neighboring switch, before a switch will add the physical link to the channel. This second major section in the chapter works through both sets of issues.

Incorrect Options on the channel-group Command

In Chapter 3, the section titled "Configuring EtherChannel" listed the small set of working configuration options on the **channel-group** command. Those rules can be summarized as follows, for a single EtherChannel:

Key Topic

1. On the local switch, all the **channel-group** commands for all the physical interfaces must use the same channel-group number.
2. The channel-group number can be different on the neighboring switches.
3. If using the **on** keyword, you must use it on the corresponding interfaces of both switches.
4. If you use the **desirable** keyword on one switch, the switch uses PAgP; the other switch must use either **desirable** or **auto**.
5. If you use the **active** keyword on one switch, the switch uses LACP; the other switch must use either **active** or **passive**.

These rules summarize the correct configuration options, but the options actually leave many more incorrect choices. The following list shows some incorrect configurations that the switches allow, even though they would result in the EtherChannel not working. The list compares the configuration on one switch to another based on the physical interface configuration. Each lists the reasons why the configuration is incorrect.

- Configuring the **on** keyword on one switch, and **desirable**, **auto**, **active**, or **passive** on the other switch. The **on** keyword does not enable PAgP, and does not enable LACP, and the other options rely on PAgP or LACP.
- Configuring the **auto** keyword on both switches. Both use PAgP, but both wait on the other switch to begin negotiations.
- Configuring the **passive** keyword on both switches. Both use LACP, but both wait on the other switch to begin negotiations.
- Configuring the **active** keyword on one switch and either **desirable** or **auto** on the other switch. The **active** keyword uses LACP, whereas the other keywords use PAgP.
- Configuring the **desirable** keyword on one switch and either **active** or

passive on the other switch. The **desirable** keyword uses PAgP, whereas the other keywords use LACP.

Example 4-2 shows an example that matches the last item in the list. In this case, SW1's two ports (F0/14 and F0/15) have been configured with the **desirable** keyword, and SW2's matching F0/16 and F0/17 have been configured with the **active** keyword. The example lists some telling status information about the failure, with notes following the example.

Example 4-2 *Incorrect Configuration Using Mismatched PortChannel Protocols*

Click here to view code image

```
SW1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate
aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+----------------------
-------------------------
1      Po1(SD)         PAgP      Fa0/14(I)   Fa0/15(I)

SW1# show interfaces status | include Po|14|15
Port      Name               Status       Vlan       Dup
Type
Fa0/14                       connected    301        a-
full  a-100 10/100BaseTX
Fa0/15                       connected    301        a-
full  a-100 10/100BaseTX
Po1                          notconnect   unassigned   a
```

Start at the top, in the legend of the **show etherchannel summary** command.

The *D* code letter means that the channel itself is down, with *S* meaning that the channel is a Layer 2 EtherChannel. Code *I* means that the physical interface is working independently from the PortChannel (described as “stand-alone”). Then, the bottom of that command’s output highlights PortChannel 1 (Po1) as Layer 2 EtherChannel in a down state (SD), with F0/14 and F0/15 as stand-alone interfaces (I).

Interestingly, because the problem is a configuration mistake, the two physical interfaces still operate independently, as if the PortChannel did not exist. The last command in the example shows that while the PortChannel 1 interface is down, the two physical interfaces are in a connected state.

Note

As a suggestion for attacking EtherChannel problems on the exam, rather than memorizing all the incorrect configuration options, concentrate on the list of correct configuration options. Then look for any differences between a given question’s configuration as compared to the known correct configurations and work from there.

Configuration Checks Before Adding Interfaces to EtherChannels

Even when the **channel-group** commands have all been configured correctly, other configuration settings can cause problems as well. This last topic examines those configuration settings and their impact.

First, a local switch checks each new physical interface that is configured to be part of an EtherChannel, comparing each new link to the existing links. That new physical interface’s settings must be the same as the existing links’ settings; otherwise, the switch does not add the new link to the list of approved and working interfaces in the channel. That is, the physical interface remains configured as part of the PortChannel, but it is not used as part of the channel, often being placed into some nonworking state.

The list of items the switch checks includes the following:

Key Topic

- Speed
- Duplex
- Operational access or trunking state (all must be access, or all must be trunks)
- If an access port, the access VLAN

- If a trunk port, the allowed VLAN list (per the **switchport trunk allowed** command)
- If a trunk port, the native VLAN
- STP interface settings

In addition, switches check the settings on the neighboring switch. To do so, the switches either use PAgP or LACP (if already in use), or use Cisco Discovery Protocol (CDP) if using manual configuration. The neighbor must match on all parameters in this list except the STP settings.

As an example, SW1 and SW2 again use two links in one EtherChannel. Before configuring the EtherChannel, SW1's F0/15 was given a different STP port cost than F0/14. Example 4-3 picks up the story just after configuring the correct **channel-group** commands, when the switch is deciding whether to use F0/14 and F0/15 in this EtherChannel.

Example 4-3 *Local Interfaces Fail in EtherChannel Because of Mismatched STP Cost*

Click here to view code image

```
*Mar  1 23:18:56.132: %PM-4-ERR_DISABLE: channel-
misconfig (STP) error detected on
  Po1, putting Fa0/14 in err-disable state
*Mar  1 23:18:56.132: %PM-4-ERR_DISABLE: channel-
misconfig (STP) error detected on
  Po1, putting Fa0/15 in err-disable state
*Mar  1 23:18:56.132: %PM-4-ERR_DISABLE: channel-
misconfig (STP) error detected on
  Po1, putting Po1 in err-disable state
*Mar  1 23:18:58.120: %LINK-3-UPDOWN: Interface
FastEthernet0/14, changed state to
  down
*Mar  1 23:18:58.137: %LINK-3-UPDOWN: Interface Port-
channel1, changed state to down
*Mar  1 23:18:58.137: %LINK-3-UPDOWN: Interface
FastEthernet0/15, changed state to
  down

SW1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate
aggregator
```

```
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+----------------------
-------------------------
1      Po1(SD)         -         Fa0/14(D)   Fa0/15(D)
```

The messages at the top of the example specifically state what the switch does when determining whether the interface settings match. In this case, SW1 detects the different STP costs. SW1 does not use F0/14, does not use F0/15, and even places them into an err-disabled state. The switch also puts the PortChannel into err-disabled state. As a result, the PortChannel is not operational, and the physical interfaces are also not operational.

To solve this problem, you must reconfigure the physical interfaces to use the same STP settings. In addition, the PortChannel and physical interfaces must be **shutdown**, and then **no shutdown**, to recover from the err-disabled state. (Note that when a switch applies the **shutdown** and **no shutdown** commands to a PortChannel, it applies those same commands to the physical interfaces, as well; so, just do the **shutdown**/**no shutdown** on the PortChannel interface.)

Analyzing the Switch Data Plane Forwarding

STP and EtherChannel both have an impact on what a switch's forwarding logic can use. STP limits which interfaces the data plane even considers using by placing some ports in a blocking state (STP) or discarding state (RSTP), which in turn tells the data plane to simply not use that port. EtherChannel gives the data plane new ports to use in the switch's MAC address table—EtherChannels—while telling the data plane to not use the underlying physical interfaces in an EtherChannel in the MAC table.

This (short) third major section of the chapter explores the impact of STP and EtherChannel on data plane logic and a switch's MAC address table.

Predicting STP Impact on MAC Tables

Consider the small LAN shown in Figure 4-4. The LAN has only three switches, with redundancy, just big enough to make the point for this next example. The LAN supports two VLANs, 1 and 2, and the engineer has configured STP such that SW3 blocks on a different port in each of the two VLANs. As a result, VLAN 1 traffic would flow from SW3 to SW1 next, and in VLAN 2, traffic would flow from SW3 to SW2 next instead.

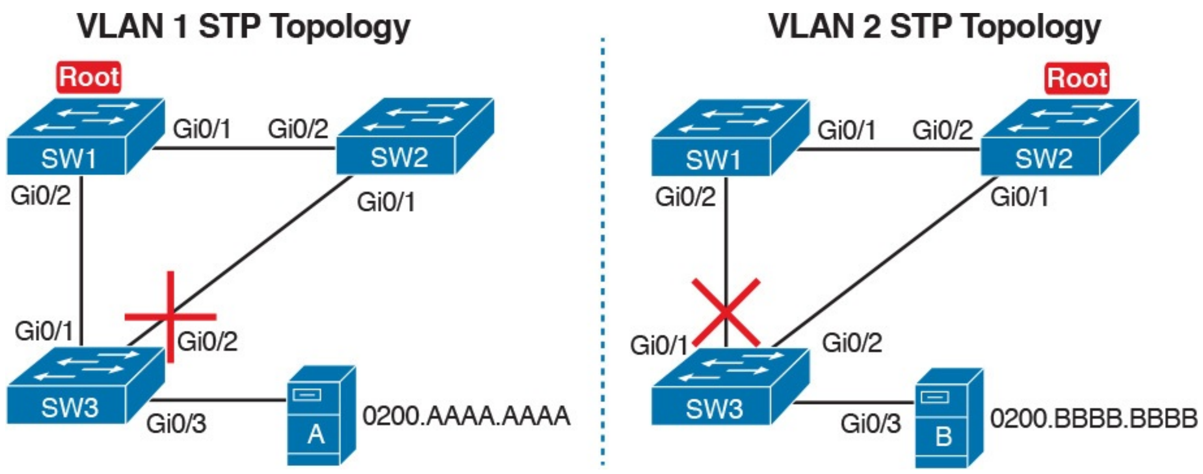


Figure 4-4 *Two Different STP Topologies for Same Physical LAN, Two Different VLANs*

Looking at diagrams like those in Figure 4-4 makes the forwarding path obvious. Although the figure shows the traffic path, that path is determined by switch MAC learning, which is then impacted by the ports on which STP has set a blocking or discarding state.

For example, consider VLAN 1's STP topology in Figure 4-4. Remember, STP blocks on a port on one switch, not on both ends of the link. So, in the case of VLAN 1, SW3's G0/2 port blocks, but SW2's G0/1 does not. Even so, by blocking on a port on one end of the link, that act effectively stops any MAC learning from happening by either device on the link. That is, SW3 learns no MAC addresses on its G0/2 port, and SW2 learns no MAC addresses on its G0/1 port, for these reasons:

- **SW2 learns no MAC addresses on G0/1:** On the blocking (SW3) end of the SW3–SW2 trunk, SW3 will not send frames out that link to SW2, so SW2 will never receive frames from which to learn MAC addresses on SW2's G0/1.
- **SW3 learns no MAC addresses on G0/2:** On the not blocking (SW2) end of the SW3–SW2 trunk, SW2 will flood frames out that port. SW3 receives those frames, but because SW3 blocks, SW3 ignores those received frames and does not learn their MAC addresses.

Given that discussion, can you predict the MAC table entries on each of the three switches for the MAC addresses of servers A and B in Figure 4-4? On switch SW2, the entry for server A, in VLAN 1, should refer to SW2's G0/2 port, pointing to SW1 next, matching the figure. But SW2's entry for server B, in VLAN 2, references SW2's G0/1 port, again matching the figure.

Example 4-4 shows the MAC tables on SW1 and SW2 as a confirmation.

Example 4-4 *Examining SW1 and SW2 Dynamic MAC Address Table Entries*

Click here to view code image

```
SW1# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  1     0200.AAAA.AAAA    DYNAMIC     Gi0/2
  2     0200.BBBB.BBBB    DYNAMIC     Gi0/1
```

```
SW2# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  1     0200.AAAA.AAAA    DYNAMIC     Gi0/2
  2     0200.BBBB.BBBB    DYNAMIC     Gi0/1
```

Predicting EtherChannel Impact on MAC Tables

Most designs use multiple links between switches, with those links configured to be part of an EtherChannel. What does that do to the MAC forwarding logic? In short, the switch uses the PortChannel interfaces, and not the physical interfaces bundled into the EtherChannel, in the MAC address table. Specifically:

MAC learning: Frames received in a physical interface that is part of a PortChannel are considered to arrive on the PortChannel interface. So, MAC learning adds the PortChannel interface rather than the physical interface to the MAC address table.

MAC forwarding: The forwarding process will find a PortChannel port as an outgoing interface when matching the MAC address table. Then the switch must take the additional step to choose the outgoing physical interface, based on the load-balancing preferences configured for that PortChannel.

For example, consider Figure 4-5, which updates previous Figure 4-4 with two-link PortChannels between each pair of switches. With VLAN 1 blocking again on switch SW3, but this time on SW3's PortChannel3 interface, what

MAC table entries would you expect to see in each switch? Similarly, what MAC table entries would you expect to see for VLAN 2, with SW3 blocking on its PortChannel2 interface?

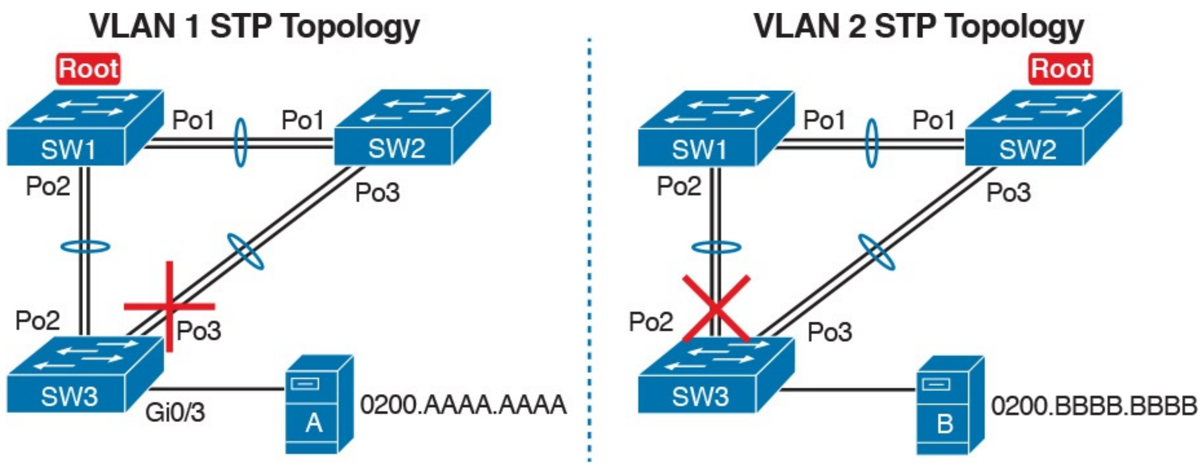


Figure 4-5 *VLAN Topology with PortChannels Between Switches*

The logic of which entries exist on which ports mirrors the logic with the earlier example surrounding Figure 4-4. In this case, the interfaces just happen to be PortChannel interfaces. Example 4-5 shows the same command from the same two switches as Example 4-4: **show mac address-table dynamic** from both SW1 and SW2. (Note that to save length, the MAC table output shows only the entries for the two servers in Figure 4-5.)

Example 4-5 *SW1 and SW2 MAC Tables with PortChannel Ports Listed*

Click here to view code image

```
SW1# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0200.AAAA.AAAA    DYNAMIC     Po2
   2    0200.BBBB.BBBB    DYNAMIC     Po1
```

```
SW2# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0200.AAAA.AAAA    DYNAMIC     Po1
   2    0200.BBBB.BBBB    DYNAMIC     Po3
```

Switches use one of many load-balancing options to then choose the physical interface to use after matching MAC table entries like those shown in Example 4-5. By default, Cisco Layer 2 switches often default to use a balancing method based on the source MAC address. In particular, the switch looks at the low-order bits of the source MAC address (which are on the far right of the MAC address in written form). This approach increases the chances that the balancing will be spread somewhat evenly based on the source MAC addresses in use.

Choosing the VLAN of Incoming Frames

To wrap up the analysis of switch data plane forwarding, this section mostly reviews topics already discussed, but it serves to emphasize some important points. The topic is simply this: How does a switch know which VLAN a frame is a part of as the frame enters a switch? You have seen all the information needed to answer this question already, but take the time to review.

First, some interfaces trunk, and in those cases, the frame arrives with a VLAN ID listed in the incoming trunking header. In other cases, the frame does not arrive with a trunking header, and the switch must look at local configuration. But because the switch will match both the destination MAC address and the frame VLAN ID when matching the MAC address table, knowing how the switch determines the VLAN ID is important.

The following list reviews and summarizes the key points of how a switch determines the VLAN ID to associate with an incoming frame:

Key Topic

Step 1. If the port is an access port, associate the frame with the configured access VLAN (**switchport access vlan** *vlan_id*).

Step 2. If the port is a voice port, or has both an IP Phone and PC (or other data device) connected to the phone:

- **A.** Associate the frames from the data device with the configured access VLAN (as configured with the **switchport access vlan** *vlan_id* command).
- **B.** Associate the frames from the phone with the VLAN ID in the 802.1Q header (as configured with the **switchport voice vlan** *vlan_id* command).

Step 3. If the port is a trunk, determine the frame's tagged VLAN, or if there is no tag, use that incoming interface's native VLAN ID (**switchport trunk native** *vlan_id*).

Troubleshooting VLANs and VLAN Trunks

A switch's data plane forwarding processes depend in part on VLANs and VLAN trunking. Before a switch can forward frames in a particular VLAN, the switch must know about a VLAN and the VLAN must be active. And before a switch can forward a frame over a VLAN trunk, the trunk must currently allow that VLAN to pass over the trunk.

This final major section in this chapter focuses on VLAN and VLAN trunking issues, specifically issues that impact the frame switching process. The issues are as follows:

Step 1. Identify all access interfaces and their assigned access VLANs and reassign into the correct VLANs if incorrect.

Step 2. Determine whether the VLANs both exist (either configured or learned with the VLAN Trunking Protocol [VTP]) and are active on each switch. If not, configure and activate the VLANs to resolve problems as needed.

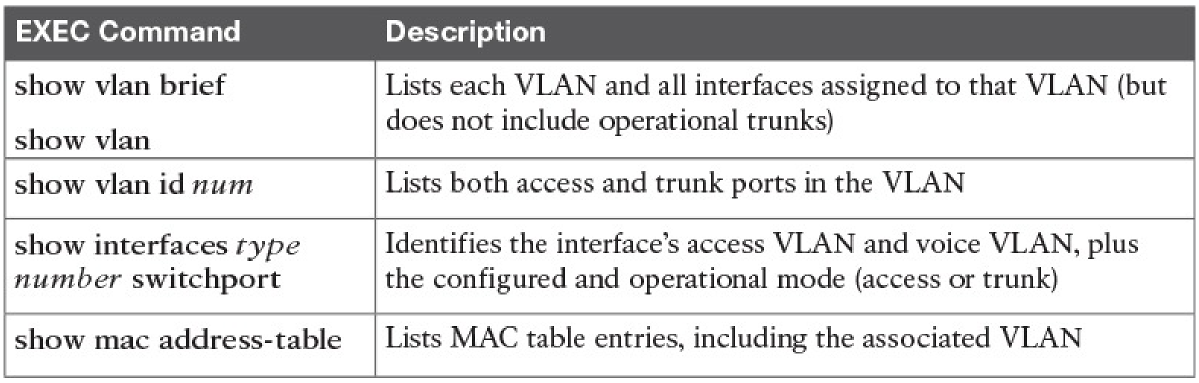
Step 3. Check the allowed VLAN lists, on the switches on both ends of the trunk, and ensure that the lists of allowed VLANs are the same.

Step 4. Check for incorrect configuration settings that result in one switch operating as a trunk, with the neighboring switch not operating as a trunk.

Step 5. Check the allowed VLANs on each trunk, to make sure that the trunk has not administratively removed a VLAN from being supported on a trunk.

Access VLAN Configuration Incorrect

To ensure that each access interface has been assigned to the correct VLAN, engineers simply need to determine which switch interfaces are access interfaces instead of trunk interfaces, determine the assigned access VLANs on each interface, and compare the information to the documentation. The **show** commands listed in Table 4-1 can be particularly helpful in this process.



EXEC Command	Description
show vlan brief show vlan	Lists each VLAN and all interfaces assigned to that VLAN (but does not include operational trunks)
show vlan id *num*	Lists both access and trunk ports in the VLAN
show interfaces *type number* switchport	Identifies the interface's access VLAN and voice VLAN, plus the configured and operational mode (access or trunk)
show mac address-table	Lists MAC table entries, including the associated VLAN

Table 4-1 Commands That Can Find Access Ports and VLANs

If possible, start this step with the **show vlan** and **show vlan brief** commands, because they list all the known VLANs and the access interfaces assigned to each VLAN. Be aware, however, that these two commands do not list operational trunks. The output does list all other interfaces (those not currently trunking), no matter whether the interface is in a working or nonworking state.

If the **show vlan** and **show interface switchport** commands are not available in a particular exam question, the **show mac address-table** command can also help identify the access VLAN. This command lists the MAC address table, with each entry including a MAC address, interface, and VLAN ID. If the exam question implies that a switch interface connects to a single device, you should only see one MAC table entry that lists that particular access interface; the VLAN ID listed for that same entry identifies the access VLAN. (You cannot make such assumptions for trunking interfaces.)

After you determine the access interfaces and associated VLANs, if the interface is assigned to the wrong VLAN, use the **switchport access vlan** *vlan-id* interface subcommand to assign the correct VLAN ID.

Access VLANs Undefined or Disabled

Switches do not forward frames for VLANs that are (a) not known because the VLAN is not configured or has not been learned with VTP or (b) the VLAN is known, but it is disabled (shut down). This section summarizes the best ways to confirm that a switch knows that a particular VLAN exists, and if it exists, determines the shutdown state of the VLAN.

First, on the issue of whether a VLAN exists on a switch, a VLAN can be defined to a switch in two ways: using the **vlan** *number* global configuration command, or it can be learned from another switch using VTP. Chapter 5, "VLAN Trunking Protocol," discusses VTP and how VTP can be used by a switch to learn about VLANs. For this discussion, consider that the only way for a switch to know about a VLAN is to have a **vlan** command configured on the local switch.

Next, the **show vlan** command always lists all VLANs known to the switch, but the **show running-config** command does not. Switches configured as VTP servers and clients do not list the **vlan** commands in the running-config file nor the startup-config file; on these switches, you must use the **show vlan** command. Switches configured to use VTP transparent mode, or that disable VTP, list the **vlan** configuration commands in the configuration files. (Use the **show vtp status** command to learn the current VTP mode of a switch.)

After you determine that a VLAN does not exist on a switch, the problem might be that the VLAN simply needs to be configured. If so, follow the VLAN configuration process as covered in detail in Chapter 1.

Even for existing VLANs, you must also verify whether the VLAN is active. The **show vlan** command should list one of two VLAN state values, depending on the current state: either *active* or *act/lshut*. The second of these states means that the VLAN is shut down. Shutting down a VLAN disables the VLAN on that switch only, so that *the switch will not forward frames in that VLAN*.

Switch IOS gives you two similar configuration methods with which to disable (**shutdown**) and enable (**no shutdown**) a VLAN. Example 4-6 shows how, first by using the global command [**no**] **shutdown vlan** *number* and then using the VLAN mode subcommand [**no**] **shutdown**. The example shows the global commands enabling and disabling VLANs 10 and 20, respectively, and using VLAN subcommands to enable and disable VLANs 30 and 40 (respectively).

Example 4-6 *Enabling and Disabling VLANs on a Switch*

Click here to view code image

```
SW2# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
------------------------
1    default                          active    Fa0/1,
Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5,
Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9,
Fa0/10, Fa0/11, Fa0/12
                                                Fa0/14,
Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18,
Fa0/19, Fa0/20, Fa0/21
```

```
                                                Fa0/22,
Fa0/23, Fa0/24, Gi0/1
10   VLAN0010                         act/lshut Fa0/13
20   VLAN0020                         active
30   VLAN0030                         act/lshut
40   VLAN0040                         active
SW2# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW2(config)# no shutdown vlan 10
SW2(config)# shutdown vlan 20
SW2(config)# vlan 30
SW2(config-vlan)# no shutdown
SW2(config-vlan)# vlan 40
SW2(config-vlan)# shutdown
SW2(config-vlan)#
```

Mismatched Trunking Operational States

Trunking can be configured correctly so that both switches forward frames for the same set of VLANs. However, trunks can also be misconfigured, with a couple of different results. In some cases, both switches conclude that their interfaces do not trunk. In other cases, one switch believes that its interface is correctly trunking, while the other switch does not.

The most common incorrect configuration—which results in both switches not trunking—is a configuration that uses the **switchport mode dynamic auto** command on both switches on the link. The word "auto" just makes us all want to think that the link would trunk automatically, but this command is both automatic and passive. As a result, both switches passively wait on the other device on the link to begin negotiations.

With this particular incorrect configuration, the **show interfaces switchport** command on both switches confirms both the administrative state (auto) and the fact that both switches operate as "static access" ports. Example 4-7 highlights those parts of the output from this command.

Example 4-7 *Operational Trunking State*

Click here to view code image

```
SW2# show interfaces gigabit0/2 switchport
Name: Gi0/2
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
```

```
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
! lines omitted for brevity
```

A different incorrect trunking configuration results in one switch with an operational state of "trunk," while the other switch has an operational state of "static access." When this combination of events happens, the interface works a little. The status on each end will be up/up or connected. Traffic in the native VLAN will actually cross the link successfully. However, traffic in all the rest of the VLANs will not cross the link.

Figure 4-6 shows the incorrect configuration along with which side trunks and which does not. The side that trunks (SW1 in this case) enables trunking always, using the command **switchport mode trunk**. However, this command does not disable Dynamic Trunking Protocol (DTP) negotiations. To cause this particular problem, SW1 also disables DTP negotiation using the **switchport nonegotiate** command. SW2's configuration also helps create the problem, by using a trunking option that relies on DTP. Because SW1 has disabled DTP, SW2's DTP negotiations fail, and SW2 does not trunk.

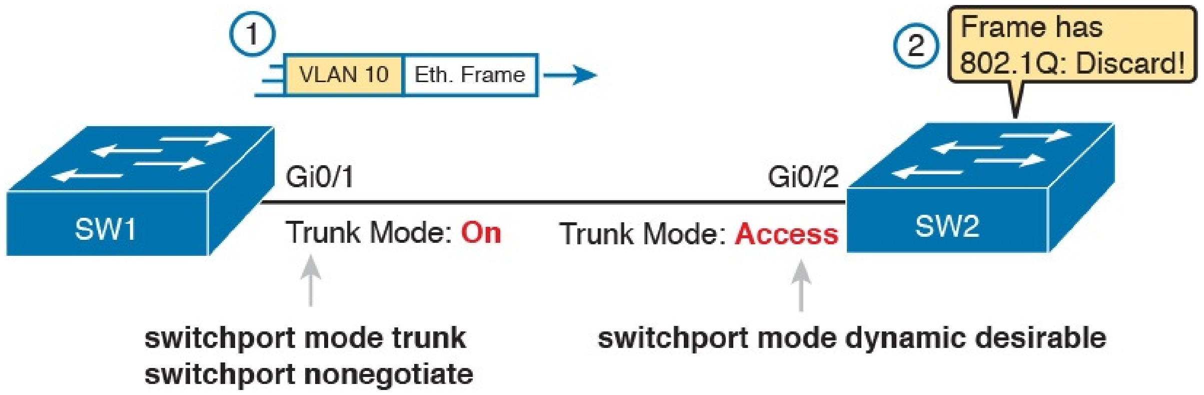


Figure 4-6 *Mismatched Trunking Operational States*

In this case, SW1 treats its G0/1 interface as a trunk, and SW2 treats its G0/2 interface as an access port (not a trunk). As shown in the figure at Step 1, SW1 could (for example) forward a frame in VLAN 10. However, SW2 would view any frame that arrives with an 802.1Q header as illegal, because SW2 treats its G0/2 port as an access port. So, SW2 discards any 802.1Q frames received on that port.

To deal with the possibility of this problem, always check the trunk's operational state on both sides of the trunk. The best commands to check trunking-related facts are **show interfaces trunk** and **show interfaces switchport**.

Note

Frankly, in real life, just avoid this kind of configuration. However, the switches do not prevent you from making these types of mistakes, so you need to be ready. Note that Chapter 1's Table 1-3 summarizes the list of options on the **switchport trunk** command, which combinations work, and which ones to completely avoid (like the combination shown here in Figure 4-6.)

Mismatched Supported VLAN List on Trunks

VLAN trunks on Cisco switches can forward traffic for all defined and active VLANs. However, a particular trunk may not forward traffic for a defined and active VLAN for a variety of other reasons. You should know how to identify which VLANs a particular trunk port currently supports, and the reasons why the switch might not be forwarding frames for a VLAN on that trunk port.

The first category in this step can be easily done using the **show interfaces trunk** command, which only lists information about currently operational trunks. The best place to begin with this command is the last section of output, which lists the VLANs whose traffic will be forwarded over the trunk. Any VLANs that make it to this final list of VLANs in the command output meet the following criteria:

- The VLAN exists and is active on the local switch (as seen in the **show vlan** command).
- The VLAN has not been removed from the *allowed VLAN list* on the trunk (as configured with the **switchport trunk allowed vlan** interface subcommand).
- The VLAN has not been VTP-pruned from the trunk. (This is a VTP feature, discussed in Chapter 5, which this section will now otherwise ignore, deferring discussion until Chapter 5. It is only listed here because the **show** command output mentions it.)
- The trunk is in an STP forwarding state in that VLAN (as also seen in the **show spanning-tree vlan** *vlan-id* command).

Example 4-8 shows a sample of the command output from the **show interfaces trunk** command, with the final section of the command output shaded. In this case, the trunk only forwards traffic in VLANs 1 and 4.

Example 4-8 *Allowed VLAN List and List of Active VLANs*

Click here to view code image

```
SW1# show interfaces trunk

Port          Mode          Encapsulation  Status        Na
vlan
Gi0/1         desirable     802.1q         trunking      1

Port          Vlans allowed on trunk
Gi0/1         1-2,4-4094

Port          Vlans allowed and active in management
domain
Gi0/1         1,4

Port          Vlans in spanning tree forwarding state and
not pruned
Gi0/1         1,4
```

The absence of a VLAN in this last part of the command's output does not necessarily mean that a problem has occurred. In fact, a VLAN might be legitimately excluded from a trunk for any of the reasons in the list just before Example 4-8. However, for a given exam question, it can be useful to know why traffic for a VLAN will not be forwarded over a trunk, and the details inside the output identify the specific reasons.

The output of the **show interfaces trunk** command creates three separate lists of VLANs, each under a separate heading. These three lists show a progression of reasons why a VLAN is not forwarded over a trunk. Table 4-2 summarizes the headings that precede each list and the reasons why a switch chooses to include or not include a VLAN in each list.

Key Topic

List Position	Heading	Reasons
First	VLANs allowed	VLANs 1–4094, minus those removed by the **switchport trunk allowed** command
Second	VLANs allowed and active...	The first list, minus VLANs not defined to the local switch (that is, there is not a **vlan** global configuration command or the switch has not learned of the VLAN with VTP), and also minus those VLANs in shutdown mode
Third	VLANs in spanning tree...	The second list, minus VLANs in an STP blocking state for that interface, and minus VLANs VTP pruned from that trunk

Table 4-2 VLAN Lists in the **show interfaces trunk** Command

Mismatched Native VLAN on a Trunk

Closing with a brief mention of one other trunking topic, you should also check a trunk's native VLAN configuration at this step. Unfortunately, it is possible to set the native VLAN ID to different VLANs on either end of the trunk, using the **switchport trunk native vlan** *vlan-id* command. If the native VLANs differ according to the two neighboring switches, the switches will accidentally cause frames to leave one VLAN and enter another.

For example, if switch SW1 sends a frame using native VLAN 1 on an 802.1Q trunk, SW1 does not add a VLAN header, as is normal for the native VLAN. When switch SW2 receives the frame, noticing that no 802.1Q header exists, SW2 assumes that the frame is part of SW2's configured native VLAN. If SW2 has been configured to think VLAN 2 is the native VLAN on that trunk, SW2 will try to forward the received frame into VLAN 2.

Chapter Review

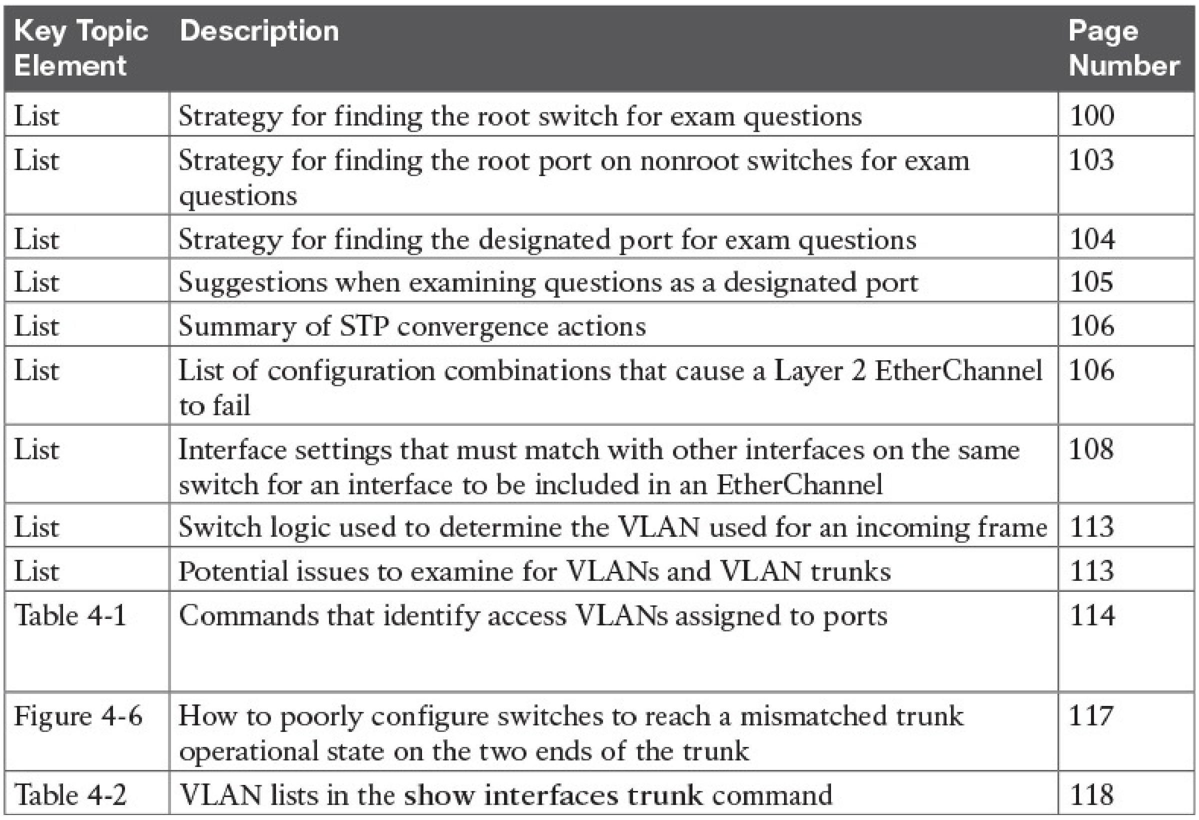
One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book, DVD, or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 4-3 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Review Element	Review Date(s)	Resource Used
Review key topics		Book, DVD/website
Review memory tables		Book, DVD/website

Table 4-3 Chapter Review Tracking

Review All the Key Topics

Key Topic



Key Topic Element	Description	Page Number
List	Strategy for finding the root switch for exam questions	100
List	Strategy for finding the root port on nonroot switches for exam questions	103
List	Strategy for finding the designated port for exam questions	104
List	Suggestions when examining questions as a designated port	105
List	Summary of STP convergence actions	106
List	List of configuration combinations that cause a Layer 2 EtherChannel to fail	106
List	Interface settings that must match with other interfaces on the same switch for an interface to be included in an EtherChannel	108
List	Switch logic used to determine the VLAN used for an incoming frame	113
List	Potential issues to examine for VLANs and VLAN trunks	113
Table 4-1	Commands that identify access VLANs assigned to ports	114
Figure 4-6	How to poorly configure switches to reach a mismatched trunk operational state on the two ends of the trunk	117
Table 4-2	VLAN lists in the **show interfaces trunk** command	118

Table 4-4 Key Topics for Chapter 4

Command References

Although this chapter does show several examples, all the commands were introduced earlier in Chapters 1 and 3, so this section does not include any command reference tables. Refer to those chapters for command reference tables.

Chapter 5. VLAN Trunking Protocol

This chapter covers the following exam topics:

1.0 LAN Switching Technologies

1.2 Configure, verify, and troubleshoot interswitch connectivity

1.2.a DTP and VTP (v1&v2)

Engineers sometimes have a love/hate relationship with VLAN Trunking Protocol (VTP). VTP serves a useful purpose, distributing the configuration of the [**no**] **vlan** *vlan-id* command among switches. As a result, the engineer configures the **vlan** command on one switch, and all the rest of the switches are automatically configured with that same command.

Unfortunately, the automated update powers of VTP can also be dangerous. For example, an engineer could delete a VLAN on one switch, not realizing that the command actually deleted the VLAN on all switches. And deleting a VLAN impacts a switch's forwarding logic: Switches do not forward frames for VLANs that are not defined to the switch.

This chapter discusses VTP, from concept through troubleshooting. The first major section discusses VTP concepts, while the second section shows how to configure and verify VTP. The third section walks through troubleshooting, with some discussion of the risks that cause some engineers to just not use VTP. (In fact, the entirety of the ICND1 Cert Guide's discussion of VLAN configuration assumes VTP uses the VTP transparent mode, which effectively disables VTP from learning and advertising VLAN configuration.)

As for exam topics, note that the Cisco exam topics that mention VTP also mention DTP. Chapter 1, "Implementing Ethernet Virtual LANs," discussed how Dynamic Trunking Protocol (DTP) is used to negotiate VLAN trunking. This chapter does not discuss DTP, leaving that topic for Chapter 1.

"Do I Know This Already?" Quiz

Take the quiz (either here, or use the PCPT software) if you want to use the score to help you decide how much time to spend on this chapter. The answers are at the bottom of the page following the quiz, and the explanations are in DVD Appendix C and in the PCPT software.

Foundation Topics Section	Questions
VLAN Trunking Protocol (VTP) Concepts	1, 2
VTP Configuration and Verification	3, 4
VTP Troubleshooting	5, 6

Table 5-1 "Do I Know This Already?" Foundation Topics Section-to-

Question Mapping

1. Which of the following VTP modes allow VLANs to be configured on a switch? (Choose two answers.)
   a. Client
   b. Server
   c. Transparent
   d. Dynamic

2. An engineer plans to connect three switches (SW1, SW2, and SW3) in a lab. Before connecting the switches, he starts by configuring all three switches as VTP servers, with matching VTP domain name and password. He then configures some VLANs on each switch so that switch SW3 has a revision number of 10, switch SW2 has a revision number of 6, and switch SW1 has a revision number of 8. Only then does the engineer connect the switches with trunks: first SW1 to SW2, then SW2 to SW3, and then SW3 to SW1. Switch SW1 is elected the STP root switch in VLAN 1. Which answer most accurately states which VLAN configuration database is used, and why?
   a. All use switch SW1's database because it has the highest revision number between the first two connected switches.
   b. All use switch SW1's database because VTP uses the same election logic as STP.
   c. All use SW3's database because SW3 has the highest revision number.
   d. All use SW2's database because SW2 has the lowest revision number.

3. An engineer compares the output of the **show vtp status** command on two neighboring switches. One switch, SW1, acts as VTP server, while the other, SW2, acts as a VTP client. What items in the command output confirm that synchronization has completed? (Choose two answers.)
   a. Both list the same "last updater" IP address and timestamp.
   b. Both list the neighbor's MAC address and the word "synchronized."
   c. SW2 (the client) lists the phrase "synchronized with server."
   d. Both list the same configuration revision number.

4. Switches SW1, SW2, SW3, and SW4 are configured as VTP server, client, transparent, and off, respectively, all using VTP version 1. A junior engineer has been told to try to configure the following two commands on each switch directly from the CLI: **vlan 200** and **vlan 2000**. Which answers correctly state which commands will be rejected,

on which switch? (Choose two answers.)

a. vlan 2000 will be rejected on SW1 (VTP server).

b. vlan 200 will be rejected on SW2 (VTP client).

c. vlan 200 will be rejected on SW3 (VTP transparent).

d. vlan 200 will be rejected on SW1 (VTP server).

5. Two neighboring LAN switches are connected with an operational 802.1Q trunk. Switch SW1 has been configured with the **vtp mode client**, **vtp domain fred**, and **vtp version 2** commands. SW1 has no other VTP configuration commands configured. Which answer lists a possible reason why switch SW2, on the other end of the trunk, is not synchronizing its VLAN database with switch SW1? (Choose two answers.)

a. SW2 has a **vtp version 1** command configured.

b. SW2 has a **vtp password G0BeeZ** command configured.

c. SW2 has a **vtp domain Fred** command configured.

d. SW2 has a **vtp mode client** command configured.

6. Switches SW1 and SW2 connect through an operational trunk. The engineer wants to use VTP to communicate VLAN configuration changes. The engineer configures a new VLAN on SW1, VLAN 44, but SW2 does not learn about the new VLAN. Which of the following configuration settings on SW1 and SW2 would be a potential root cause why SW2 does not learn about VLAN 44? (Choose two answers.)

a. VTP domain names of larry and LARRY, respectively

b. VTP passwords of bob and BOB, respectively

c. VTP pruning enabled and disabled, respectively

d. VTP modes of server and client, respectively

Answers to the "Do I Know This Already?" quiz:

1 B, C **2** C **3** A, D **4** A, B **5** B, C **6** A, B

Foundation Topics

VLAN Trunking Protocol (VTP) Concepts

The Cisco-proprietary *VLAN Trunking Protocol* (VTP) provides a means by which Cisco switches can exchange VLAN configuration information. In particular, VTP advertises about the existence of each VLAN based on its VLAN ID and the VLAN name.

This first major section of the chapter discusses the major features of VTP in concept, in preparation for the VTP implementation (second section) and VTP troubleshooting (third section).

Basic VTP Operation

Think for a moment about what has to happen in a small network of four switches when you need to add two new hosts, and to put those hosts in a new VLAN that did not exist before. Figure 5-1 shows some of the main configuration concepts.

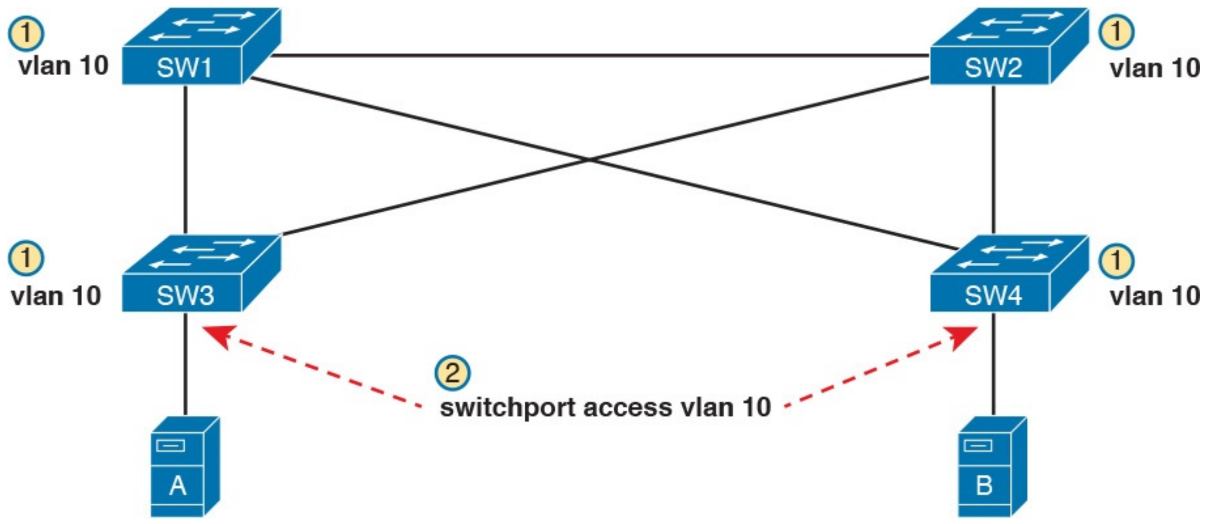


Figure 5-1 *Commands to Add Support for VLAN 10 in a Sample LAN*

First, remember that for a switch to be able to forward frames in a VLAN, that VLAN must be defined on that switch. In this case, Step 1 shows the independent configuration of VLAN 10 on the four switches: the two distribution switches and the two access layer switches. With the rules discussed in Chapter 1 (which assumed VTP transparent mode, by the way), all four switches need to be configured with the **vlan 10** command.

Step 2 shows the additional step to configure each access port to be in VLAN 10 as per the design. That is, in addition to creating the VLAN, the individual ports need to be added to the VLAN, as shown for servers A and B with the **switchport access vlan 10** command.

VTP, when used for its intended purpose, would allow the engineer to create the VLAN (the **vlan 10** command) on one switch only, with VTP then automatically updating the configuration of the other switches.

VTP defines a Layer 2 messaging protocol that the switches can use to exchange VLAN configuration information. When a switch changes its VLAN configuration—including the **vlan** *vlan-id* command—VTP causes all the switches to synchronize their VLAN configuration to include the same VLAN IDs and VLAN names. The process is somewhat like a routing protocol, with each switch sending periodic VTP messages. However, routing protocols advertise information about the IP network, whereas VTP advertises VLAN configuration.

Figure 5-2 shows one example of how VTP works in the same scenario used for Figure 5-1. Figure 5-2 starts with the need for a new VLAN 10, and two servers to be added to that VLAN. At Step 1, the network engineer creates the VLAN with the **vlan 10** command on switch SW1. SW1 then uses VTP to advertise that new VLAN configuration to the other switches, as shown at Step 2; note that the other three switches do not need to be configured with the **vlan 10** command. At Step 3, the network engineer still must configure the access ports with the **switchport access vlan 10** command, because VTP does not advertise the interface and access VLAN configuration.

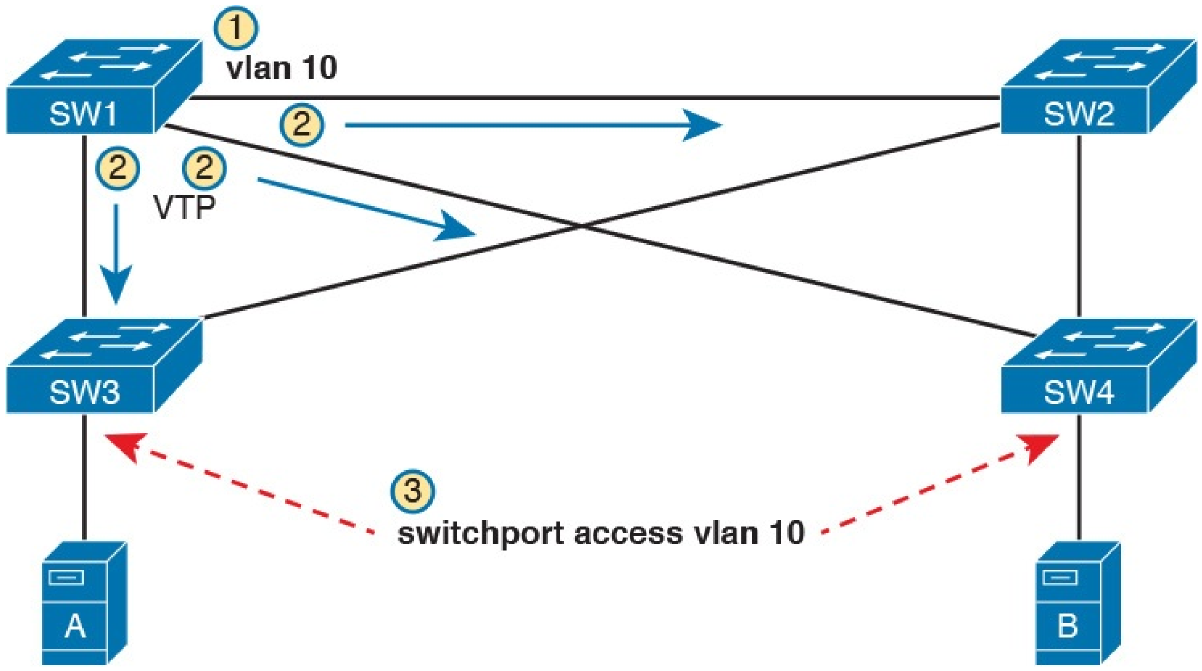


Figure 5-2 *Distributing the* **vlan 10** *Command with VTP*

VTP advertises the **vlan** *vlan-id* command, the **name** *vlan-name* subcommand, and several VTP-specific commands. Of particular importance, note that VTP does not advertise the command that associates an access port with a particular VLAN (**switchport access vlan** *vlan-id*), so those still need to be configured on the individual switches.

Also, for historical reasons, VTP limits VTP servers and clients to use VLANs 1 through 1005. This range of VLAN IDs is known as standard range VLANs, and includes VLAN 1, which is the default access VLAN and the default native VLAN on each port. The standard range ends with four reserved VLANs, 1002–1005, which are reserved for historical reasons. VTP servers can then configure any of the other standard range VLAN IDs (2–1001) and advertise those.

Note that switches in VTP transparent mode, or a switch that has disabled VTP, can configure and use extended range VLANs, which in IOS switches extends the VLAN ID range up to 4094.

Synchronizing the VTP Database

To use VTP to announce and/or learn VLAN configuration information, a switch must use either VTP *server mode* or *client mode*. The third VTP mode, *transparent mode*, tells a switch to not learn VLAN configuration and to not advertise VLAN configuration, effectively making a VTP transparent mode switch act as if it were not there, at least for the purposes of VTP. This next topic works through the mechanisms used by switches acting as either VTP server or client.

VTP servers allow the network engineer to create VLANs (and other related commands) from the CLI, whereas VTP clients do not allow the network engineer to create VLANs. You have seen many instances of the **vlan** *vlan-id* command at this point in your study, the command that creates a new VLAN in a switch. VTP servers are allowed to continue to use this command to create VLANs, but switches placed in VTP client mode reject the **vlan** *vlan-id* command, because VTP client switches cannot create VLANs.

With that main difference in mind, VTP servers allow the creation of VLANs (and related configuration) via the usual commands. The server then advertises that configuration information over VLAN trunks. The overall flow works something like this:

1. For each trunk, send VTP messages, and listen to receive them.
2. Check my local VTP parameters versus the VTP parameters announced in the VTP messages received on a trunk.
3. If the VTP parameters match, attempt to synchronize the VLAN configuration databases between the two switches.

> **Note**
>
> The name *VLAN Trunking Protocol* is based on the fact that this protocol works specifically over VLAN trunks, as noted in item 1 in this list.

Done correctly, VTP causes all the switches in the same administrative *VTP domain*—the set of switches with the same domain name and password—to converge to have the exact same configuration of VLAN information. Over time, each time the VLAN configuration is changed on any VTP server, all other switches in the VTP automatically learn of those configuration changes.

VTP does not think of the VLAN configuration as lots of small pieces of information, but rather as one *VLAN configuration database*. The configuration database has a configuration revision number which is incremented by 1 each time a server changes the VLAN configuration. The VTP synchronization process hinges on the idea of making sure each switch uses the VLAN configuration database that has the best (highest) revision number.

Figure 5-3 begins an example that demonstrates how the VLAN configuration database revision numbers work. At the beginning of the example, all the switches have converged to use the VLAN database that uses revision number 3. The example then shows:

1. The network engineer defines a new VLAN with the **vlan 10** command on switch SW1.
2. SW1, a VTP server, changes the VTP revision number for its own VLAN configuration database from 3 to 4.
3. SW1 sends VTP messages over the VLAN trunk to SW2 to begin the process of telling SW2 about the new VTP revision number for the VLAN configuration database.

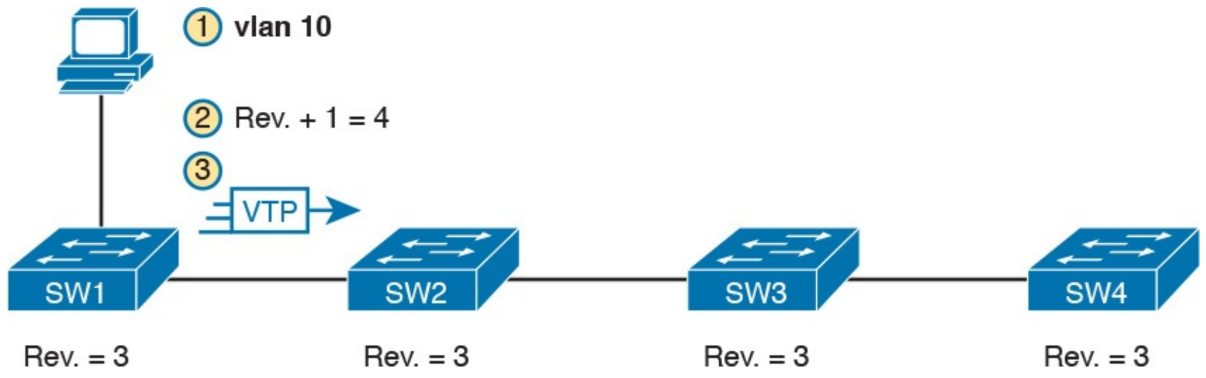


Figure 5-3 *Adding VLAN 10 at Switch SW1, Starting with All at Revision Number 3*

At this point, only switch SW1 has the best VLAN configuration database with the highest revision number (4). Figure 5-4 shows the next few steps, picking up the process where Figure 5-3 stopped. Upon receiving the VTP messages from SW1, as shown in Step 3 of Figure 5-3, at Step 4 in Figure 5-4, SW2 starts using that new LAN database. Step 5 emphasizes the fact that as a result, SW2 now knows about VLAN 10. SW2 then sends VTP messages over the trunk to the next switch, SW3 (Step 6).

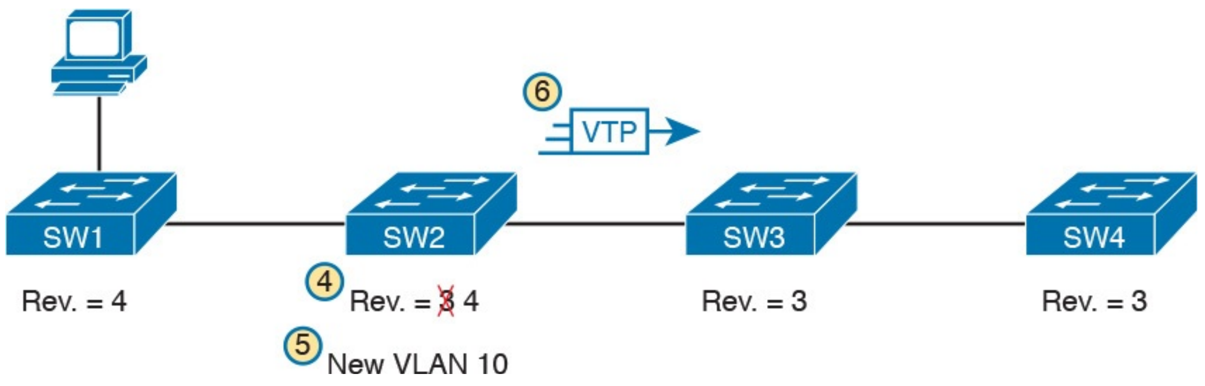


Figure 5-4 *Adding VLAN 10 at Switch SW1, Starting with All at Revision Number 3*

With VTP working correctly on all four switches, all the switches will eventually use the exact same configuration, with VTP revision number 4, as advertised with VTP.

Figure 5-4 also shows a great example of one key similarity between VTP clients and servers: both will learn and update their VLAN database from VTP messages received from another switch. Note that the process shown in Figures 5-3 and 5-4 works the same whether switches SW2, SW3, and SW4 are VTP clients or servers, in any combination. In this scenario, the only switch that must be a VTP server is switch SW1, where the **vlan 10** command was configured; a VTP client would have rejected the command.

For instance, in Figure 5-5, imagine switches SW2 and SW4 were VTP clients, but switch SW3 was a VTP server. With the same scenario discussed in Figures 5-3 and 5-4, the new VLAN configuration database is propagated just as described in those earlier figures, with SW2 (client), SW3 (server), and SW4 (client) all learning of and using the new database with revision number 4.

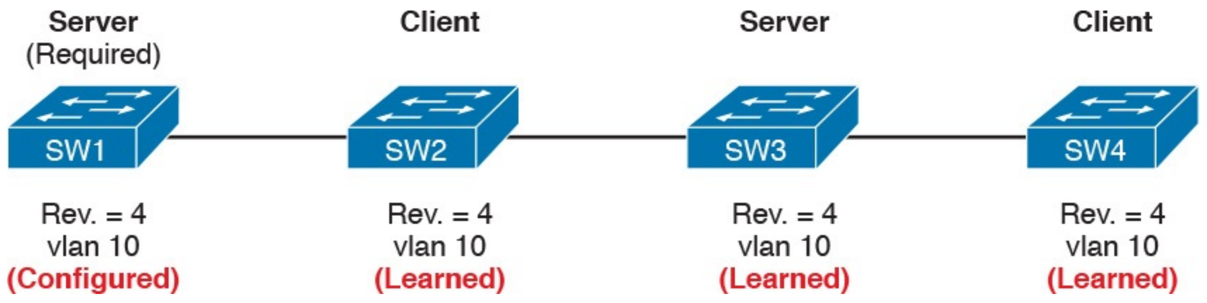


Figure 5-5 *Stable State, VTP Revision Number 4, All Switches Know VLAN 10*

Note

The complete process by which a server changes the VLAN configuration and all VTP switches learn the new configuration, resulting in all switches knowing the same VLAN IDs and name, is called *VTP synchronization*.

After VTP synchronization is completed, VTP servers and clients also send periodic VTP messages every 5 minutes. If nothing changes, the messages keep listing the same VLAN database revision number, and no changes occur. Then when the configuration changes in one of the VTP servers, that switch increments its VTP revision number by 1, and its next VTP messages announce a new VTP revision number, so that the entire VTP domain (clients and servers) synchronize to use the new VLAN database.

Requirements for VTP to Work Between Two Switches

When a VTP client or server connects to another VTP client or server switch, Cisco IOS requires that the following three facts be true before the two switches will process VTP messages received from the neighboring switch:

- The link between the switches must be operating as a VLAN trunk (ISL or 802.1Q).
- The two switches' case-sensitive VTP domain name must match.
- If configured on at least one of the switches, both switches must have configured the same case-sensitive VTP password.

The VTP domain name provides a design tool by which engineers can create multiple groups of VTP switches, called *VTP domains*, whose VLAN configurations are autonomous. To do so, the engineer can configure one set of switches in one VTP domain and another set in another VTP domain. Switches in one domain will ignore VTP messages from switches in the other domain, and vice versa.

The VTP password mechanism provides a means by which a switch can prevent malicious attackers from forcing a switch to change its VLAN configuration. The password itself is never transmitted in clear text.

VTP Version 1 Versus Version 2

Cisco supports three VTP versions, aptly numbered versions 1, 2, and 3. Interestingly, the current ICND2/CCNA exam topics mention versions 1 and 2 specifically, but omit version 3. Version 3 adds more capabilities and features beyond versions 1 and 2, and as a result is a little more complex. Versions 1 and 2 are relatively similar, with version 2 updating version 1 to provide some specific feature updates. For example, version 2 added support for a type of LAN called Token Ring, but Token Ring is no longer even found in Cisco's product line.

For the purposes of configuring, verifying, and troubleshooting VTP today, versions 1 and 2 have no meaningful difference. For instance, two switches can be configured as VTP servers, one using VTP version 1 and one using VTP version 2, and they do exchange VTP messages and learn from each other.

The one difference between VTP versions 1 and 2 that might matter has to do with the behavior of a VTP transparent mode switch. By design, VTP transparent mode is meant to allow a switch to be configured to not synchronize with other switches, but to also pass the VTP messages to VTP servers and clients. That is, the transparent mode switch is transparent to the intended purpose of VTP: servers and clients synchronizing. One of the requirements for transparent mode switches to forward the VTP messages sent by servers and clients is that the VTP versions must match.

VTP Pruning

By default, Cisco IOS on LAN switches allows frames in all configured VLANs to be passed over a trunk. Switches flood broadcasts (and unknown destination unicasts) in each active VLAN out these trunks.

However, using VTP can cause too much flooded traffic to flow into parts of the network. VTP advertises any new VLAN configured in a VTP server to the other server and client switches in the VTP domain. However, when it comes to frame forwarding, there may not be any need to flood frames to all switches, because some switches may not connect to devices in a particular VLAN. For example, in a campus LAN with 100 switches, all the devices in VLAN 50 may exist on only 3 to 4 switches. However, if VTP advertises VLAN 50 to all the switches, a broadcast in VLAN 50 could be flooded to all 100 switches.

One solution to manage the flow of broadcasts is to manually configure the allowed VLAN lists on the various VLAN trunks. However, doing so requires a manual configuration process. A better option might be to allow VTP to dynamically determine which switches do not have access ports in each VLAN, and prune (remove) those VLANs from the appropriate trunks to limit flooding. *VTP pruning* simply means that the appropriate switch trunk interfaces do not flood frames in that VLAN.

Note

The section “Mismatched Supported VLAN List on Trunks” in Chapter 4, “LAN Troubleshooting,” discusses the various reasons why a switch trunk does not forward frames in a VLAN, including the allowed VLAN list. That section also briefly references VTP pruning.

Figure 5-6 shows an example of VTP pruning, showing a design that makes the VTP pruning feature more obvious. In this figure, two VLANs are used: 10 and 20. However, only switch SW1 has access ports in VLAN 10, and only switches SW2 and SW3 have access ports in VLAN 20. With this design, a frame in VLAN 20 does not need to be flooded to the left to switch SW1, and a frame in VLAN 10 does not need to be flooded to the right to switches SW2 and SW3.

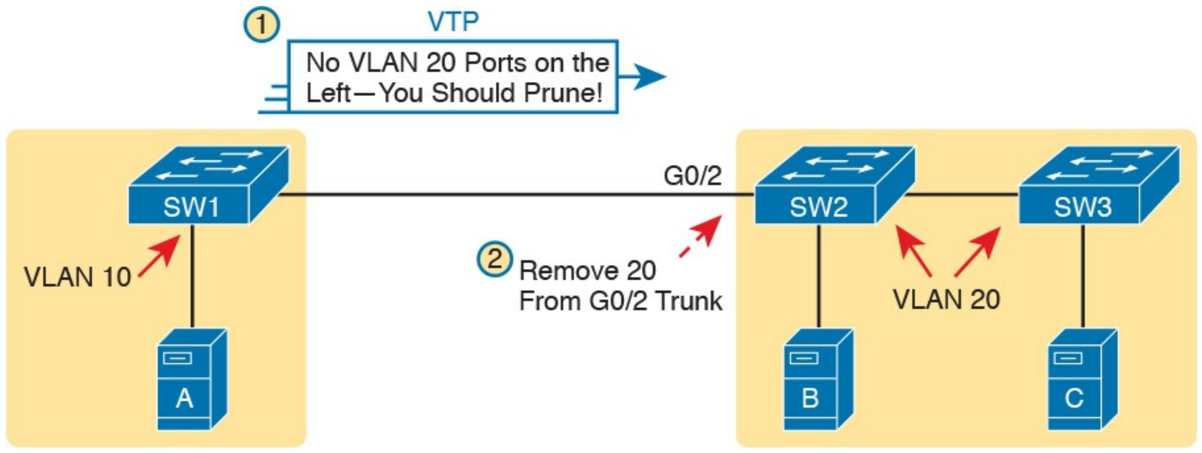

1
VTP
No VLAN 20 Ports on the Left—You Should Prune!
G0/2
SW1
SW2
SW3
VLAN 10
2
Remove 20 From G0/2 Trunk
VLAN 20
A
B
C

Figure 5-6 *VTP Pruning Example*

Figure 5-6 shows two steps that result in VTP pruning VLAN 10 from SW2's G0/2 trunk:

Step 1. SW1 knows about VLAN 20 from VTP, but switch SW1 does not have access ports in VLAN 20. So SW1 announces to SW2 that SW1 would like to prune VLAN 20, so that SW1 no longer receives data frames in VLAN 20.

Step 2. VTP on switch SW2 prunes VLAN 20 from its G0/2 trunk. As a result, SW2 will no longer flood VLAN 20 frames out trunk G0/2 to SW1.

VTP pruning increases the available bandwidth by restricting flooded traffic. VTP pruning is one of the two most compelling reasons to use VTP, with the other reason being to make VLAN configuration easier and more consistent.

Summary of VTP Features

Table 5-2 offers a comparative overview of the three VTP modes.

Key Topic

Function	Server	Client	Transparent
Only sends VTP messages out ISL or 802.1Q trunks	Yes	Yes	Yes
Allows CLI configuration of VLANs	Yes	No	Yes
Can use normal range VLANs (1–1005)	Yes	Yes	Yes
Can use extended range VLANs (1006–4095)	No	No	Yes
Synchronizes its own config database when receiving VTP messages with a higher revision number	Yes	Yes	No
Creates and sends periodic VTP updates every 5 minutes	Yes	Yes	No
Does not process received VTP updates, but does forward received VTP updates out other trunks	No	No	Yes

Table 5-2 VTP Features

VTP Configuration and Verification

VTP configuration requires only a few simple steps, but VTP has the power to cause significant problems, either by accidental poor configuration choices or by malicious attacks. This second major section of the chapter focuses on configuring VTP correctly and verifying its operation. The third major section then looks at troubleshooting VTP, which includes being careful to avoid harmful scenarios.

Using VTP: Configuring Servers and Clients

Before configuring VTP, the network engineer needs to make some choices. In particular, assuming that the engineer wants to make use of VTP's features, the engineer needs to decide which switches will be in the same VTP domain, meaning that these switches will learn VLAN configuration information from each other. The VTP domain name must be chosen, along with an optional but recommended VTP password. (Both the domain name and password are case sensitive.) The engineer must also choose which switches will be servers (usually at least two for redundancy) and which will be clients.

After the planning steps are completed, the following steps can be used to configure VTP:

Config Checklist

Step 1. Use the **vtp mode** {**server** | **client**} command in global configuration mode to enable VTP on the switch as either a server or client.

Step 2. On both clients and servers, use the **vtp domain** *domain-name* command in global configuration mode to configure the case-sensitive VTP domain name.

Step 3. (Optional) On both clients and servers, use the **vtp password**

password-value command in global configuration mode to configure the case-sensitive password.

Step 4. (Optional) On servers, use the **vtp pruning** global configuration command to make the domain-wide VTP pruning choice.

Step 5. (Optional) On both clients and servers, use the **vtp version** {**1** | **2**} command in global configuration mode to tell the local switch whether to use VTP version 1 or 2.

As a network to use in the upcoming configuration examples, Figure 5-7 shows a LAN with the current VTP settings on each switch. At the beginning of the example in this section, both switches have all default VTP configuration: VTP server mode with a null domain name and password. With these default settings, even if the link between two switches is a trunk, VTP would still not work.

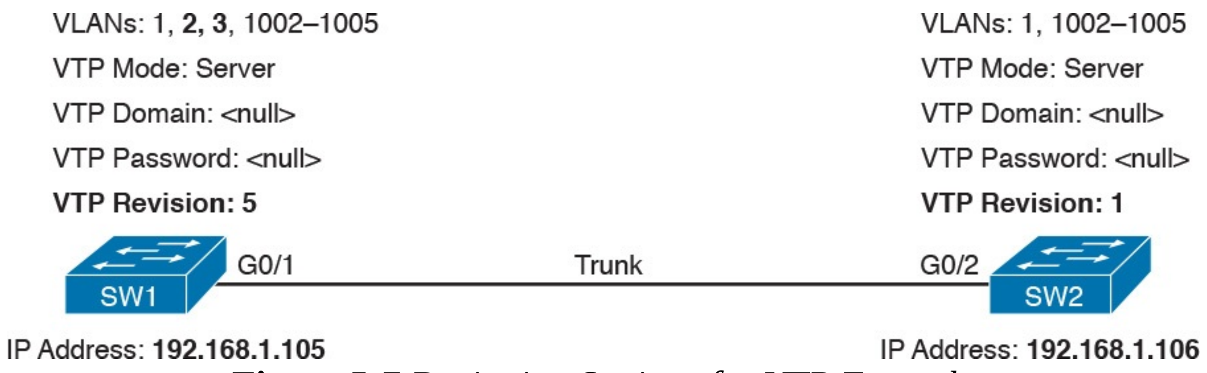


Figure 5-7 *Beginning Settings for VTP Example*

Per the figure, the switches do have some related configuration beyond the VTP configuration. SW1 has been configured to know about two additional VLANs (VLAN 2 and 3). Additionally, both switches have been configured with IP addresses—a fact that will be useful in upcoming **show** command output.

To move toward using VTP in these switches, and synchronizing their VLAN configuration databases, Figure 5-8 repeats Figure 5-7, but with some new configuration settings in bold text. Note that both switches now use the same VTP domain name and password. Switch SW1 remains at the default setting of being a VTP server, while switch SW2 is now configured to be a VTP client. Now, with matching VTP domain and password, and with a trunk between the two switches, the two switches will use VTP successfully.

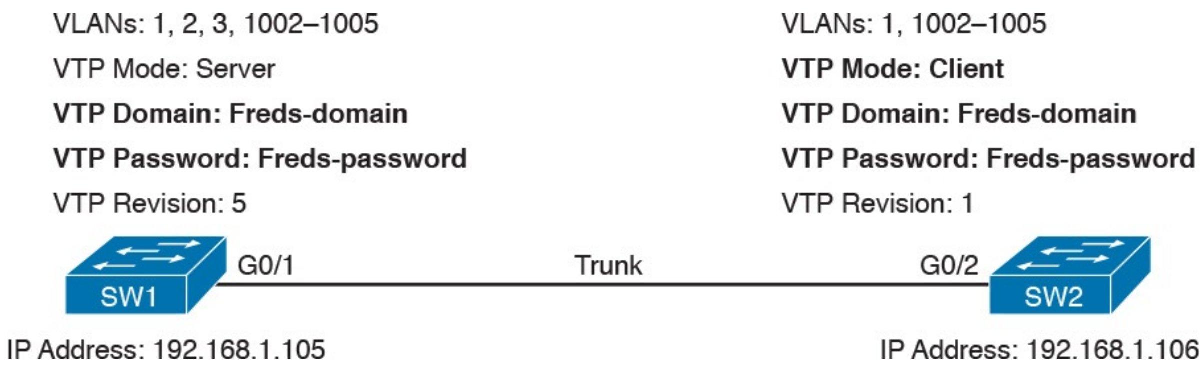


Figure 5-8 *New VTP Configuration Settings Planned for Example 5-1*

Example 5-1 shows the configuration shown in Figure 5-8 as added to each switch.

Example 5-1 *Basic VTP Client and Server Configuration*

Click here to view code image

```
! IOS generates at least one informational message
after each VTP command listed
! below. Those lines are not added as text by the
author; they are generated by IOS.
SW1# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW1(config)# vtp mode server
Setting device to VTP SERVER mode
SW1(config)# vtp domain Freds-domain
Changing VTP domain name from NULL to Freds-domain
SW1(config)# vtp password Freds-password
Setting device VLAN database password to Freds-password
SW1(config)# vtp pruning
Pruning switched on
SW1(config)# ^Z
```

```
! Switching to SW2 now
SW2# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW2(config)# vtp mode client
Setting device to VTP CLIENT mode.
SW2(config)# vtp domain Freds-domain
Domain name already set to Freds-domain.
SW2(config)# vtp password Freds-password
Setting device VLAN database password to Freds-password
SW2(config)# ^Z
```

Make sure and take the time to work through the configuration commands on both switches. The domain name and password, case-sensitive, match. Also, SW2, as client, does not need the **vtp pruning** command, because the VTP server dictates to the domain whether or not pruning is used throughout the domain. (Note that all VTP servers should be configured with the same VTP pruning setting.)

Verifying Switches Synchronized Databases

Configuring VTP takes only a little work, as shown in Example 5-1. Most of the interesting activity with VTP happens in what it learns dynamically, and how VTP accomplishes that learning. For instance, Figure 5-7 showed the switch SW1 had revision number 5 for its VLAN configuration database, while SW2's was revision 1. Once configured as shown in Example 5-1, the following logic happened through an exchange of VTP messages between SW1 and SW2:

1. SW1 and SW2 exchanged VTP messages.
2. SW2 realized that its own revision number (1) was lower (worse) than SW1's revision number 5.
3. SW2 received a copy of SW1's VLAN database and updated SW2's own VLAN (and related) configuration.
4. SW2's revision number also updated to revision number 5.

To confirm that two neighboring switches synchronized their VLAN database, use the **show vtp status** command. Example 5-2 shows this command first on switch SW2, which had a lower revision number (1) at the start of the example, so it should have synchronized its VLAN configuration database with switch SW1. The example shows the output of the **show vtp status** command first on switch SW2, and then from switch SW1.

Example 5-2 *Demonstrating the Switch SW2's VLAN Database Updated to Revision 5*

Click here to view code image

```
! First, the output from SW2, the VTP Client, formerly
revision 1, now revision 5
SW2# show vtp status
VTP Version capable              : 1 to 3
VTP version running              : 1
VTP Domain Name                  : Freds-domain
VTP Pruning Mode                 : Enabled
VTP Traps Generation             : Disabled
Device ID                        : bcc4.938b.a180
```

```
Configuration last modified by 192.168.1.105 at 2-21-16
11:45:33
Local updater ID is 192.168.1.105 on interface Vl1
(lowest numbered VLAN interface
  found)

Feature VLAN:
--------------
VTP Operating Mode                : Client
Maximum VLANs supported locally   : 1005
Number of existing VLANs          : 7
Configuration Revision            : 5
MD5 digest                        : 0xF3 0x07 0x44 0xA4
0xDE 0x82 0xCD 0xB0
                                    0x9E 0x8F 0x0B 0xD1
0xFD 0xE7 0xE7 0xB3
```

```
! Switching to SW1 now; all highlighted items match
switch SW2
! Back on SW1, the output below confirms the same
revision number as SW2, meaning
! that the two switches have synchronized their VLAN
databases.
SW1# show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : Freds-domain
VTP Pruning Mode                : Enabled
VTP Traps Generation            : Disabled
Device ID                       : bcc4.938b.e500
Configuration last modified by 192.168.1.105 at 2-21-16
11:45:33
Local updater ID is 192.168.1.105 on interface Vl1
(lowest numbered VLAN interface
  found)

Feature VLAN:
--------------
VTP Operating Mode                : Server
Maximum VLANs supported locally   : 1005
Number of existing VLANs          : 7
Configuration Revision            : 5
MD5 digest                        : 0xF3 0x07 0x44 0xA4
0xDE 0x82 0xCD 0xB0
                                    0x9E 0x8F 0x0B 0xD1
0xFD 0xE7 0xE7 0xB3
SW1# show vtp password
VTP Password: Freds-password
```

The example shows two facts that confirm that the two switches have synchronized to use the same VLAN configuration database due to VTP:

- The highlighted line that states “Configuration last modified by...” lists the same IP address and timestamp. Both SW1 and SW2 list the exact same switch, with address 192.168.1.105. (Per Figure 5-8, 192.168.1.105 is switch SW1.) Also, note the text on SW1 lists “Local updater ID is 192.168.1.105...” which means that the local switch (SW1) is 192.168.1.105. The fact that both switches list the same IP address and timestamp confirm that they use the same database, in this case as supplied by 192.168.1.105, which is switch SW1.
- The “Configuration Revision” of 5 listed by both switches also confirms that they both use the same VLAN database.

Note

Using NTP along with VTP can be useful so that the timestamps in the **show vtp status** command on neighboring switches have the same time listed.

Beyond those two key facts, the **show vtp status** command shows several key pieces of information that must match on two neighboring switches before they can succeed at exchanging their database. As highlighted only in switch SW1’s output in Example 5-2:

- Both use the same domain name (Freds-domain).
- Both have the same MD5 digest.

Note that while it is a good practice to set the switches to all use either version 1 or version 2, mismatched versions do not prevent VTP servers and clients from exchanging VTP configuration databases.

The last item in the list, about the MD5 hash, needs a little further explanation. VTP on a switch takes the domain name and the VTP password and applies MD5 to create an MD5 digest, as displayed in the **show vtp status** command’s output. If either the domain name or password does not match, the MD5 digests will not match, and the two switches will not exchange VLAN configuration with VTP. (Note that the end of Example 5-2 lists a sample **show vtp password** command, which lists the clear text VTP password.)

Any command that lists the VLANs known to a switch can also confirm that VTP worked. Once a VTP client or server learns a new VLAN configuration database from a neighbor, its list of VLANs should be identical to that of the neighbor.

For instance, with the configuration suggested in Figure 5-8, as shown in Example 5-1, VTP server SW1 began with VLANs 1, 2, 3 and default VLANs 1002–1005, while switch SW2 only knew about the default VLANs: 1 and 1002–1005. Example 5-3 lists the output of **show vlan brief** on switch SW2, confirming that it now also knows about VLANs 2 and 3. Note that switch SW2 also learned the names of the VLANs, not just the VLAN IDs.

Example 5-3 *Switch SW2 Now Knows About VLANs 2 and 3*

Click here to view code image

```
SW2# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------
------------------------
1    default                          active    Fa0/1,
Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5,
Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9,
Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13,
Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17,
Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21,
Fa0/22, Fa0/23, Fa0/24
                                                Gi0/1
2    Freds-vlan                       active
3    VLAN0003                         active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

Storing the VTP and Related Configuration

Interestingly, even though VTP synchronizes VLAN and VTP configuration, you cannot just issue a **show running-config** command to discover if a switch has synchronized its VLAN configuration database. VTP does not place the configuration commands into the running-config or startup-config file of the VTP server or client. Instead, VTP server and client mode switches store the **vtp** configuration commands, and some VLAN configuration commands, in the vlan.dat file in flash. To verify these configuration commands and their settings, use the **show vtp status** and **show vlan** commands.

Figure 5-9 shows an example. It shows three key VTP commands (**vtp mode**, **vtp domain**, and **vtp password**), plus a **vlan 10** command that creates VLAN 10. It also shows the **switchport access vlan 10** interface subcommand for contrast. Of these, on a VTP server or client, only the **switchport access vlan 10** command would be part of the running-config or startup-config file.

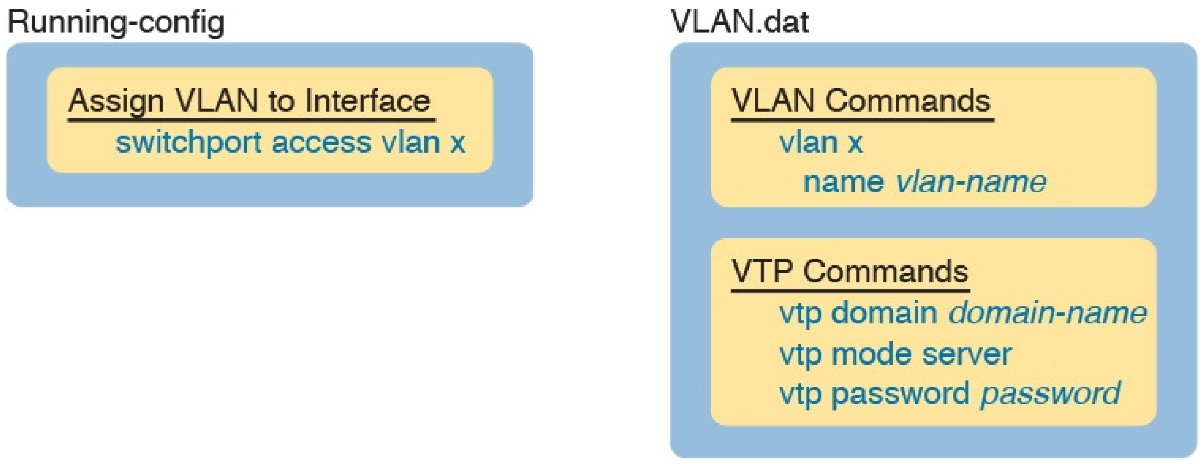
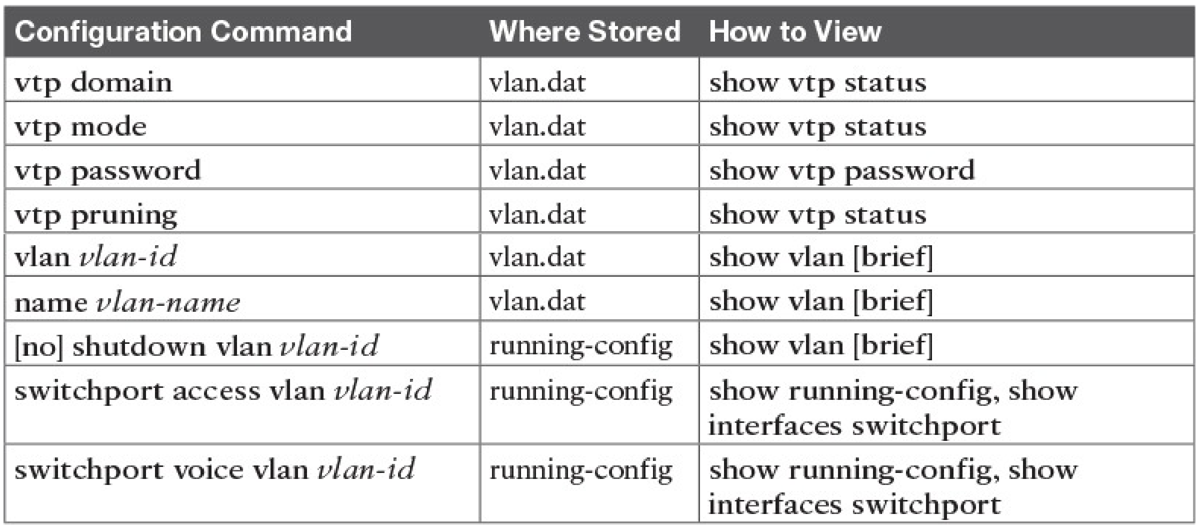


Figure 5-9 *Where VTP Stores Configuration: VTP Client and Server*

There is no equivalent of a **show running-config** command to display the contents of the vlan.dat file. Instead, you have to use various **show vtp** and **show vlan** commands to view information about VLANs and VTP. For reference, Table 5-3 lists the VLAN-related configuration commands, the location in which a VTP server or client stores the commands, and how to view the settings for the commands.



Configuration Command	Where Stored	How to View
vtp domain	vlan.dat	show vtp status
vtp mode	vlan.dat	show vtp status
vtp password	vlan.dat	show vtp password
vtp pruning	vlan.dat	show vtp status
vlan *vlan-id*	vlan.dat	show vlan [brief]
name *vlan-name*	vlan.dat	show vlan [brief]
[no] shutdown vlan *vlan-id*	running-config	show vlan [brief]
switchport access vlan *vlan-id*	running-config	show running-config, show interfaces switchport
switchport voice vlan *vlan-id*	running-config	show running-config, show interfaces switchport

Table 5-3 Where VTP Clients and Servers Store VLAN-Related Configuration

Note that switches using VTP transparent mode (**vtp mode transparent**), or with VTP disabled (**vtp mode off**), store all the commands listed in Table 5-3 in the running-config and startup-config files.

An interesting side effect of how VTP stores configuration is that when you use a VTP client or server switch in a lab, and you want to remove all the configuration to start with a clean switch with all default VTP and VLAN configuration, you must issue more than the **erase startup-config** command. If you only erase the startup-config and reload the switch, the switch remembers all VLAN config and VTP configuration that is instead stored in the vlan.dat file in flash. To remove those configuration details before reloading a switch, you would have to delete the vlan.dat file in flash with a command such as **delete flash:vlan.dat**.

Avoiding Using VTP

For most of the history of VTP, one option existed for avoiding using VTP: using VTP transparent mode. That is, each switch technically had to use VTP in one of three modes (server, client, or transparent).

In transparent mode, a switch never updates its VLAN database based on a received VTP message, and never causes other switches to update their databases based on the transparent mode switch's VLAN database. The only VTP action performed by the switch is to forward VTP messages received on one trunk out all the other trunks, which allows other VTP clients and servers to work correctly.

Configuring VTP transparent mode is simple: Just issue the **vtp mode transparent** command in global configuration mode.

Cisco eventually added an option to disable VTP altogether, with the **vtp**

mode off global command. Note that one key difference exists versus using transparent mode: switches using **vtp mode off** do not forward VTP messages. In short, if you want a switch to ignore VTP, but forward VTP message from other switches, use transparent mode. If you want a switch to ignore VTP, including not forwarding any VTP messages, disable VTP.

VTP Troubleshooting

Troubleshooting VTP can be both simple and tricky at the same time. To troubleshoot issues in which VTP fails to cause synchronization to happen, you just have to work a short checklist, find the configuration or status issue, and solve the problem. From the complete opposite direction, VTP can cause synchronization, but with bad results, using the wrong switch's VLAN database. This last section looks at the straightforward case of troubleshooting why VTP does not synchronize, as well as a few cases as to the dangers of VTP synchronizing with unfortunate results.

Determining Why VTP Is Not Synchronizing

VTP troubleshooting can be broken down a pair of neighboring switches at a time. For any VTP domain, with a number of switches, find any two neighboring switches. Then troubleshoot to discover whether those two switches fail to meet the requirements to allow VTP to synchronize, and then fix the problem. Then work through every pair until VTP works throughout the VTP domain.

The troubleshooting process must begin with some basics. You need to learn about the LAN topology to then find and choose some neighboring switches to investigate. Then you need to determine whether the neighbors have synchronized or not, mainly by checking their list of VLANs, or by looking at information in the **show vtp status** command. For any pair of neighboring switches that have not synchronized, work through the list of configuration settings until the problem is fixed.

The following list details a good process to find VTP configuration problems, organized into a list for easier study and reference.

Step 1. Confirm the switch names, topology (including which interfaces connect which switches), and switch VTP modes.

Step 2. Identify sets of two neighboring switches that should be either VTP clients or servers whose VLAN databases differ with the **show vlan** command.

Step 3. On each pair of two neighboring switches whose databases differ, verify the following:

A. Because VTP messages only flow over trunks, at least one operational trunk should exist between the two switches (use the **show interfaces trunk**, **show interfaces switchport**, or **show cdp neighbors** command).

B. The switches must have the same (case-sensitive) VTP domain name (**show vtp status**).

C. If configured, the switches must have the same (case-sensitive) VTP password (**show vtp password**).

D. The MD5 digest should be the same, as evidence that both the domain name and any configured passwords are the same on both switches (**show vtp status**).

E. While VTP pruning should be enabled or disabled on all servers in the same domain, having two servers configured with opposite pruning settings does not prevent the synchronization process.

Step 4. For each pair of switches identified in Step 3, solve the problem by either troubleshooting the trunking problem or reconfiguring a switch to correctly match the domain name or password.

VTP also has a few related commands that you might think would prevent synchronization, but they do not. Remember these facts about VTP for items that do not cause a problem for VTP synchronization:

- The VTP pruning setting does not have to match on neighboring switches (even though in a real VTP network you would likely use the same setting on all switches).
- The VTP version does not have to match between two switches that are any combination of VTP server and client for neighboring switches to synchronize.
- When deciding if VTP has synchronized, note that the administrative status of a VLAN (per the **shutdown vlan** *vlan-id* global configuration command and the **shutdown** command in VLAN configuration mode) is not communicated by VTP. So two neighboring switches can know about the same VLAN, with that VLAN shut down on one switch and active on the other.

Common Rejections When Configuring VTP

VTP clients cannot configure VLANs at all, to either add them, delete them, or name them. VTP servers (when using VTP versions 1 and 2) have the restriction of working with standard number VLANs only. This next short topic looks at the error messages shown when you attempt to add those

VLANs in spite of what the chapter claims is allowed, just so you know what the error message looks like.

Example 5-4 shows some output on a switch (SW3) that is a VTP client. Focus first on the rejection of the **vlan 200** command. The result is clear and obvious: The user issued the **vlan 200** command, and IOS lists an error message about the switch being a VTP client.

Example 5-4 *Attempting* **vlan** *Commands on VTP Clients and Servers*

Click here to view code image

```
SW3# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
SW3(config)# vlan 200
VTP VLAN configuration not allowed when device is in
CLIENT mode.
SW3(config)# vlan 2000
SW3(config-vlan)# exit
% Failed to create VLANs 2000
Extended VLAN(s) not allowed in current VTP mode.
%Failed to commit extended VLAN(s) changes.
SW3(config)#
```

The second half of the example shows a couple of oddities. First, the **vlan 200** command is immediately rejected. Second, the **vlan 2000** command is also rejected, but not immediately. IOS, in an odd twist of logic, does not actually try and add the configuration of extended mode VLANs until the user exits VLAN configuration mode. Once the **exit** command was issued, IOS issued the three highlighted error messages—all messages that confirm in some way that the VLAN 2000 was not created.

Note that on a VTP server, the **vlan 200** command would have been accepted but the **vlan 2000** command would have been rejected, with the same process as shown in the example.

Problems When Adding Switches to a Network

VTP can be running just fine for months, and then one day, the help desk receives a rash of calls describing cases in which large groups of users can no longer use the network. After further examination, it appears that most every VLAN in the campus has been deleted. The switches still have many interfaces with **switchport access vlan** commands that refer to the now-deleted VLANs. None of the devices on those now-deleted VLANs work, because Cisco switches do not forward frames for nonexistent VLANs.

VTP can cause the kind of pervasive LAN problems described in that previous paragraph, so you have to be careful when using VTP. This kind of problem can occur when a new switch is connected to an existing network. Whether this problem happens by accident or as a denial of service (DoS) attack, the root cause is this:

> When two neighboring switches first connect with a trunk, and they also meet all the requirements to synchronize with VTP, the switch with the lower revision number accepts the VLAN database from the neighbor with the higher revision number.

Note in particular that the preceding statement says nothing about which switch is the server or client, or which switch is the older production switch versus the newly added switch. That is, no matter whether a server has the higher revision number or the client does, the two switches converge to both use the VLAN database with the higher revision number. There is no logic about which switch might be client or server, or which switch is the new switch in the network and which is the old established switch.

This VTP behavior of using the higher revision number when connecting new switches has some pretty powerful implications. For instance, consider the following scenario: Someone is studying for the CCNA R&S exam, using the equipment in the small lab room at work. The lab has a couple of LAN switches isolated from the production network—that is, the switches have no links even cabled to the production network. But because the engineer knows the VTP domain name and password used in production, when configuring in the lab, the engineer uses that same VTP domain name and password. That causes no problems (yet), because the lab switches do not even connect to the production network. (In real life, use a different VTP domain name and password in your lab gear!)

This same engineer continues CCNA studying and testing in the lab, making lots of changes to the VLAN configuration. Each change kicks the VLAN configuration database revision number up by 1. Eventually, the lab switches have a high VTP configuration revision number, so high that the number is higher than that of the production switches. But the lab is still isolated, so there is still no problem.

Do you see the danger? All that has to happen now is for someone to connect a link from a lab switch to a production switch and make it trunk. For instance, imagine now that some other engineer decides to do some testing in the lab and does not think to check the VTP status on the lab switches versus the production switches. That second engineer walks into the lab and connects the lab switches to the production network. The link negotiates trunking...VTP synchronizes between a lab switch and a production switch...and those two switches discover that the lab switch’s configuration database has a higher revision number. At this point, VTP is now happily doing its job, synchronizing the VLAN configuration database, but unfortunately, VTP is distributing the lab's VLAN configuration, deleting production VLANs.

In real life, you have several ways to help reduce the chance of such problems when installing a new switch to an existing VTP domain. In particular, before connecting a new switch to an existing VTP domain, reset the new switch's VTP revision number to 0 by either of the following methods:

- Configure the new switch for VTP transparent mode and then back to VTP client or server mode.
- Erase the new switch's vlan.dat file in flash and reload the switch. (The vlan.dat file contains the switch's VLAN database, including the revision number.)

Besides the suggestion of resetting the VLAN database revision number before installing a new switch, a couple of other good VTP conventions, called best practices, can help avoid some of the pitfalls of VTP:

Key Topic

- If you do not intend to use VTP, configure each switch to use transparent mode (**vtp mode transparent**) or off mode (**vtp mode off**).
- If using VTP server or client mode, always use a VTP password. That way a switch that uses default settings (server mode, with no password set) will not accidentally overwrite the production VLAN database if connected to the production network with a trunk.
- In a lab, if using VTP, always use a different domain name and password than you use in production.
- Disable trunking with the **switchport mode access** and **switchport nonegotiate** commands on all interfaces except known trunks, preventing VTP attacks by preventing the dynamic establishment of trunks.

It is possible that an attacker might attempt a DoS attack using VTP. Preventing the negotiation of trunking on most ports can greatly reduce the attacker's opportunities to even try. Also, with a VTP password set on all switches, even if the attacker manages to get trunking working between the attacker's switch and a production switch, the attacker would then have to know the password to do any harm. And of course, either using transparent mode or disabling VTP completely removes the risk.

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book, DVD, or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 5-4 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Review Element	Review Date(s)	Resource Used
Review key topics		Book, DVD/website
Review key terms		Book, DVD/website
Answer DIKTA questions		Book, PCPT
Do labs		Blog
Review memory tables		DVD/website
Review config checklists		Book, DVD/website
Review command tables		Book

Table 5-4 Chapter Review Tracking

Review All the Key Topics

Key Topic

Key Topic Element	Description	Page Number
List	Requirements for VTP to work between two switches	126
Table 5-2	VTP features summary	128
Figure 5-9	Description of where VTP client and server switches store configuration	134
Table 5-3	List of commands and where a VTP client and server store those commands	134
List	Troubleshooting checklist for VTP	136
List	VTP best practices	139

Table 5-5 Key Topics for Chapter 5

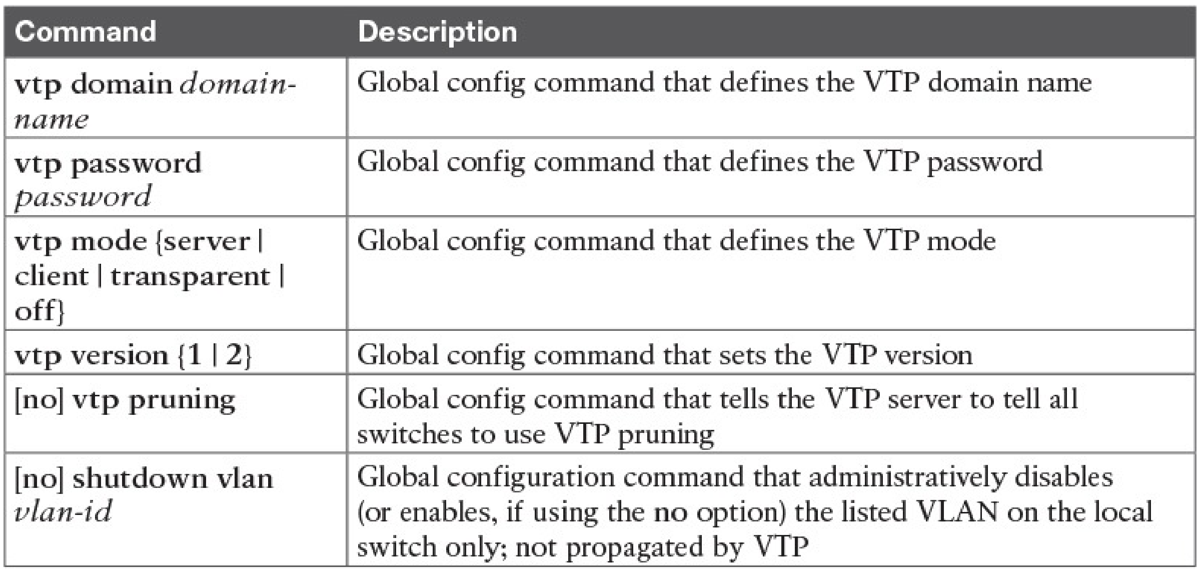
Key Terms You Should Know

VLAN configuration database
configuration revision number
vlan.dat
VTP
VTP client mode
VTP pruning
VTP server mode
VTP transparent mode

VTP synchronization

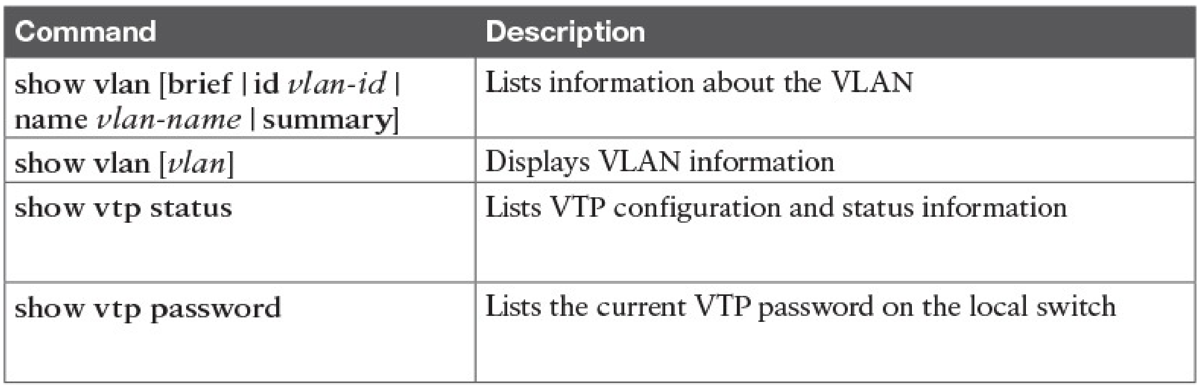
Command References

Tables 5-6 and 5-7 list configuration and verification commands used in this chapter, respectively. As an easy review exercise, cover the left column in a table, read the right column, and try to recall the command without looking. Then repeat the exercise, covering the right column, and try to recall what the command does.



Command	Description
vtp domain *domain-name*	Global config command that defines the VTP domain name
vtp password *password*	Global config command that defines the VTP password
vtp mode {server | client | transparent | off}	Global config command that defines the VTP mode
vtp version {1 | 2}	Global config command that sets the VTP version
[no] vtp pruning	Global config command that tells the VTP server to tell all switches to use VTP pruning
[no] shutdown vlan *vlan-id*	Global configuration command that administratively disables (or enables, if using the no option) the listed VLAN on the local switch only; not propagated by VTP

Table 5-6 Chapter 5 Configuration Command Reference



Command	Description
show vlan [brief | id *vlan-id* | name *vlan-name* | summary]	Lists information about the VLAN
show vlan [*vlan*]	Displays VLAN information
show vtp status	Lists VTP configuration and status information
show vtp password	Lists the current VTP password on the local switch

Table 5-7 Chapter 5 EXEC Command Reference

Chapter 6. Miscellaneous LAN Topics

This chapter covers the following exam topics:

1.0 LAN Switching Technologies

1.6 Describe the benefits of switch stacking and chassis aggregation

1.7 Describe common access layer threat mitigation techniques

1.7.a 802.1x

1.7.b DHCP snooping

5.0 Infrastructure Maintenance

5.4 Describe device management using AAA with TACACS+ and RADIUS

Between this book and the ICND1 100-105 Cert Guide, 14 chapters have been devoted to topics specific to LANs. This chapter is the last of those chapters. This chapter completes the LAN-specific discussion with a few small topics that just do not fit neatly in the other chapters.

The chapter begins with three security topics. The first section addresses IEEE 802.1x, which defines a mechanism to secure user access to a LAN by requiring the user to supply a username and password before a switch allows the device's sent frames into the LAN. This tool helps secure the network against attackers gaining access to the network. The second section, "AAA Authentication," discusses network device security, protecting router and switch CLI access by requiring username/password login with an external authentication server. The third section, "DHCP Snooping," explores how switches can prevent security attacks that take advantage of DHCP messages and functions. By watching DHCP messages, noticing when they are used in abnormal ways not normally done with DHCP, DHCP can prevent attacks by simply filtering certain DHCP messages.

The final of the four major sections in this chapter looks at two similar design tools that make multiple switches act like one switch: switch stacking and chassis aggregation. Switch stacking allows a set of similar switches that sit near to each other (in the same wiring closet, typically in the same part of the same rack) to be cabled together and then act like a single switch. Using a switch stack greatly reduces the amount of work to manage the switch, and it reduces the overhead of control and management protocols used in the network. Switch chassis aggregation has many of the same benefits, but is supported more often as a distribution or core switch feature, with switch stacking as a more typical access layer switch feature.

All four sections of this chapter have a matching exam topic that uses the verb "describe," so this chapter touches on the basic descriptions of a tool, rather than deep configuration. A few of the topics will show some configuration as a means to describe the topic, but the chapter is meant to help you come away with an understanding of the fundamentals, rather than an ability to configure the features.

"Do I Know This Already?" Quiz

Take the quiz (either here, or use the PCPT software) if you want to use the score to help you decide how much time to spend on this chapter. The answers are at the bottom of the page following the quiz, and the explanations are in DVD Appendix C and in the PCPT software.

Foundation Topics Section	Questions
Securing Access with IEEE 802.1x	1
AAA Authentication	2
DHCP Snooping	3–4
Switch Stacking and Chassis Aggregation	5

Table 6-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

1. With IEEE 802.1x, which role does a LAN switch typically play?
   a. Authentication server
   b. Supplicant
   c. Translator
   d. Authenticator
2. Which of the following answers is true of TACACS+ but not true of RADIUS?
   a. The protocol encrypts the password for transmission.
   b. Uses UDP as the transport protocol.
   c. Supports ability to authorize different users to use different subsets of CLI commands.
   d. Defined by an RFC.
3. An engineer hears about DHCP snooping and decides to implement it. The network includes devices that act primarily as Layer 2 switches, multilayer switches (that is, they perform both Layer 2 and Layer 3 switching), and routers. Which of the following are the devices on which DHCP snooping could be implemented? (Choose two answers.)
   a. Layer 2 switches
   b. Routers

c. Multilayer switches

d. A LAN hub

4. Layer 2 switch SW2 connects to several devices: a Layer 2 switch (SW1), a router, a DHCP server, and three PCs (PC1, PC2, and PC3). All PCs are expected to use DHCP to lease their IP addresses. A network engineer implements DHCP snooping on switch SW2. Unknown to the engineer, a malicious attacker is using PC3. Which of the following is the most likely DHCP snooping trust state configuration on SW2 for the ports connected to the listed devices? (Choose two answers.)

   a. The port connected to the router is untrusted.

   b. The port connected to switch SW1 is trusted.

   c. The port connected to PC1 is untrusted.

   d. The port connected to PC3 is trusted.

5. A network engineer takes four 2960-X switches and creates a switch stack using either FlexStack or FlexStack-Plus stacking from Cisco. Now consider data plane functions, such as frame forwarding; control plane functions, such as STP and VTP; and management plane functions, such as Telnet and SSH support. Once the stack is cabled and working, which of the following is true about how the stack of four switches works?

   a. The stack acts as one switch for data plane functions, and separate switches for control and management plane functions.

   b. The stack acts as one switch for data plane and control plane functions, and separate switches for management plane functions.

   c. The stack acts as one switch for data plane, control plane, and management plane functions.

   d. The stack does not act as one switch for data, control, or management plane functions, instead providing backup uplinks if all of one switch's uplinks fail.

Answers to the "Do I Know This Already?" quiz:

1 D 2 C 3 A, C 4 B, C 5 C

Foundation Topics

Securing Access with IEEE 802.1x

In some enterprise LANs, the LAN is built with cables run to each desk, in every cubicle and every office. When you move to a new space, all you have to do is connect a short patch cable from your PC to the RJ-45 socket on the wall and you are connected to the network. Once booted, your PC can send packets anywhere in the network. Security? That happens mostly at the devices you try to access, for instance, when you log in to a server.

That previous paragraph is true of how many networks work. That attitude views the network as an open highway between the endpoints in the network, and the network is there to create connectivity, with high availability, and to make it easy to connect your device. Those goals are worthy goals. However, making the LAN accessible to anyone, so that anyone can attempt to connect to servers in the network, allows attackers to connect and then try and break in to the security that protects the server. That approach may be too insecure. For instance, any attacker who could gain physical access could plug in his laptop and start running tools to try to exploit all those servers attached to the internal network.

Today, many companies secure access to the network. Sure, they begin by creating basic connectivity: cabling the LAN and connecting cables to all the desks at all the cubicles and offices. All those ports physically work. But a user cannot just plug in her PC and start working; she must go through a security process before the LAN switch will allow the user to send any other messages in the network.

Switches can use IEEE standard 802.1x to secure access to LAN ports. To set up the feature, the LAN switches must be configured to enable 802.1x. Additionally, the IT staff must implement an authentication, authorization, and accounting (AAA) server. The AAA server (commonly pronounced "triple A" server) will keep the usernames and passwords, and when the user supplies that information, it is the AAA server that determines if what the user typed was correct or not.

Once implemented, the LAN switch acts as an 802.1x *authenticator*, as shown in Figure 6-1. As an 802.1x authenticator, a switch can be configured to enable some ports for 802.1x, most likely the access ports connected to end users. Enabling a port for 802.1x defines that when the port first comes up, the switch filters all incoming traffic (other than 802.1x traffic). 802.1x must first authenticate the user that uses that device.

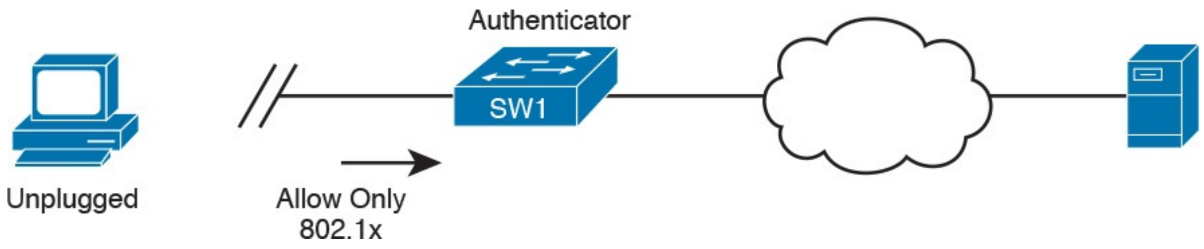


Figure 6-1 *Switch as 802.1x Authenticator, with AAA Server, and PC Not*

Note that the switch usually configures access ports that connect to end users with 802.1x, but does not enable 802.1x on ports connected to IT-controlled devices, such as trunk ports, or ports connected in parts of the network that are physically more secure.

The 802.1x authentication process works like the flow in Figure 6-2. Once the PC connects and the port comes up, the switch uses 802.1x messages to ask the PC to supply a username/password. The PC user must then supply that information. For that process to work, the end-user device must be using an 802.1x client called a *supplicant*; many OSs include an 802.1x supplicant, so it may just be seen as a part of the OS settings.

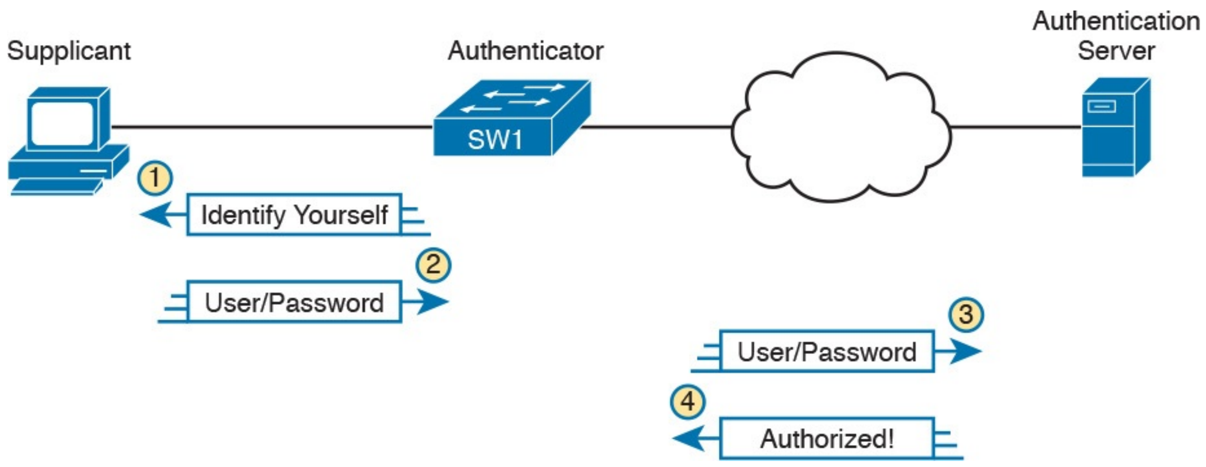


Figure 6-2 *Generic 802.1x Authentication Flows*

At Steps 3 and 4 in Figure 6-2, the switch authenticates the user, to find out if the username and password combination is legitimate. The switch, acting as 802.1x authenticator, asks the AAA server if the supplied username and password combo is correct, with the AAA server answering back. If the username and password were correct, then the switch authorizes the port. Once authorized, the switch no longer filters incoming messages on that port. If the username/password check shows that the username/password was incorrect, or the process fails for any reason, the port remains in an unauthorized state. The user can continue to retry the attempt.

Figure 6-3 rounds out this topic by showing an example of one key protocol used by 802.1x: Extensible Authentication Protocol (EAP). The switch (the authenticator) uses RADIUS between itself and the AAA server, which itself uses IP and UDP. However, 802.1x, an Ethernet protocol, does not use IP or UDP. But 802.1x wants to exchange some authentication information all the way to the RADIUS AAA server. The solution is to use EAP, as shown in Figure 6-3.

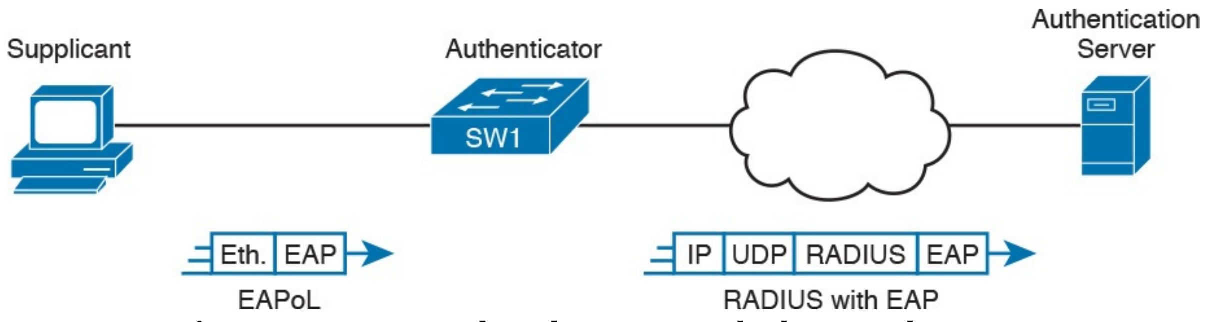


Figure 6-3 *EAP and Radius Protocol Flows with 802.1x*

As shown in the figure, the EAP message flows from the supplicant to the authentication server, just in different types of messages. The flow from the supplicant (the end-user device) to the switch transports the EAP message directly in an Ethernet frame with an encapsulation called EAP over LAN (EAPoL). The flow from the authenticator (switch) to the authentication server flows in an IP packet. In fact, it looks much like a normal message used by the RADIUS protocol (RFC 2865). The RADIUS protocol works as a UDP application, with an IP and UDP header, as shown in the figure.

Now that you have heard some of the details and terminology, this list summarizes the entire process:

- A AAA server must be configured with usernames and passwords.
- Each LAN switch must be enabled for 802.1x, to enable the switch as an authenticator, to configure the IP address of the AAA server, and to enable 802.1x on the required ports.
- Users must know a username/password combination that exists on the AAA server, or they will not be able to access the network from any device.

AAA Authentication

The ICND1 100-105 Cert Guide discusses many details about device management, in particular how to secure network devices. However, all those device security methods shown in the ICND1 half of the CCNA R&S exam topics use locally configured information to secure the login to the networking device.

Using locally configured usernames and passwords configured on the switch causes some administrative headaches. For instance, every switch and router needs the configuration for all users who might need to log in to the devices. Good security practices tell us to change our passwords regularly, but logging in to hundreds of devices to change passwords is a large task, so often, the passwords remain unchanged for long periods.

A better option would be to use an external AAA server. The AAA server centralizes and secures all the username/password pairs. The switches and routers still need some local security configuration to refer to the AAA server, but the username/password exist centrally, greatly reducing the administrative effort, and increasing the chance that passwords are changed regularly and are more secure. It is also easier to track which users logged in to which devices and when, and to revoke access as people leave their current job.

This short section discusses the basics of how networking devices can use a AAA server.

AAA Login Process

First, to use AAA, the site would need to install and configure a AAA server, such as the Cisco Access Control Server (ACS). Cisco ACS is AAA software that you can install on your own server (physical or virtual).

The networking devices would each then need new configuration to tell the device to start using the AAA server. That configuration would point to the IP address of the AAA server, and define which AAA protocol to use: either TACACS+ or RADIUS. The configuration includes details about TCP (TACACS+) or UDP (RADIUS) ports to use.

When using a AAA server for authentication, the switch (or router) simply sends a message to the AAA server asking whether the username and password are allowed, and the AAA server replies. Figure 6-4 shows an example, with the user first supplying his username/password, the switch asking the AAA server, and the server replying to the switch stating that the username/password is valid.

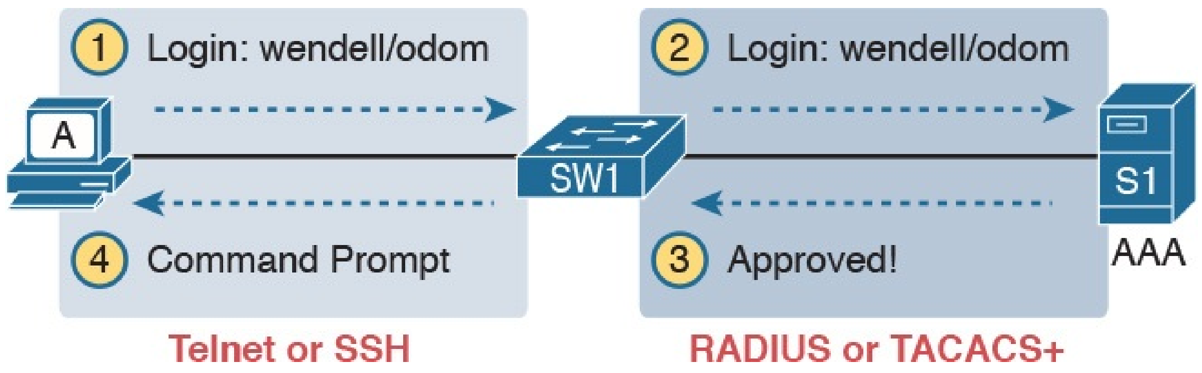


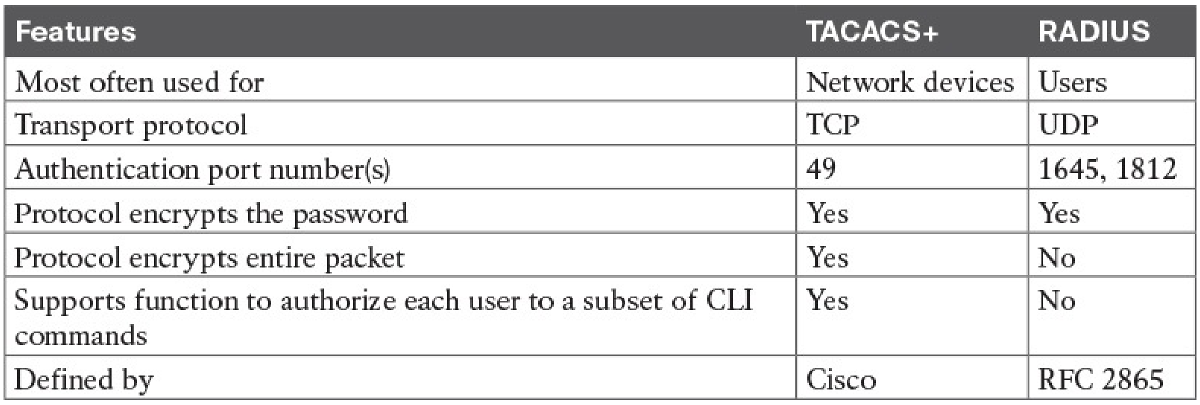
Figure 6-4 *Basic Authentication Process with an External AAA Server*

TACACS+ and RADIUS Protocols

While Figure 6-4 shows the general idea, note that the information flows with a couple of different protocols. On the left, the connection between the user and the switch or router uses Telnet or SSH. On the right, the switch and AAA server typically use either the RADIUS or TACACS+ protocol, both of which encrypt the passwords as they traverse the network.

The AAA server can also provide authorization and accounting features as well. For instance, for networking devices, IOS can be configured so that each user can be allowed to use only a specific subset of CLI commands. So, instead of having basically two levels of authority—user mode and privileged mode—each device can configure a custom set of command authority settings per user. Alternately, those details can be centrally configured at the AAA server, rather than configuring the details at each device. As a result, different users can be allowed to use different subsets of the commands, but as identified through requests to the AAA server, rather than repetitive laborious configuration on every device. (Note that TACACS+ supports this particular command authorization function, whereas RADIUS does not.)

Table 6-2 lists some basic comparison points between TACACS+ and RADIUS.



Features	TACACS+	RADIUS
Most often used for	Network devices	Users
Transport protocol	TCP	UDP
Authentication port number(s)	49	1645, 1812
Protocol encrypts the password	Yes	Yes
Protocol encrypts entire packet	Yes	No
Supports function to authorize each user to a subset of CLI commands	Yes	No
Defined by	Cisco	RFC 2865

Table 6-2 Comparisons Between TACACS+ and RADIUS

AAA Configuration Examples

Learning how to configure a router or switch to use a AAA server can be difficult. AAA requires that you learn several new commands. Besides all that, enabling AAA actually changes the rules used on a router for login authentication—for instance, you cannot just add a **login** command on the console line anymore after you have enabled AAA.

The exam topics use the phrase “describe” regarding AAA features, rather than configure, verify, or troubleshoot. However, to understand AAA on switches or routers, it helps to work through an example configuration. This next topic focuses on the big ideas behind a AAA configuration, instead of worrying about working through all the parameters, verifying the results, or memorizing a checklist of commands. The goal is to help you see how AAA on a switch or router changes login security.

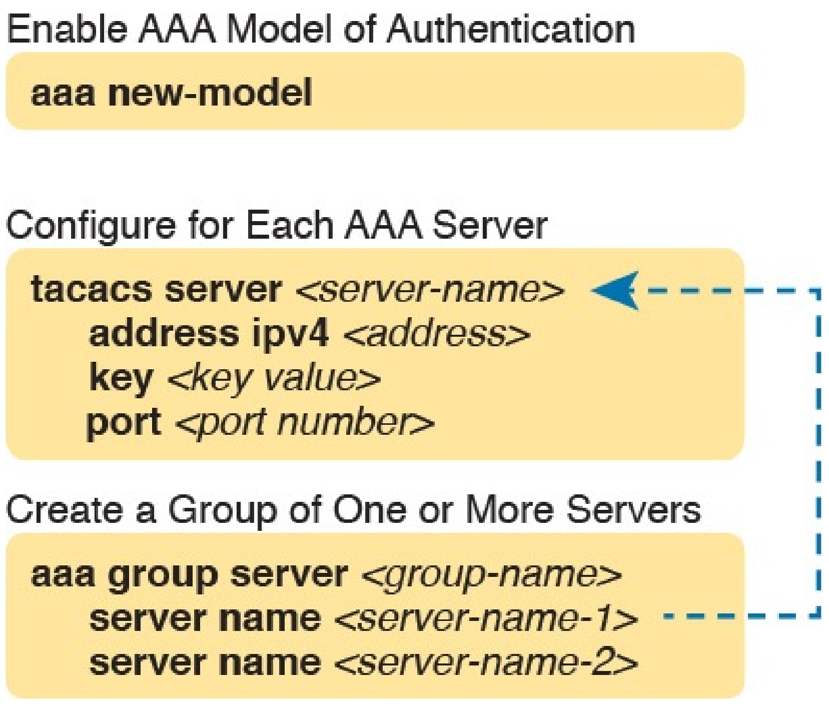
Note

Throughout this book and the ICND1 Cert Guide, the login security details work the same on both routers and switches, with the exception that switches do not have an auxiliary port, whereas routers often do. But the configuration works the same on both routers and switches, so when this section mentions a switch for login security, the same concept applies to routers as well.

Everything you learned about switch login security for ICND1 in the ICND1 Cert Guide assumed an unstated default global command: **no aaa new-model**. That is, you had not yet added the **aaa new-model** global command to the configuration. Configuring AAA requires the **aaa new-model** command, and this single global command changes how that switch does login security.

The **aaa new-model** global command enables AAA services in the local switch (or router). It even enables new commands, commands that would not have been accepted before, and that would not have shown up when getting help with a question mark from the CLI. The **aaa new-model** command also changes some default login authentication settings. So, think of this command as the dividing line between using the original simple ways of login security versus a more advanced method.

After configuring the **aaa new-model** command on a switch, you need to define each AAA server, plus configure one or more groups of AAA servers aptly named a AAA group. For each AAA server, configure its IP address and a key, and optionally the TCP or UDP port number used, as seen in the middle part of Figure 6-5. Then you create a server group for each group of AAA servers to group one or more AAA servers, as seen in the bottom of Figure 6-5. (Other configuration settings will then refer to the AAA server group rather than the AAA server.)


Enable AAA Model of Authentication
Configure for Each AAA Server
Create a Group of One or More Servers

```
aaa new-model

tacacs server <server-name>
    address ipv4 <address>
    key <key value>
    port <port number>

aaa group server <group-name>
    server name <server-name-1>
    server name <server-name-2>
```

Figure 6-5 *Enabling AAA and Defining AAA Servers and Groups*

The configuration concepts in Figure 6-5 still have not completed the task of configuring AAA authentication. IOS uses the following additional logic to connect the rest of the logic:

- IOS does login authentication for the console, vty, and aux port, by default, based on the setting of the **aaa authentication login default** global command.
- The **aaa authentication login default** *method1 method2*... global command lists different authentication methods, including referencing a AAA group to be used (as shown at the bottom of Figure 6-5).
- The methods include: a defined AAA group of AAA servers; **local**, meaning a locally configured list of usernames/passwords; or **line**, meaning to use the password defined by the **password** line subcommand.

Basically, when you want to use AAA for login authentication on the console or vty lines, the most straightforward option uses the **aaa authentication login default** command. As Figure 6-6 shows with this command, it lists multiple authentication methods. The switch tries the first method, and if that method returns a definitive answer, the process is done. However, if that method is not available (for instance, none of the AAA servers is reachable over the IP network), IOS on the local device moves on to the next method.

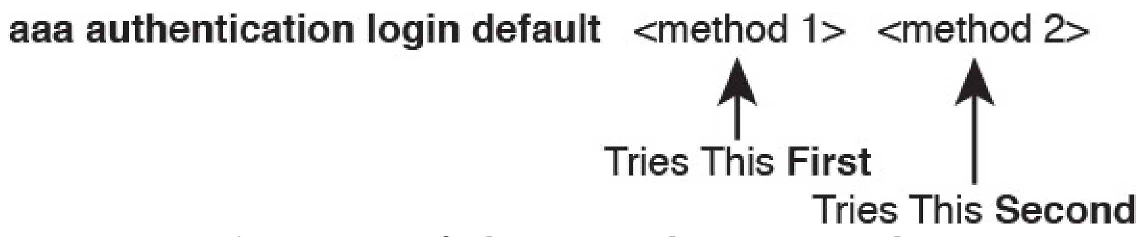


Figure 6-6 *Default Login Authentication Rules*

The idea of defining at least a couple of methods for login authentication makes good sense. For instance, the first method could be a AAA group so that each engineer logs in to each device with that engineer's unique username and password. However, you would not want the engineer to fail to log in just because the IP network is having problems and the AAA servers cannot send packets back to the switch. So, using a backup login method (a second method listed in the command) makes good sense.

Figure 6-7 shows three sample commands for perspective. All three commands reference the same AAA group (WO-AAA-Group). The command labeled with a 1 in the figure takes a shortsighted approach, using only one authentication method with the AAA group. Command 2 in the figure uses two authentication methods: one with AAA and a second method (**local**). (This command's **local** keyword refers to the list of local **username** commands as configured on the local switch.) Command 3 in the figure again uses a AAA group as the first method, followed by the keyword **login**, which tells IOS to use the **password** line subcommand.

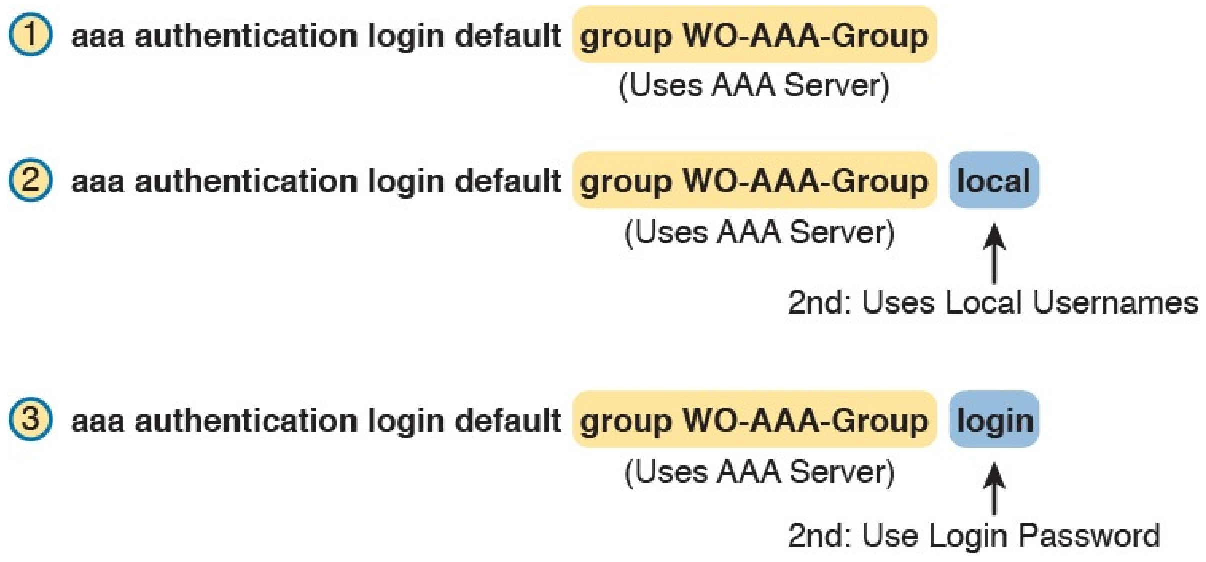

1 aaa authentication login default group WO-AAA-Group
(Uses AAA Server)
2 aaa authentication login default group WO-AAA-Group local
(Uses AAA Server)
2nd: Uses Local Usernames
3 aaa authentication login default group WO-AAA-Group login
(Uses AAA Server)
2nd: Use Login Password

Figure 6-7 *Examples of AAA Login Authentication Method Combinations*

DHCP Snooping

To understand the kinds of risks that exist in modern networks, you have to first understand the rules. Then you have to think about how an attacker might take advantage of those rules in different ways. Some attacks might cause harm, and might be called a denial-of-service (DoS) attack. Or an attacker may gather more data to prepare for some other attack. Whatever the goal, for every protocol and function you learn in networking, there are possible methods to take advantage of those features to give an attacker an advantage.

Cisco chose to add one exam topic for this current CCNA R&S exam that focuses on mitigating attacks based on a specific protocol: DHCP. DHCP has become a very popular protocol, used in most every enterprise, home, and service provider. As a result, attackers have looked for methods to take advantage of DHCP. One way to help mitigate the risks of DHCP is to use a LAN switch feature called DHCP snooping.

This third of four major sections of the chapter works through the basics of DHCP snooping. It starts with the main idea, and then shows one example of how an attacker can misuse DHCP to gain an advantage. The last section explains the logic used by DHCP snooping.

DHCP Snooping Basics

DHCP snooping on a switch acts like a firewall or an ACL in many ways. It will watch for incoming messages on either all ports or some ports (depending on the configuration). It will look for DHCP messages, ignoring all non-DHCP messages and allowing those through. For any DHCP messages, the switch's DHCP snooping logic will make a choice: allow the message or discard the message.

To be clear, DHCP snooping is a Layer 2 switch feature, not a router feature. Specifically, any switch that performs Layer 2 switching, whether it does only Layer 2 switching or acts as a multilayer switch, typically supports DHCP snooping. DHCP snooping must be done on a device that sits between devices in the same VLAN, which is the role of a Layer 2 switch rather than a Layer 3 switch or router.

The first big idea with DHCP snooping is the idea of trusted ports and untrusted ports. To understand why, ponder for a moment all the devices that might be connected to one switch. The list includes routers, servers, and even other switches. It includes end-user devices, such as PCs. It includes wireless access points, which in turn connect to end-user devices. Figure 6-8 shows a representation.

Key Topic

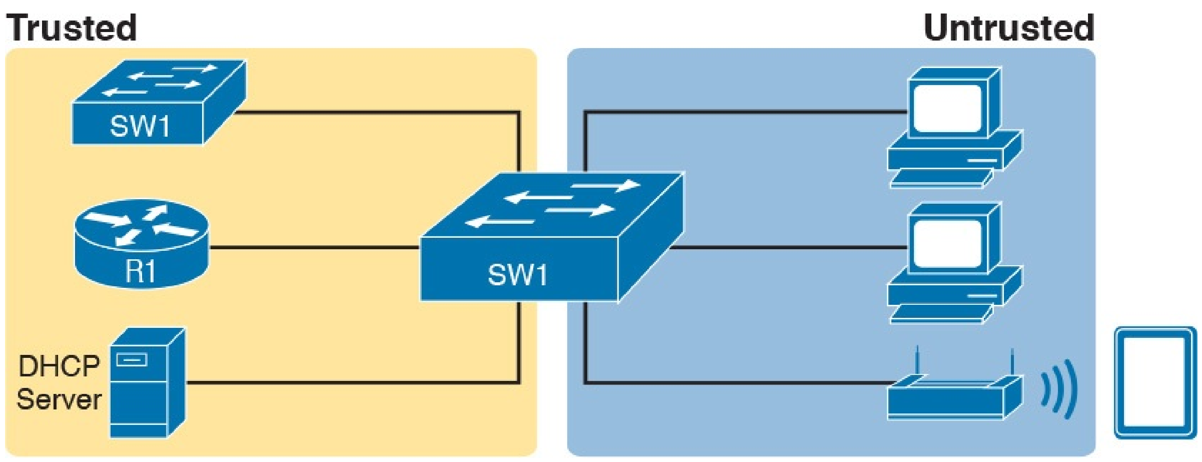


Figure 6-8 *DHCP Snooping Basics: Client Ports Are Untrusted*

DHCP snooping begins with the assumption that end-user devices are untrusted, while devices more within the control of the IT department are trusted. However, a device on an untrusted port can still use DHCP. Instead, making a port untrusted for DHCP snooping means this:

> Watch for incoming DHCP messages, and discard any that are considered to be abnormal for an untrusted port and therefore likely to be part of some kind of attack.

An Example DHCP-based Attack

To give you perspective, Figure 6-9 shows a legitimate user's PC on the far right and the legitimate DHCP sever on the far left. However, an attacker has connected his laptop to the LAN and started his DHCP attack. Remember, PC1's first DHCP message will be a LAN broadcast, so the attacker's PC will receive those LAN broadcasts from any DHCP clients like PC1. (In this case, assume PC1 is attempting to lease an IP address while the attacker is making his attack.)

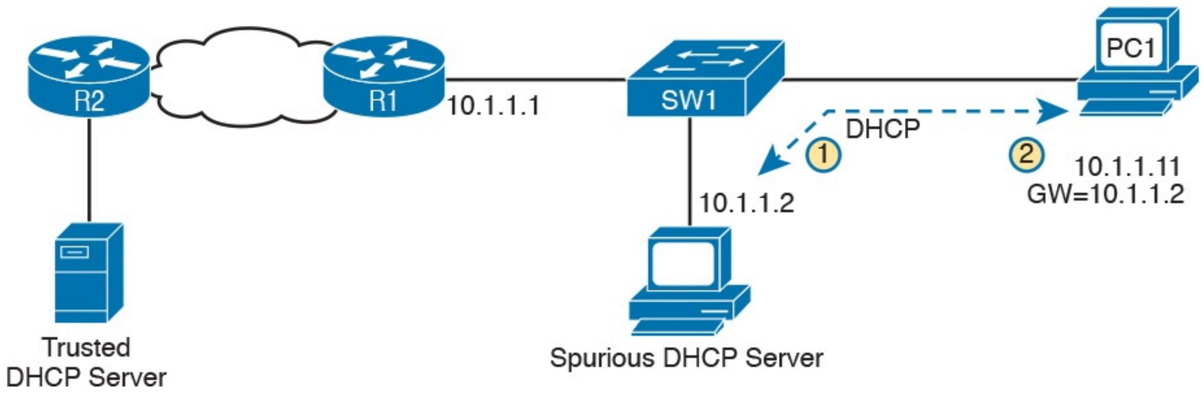


Figure 6-9 *DHCP Attack Supplies Good IP Address but Wrong Default Gateway*

In this example, the DHCP server created and used by the attacker actually leases a useful IP address to PC1, in the correct subnet, with the correct mask.

Why? The attacker wants PC1 to function, but with one twist. Notice the default gateway assigned to PC1: 10.1.1.2, which is the attacker's PC address, rather than 10.1.1.1, which is R1's address. Now PC1 thinks it has all it needs to connect to the network, and it does—but now all the packets sent by PC1 flow first through the attacker's PC, creating a man-in-the-middle attack, as shown in Figure 6-10.

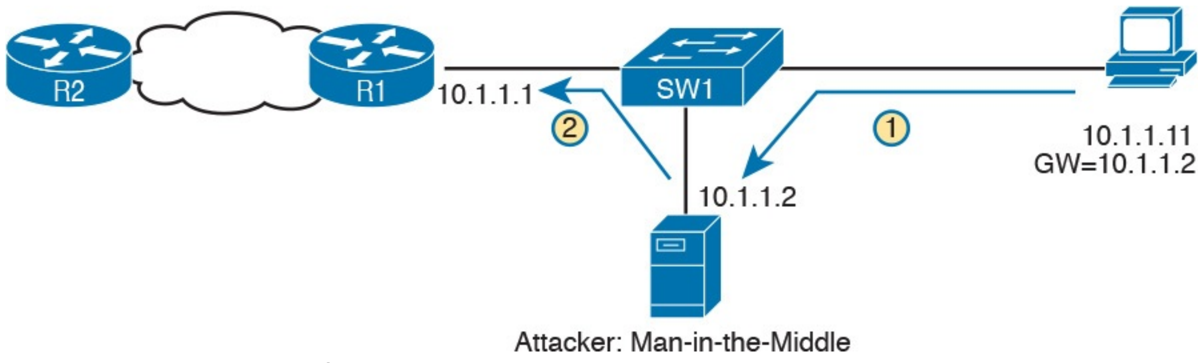


Figure 6-10 *Unfortunate Result: DHCP Attack Leads to Man-in-the-Middle*

The two steps in the figure show data flow once DHCP has completed. For any traffic destined to leave the subnet, PC1 sends its packets to its default gateway, 10.1.1.2, which happens to be the attacker. The attacker forwards the packets to R1. The PC1 user can connect to any and all applications just like normal, but now the attacker can keep a copy of anything sent by PC1.

How DHCP Snooping Works

The preceding example shows just one attack. Some attacks use an extra DHCP server (called a spurious DHCP server), and some attacks happen by using DHCP client functions in different ways. DHCP snooping considers how DHCP should work and filters out any messages that would not be part of a normal use of DHCP.

DHCP snooping needs a few configuration settings. First, the engineer enables DHCP snooping either globally on a switch or by VLAN (that is, enabled on some VLANs, and not on others). Once enabled, all ports are considered untrusted until configured as trusted.

Next, some switch ports need to be configured as trusted. Any switch ports connected to legitimate DHCP servers should be trusted. Additionally, ports connected to other switches, and ports connected to routers, should also be trusted. Why? Trusted ports are basically ports that could receive messages from legitimate DHCP servers in the network. The legitimate DHCP servers in a network are well known.

Just for a quick review, the ICND1 Cert Guide described the DHCP messages used in normal DHCP lease flows (DISCOVER, OFFER, REQUEST, ACK

[DORA]). For these and other DHCP messages, a message is normally sent by either a DHCP client or a server, but not both. In the DORA messages, the client sends the DISCOVER and REQUEST, and the server sends the OFFER and ACK. Knowing that only DHCP servers should send DHCP OFFER and ACK messages, DHCP snooping allows incoming OFFER and ACK messages on trusted ports, but filters those messages if they arrive on untrusted ports.

So, the first rule of DHCP snooping is for the switch to trust any ports on which legitimate messages from trusted DHCP servers might arrive. Conversely, by leaving a port untrusted, the switch is choosing to discard any incoming DHCP server-only messages. Figure 6-11 summarizes these points, with the legitimate DHCP server on the left, on a port marked as trusted.

Key Topic

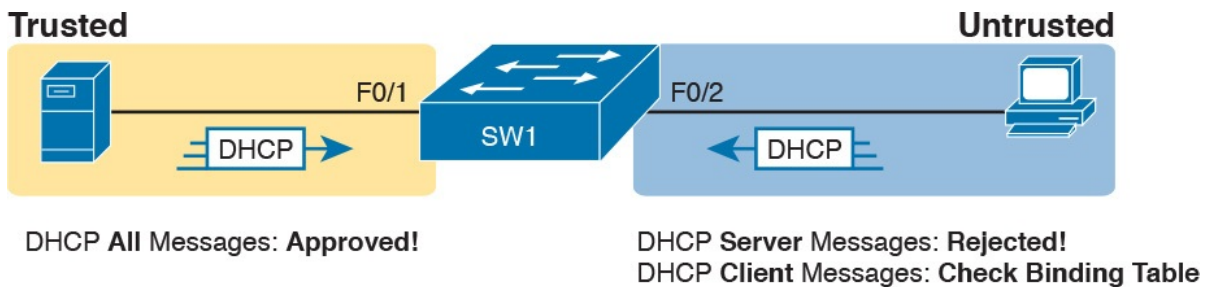
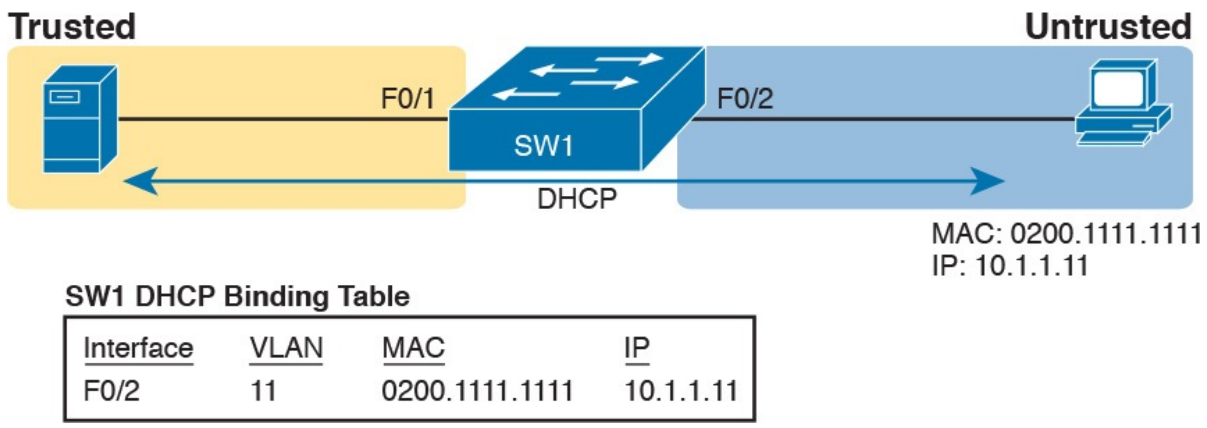

Trusted
Untrusted
F0/1
F0/2
SW1
DHCP
DHCP
DHCP **All** Messages: **Approved!**
DHCP **Server** Messages: **Rejected!**
DHCP **Client** Messages: **Check Binding Table**

Figure 6-11 *Summary of Rules for DHCP Snooping*

The logic for untrusted DHCP ports is a little more challenging. Basically, the untrusted ports are the real user population, all of which rely heavily on DHCP. Those ports also include those few people trying to attack the network with DHCP, and you cannot predict which of the untrusted ports have legitimate users and which are attacking the network. So the DHCP snooping function has to watch the DHCP messages over time, and even keep some state information in a DHCP Binding Table, so that it can decide when a DHCP message should be discarded.

The DHCP Binding Table is a list of key pieces of information about each successful lease of an IPv4 address. Each new DHCP message received on an untrusted port can then be compared to the DHCP Binding Table, and if the switch detects conflicts when comparing the DHCP message to the Binding Table, then the switch will discard the message.

To understand more specifically, first look at Figure 6-12, which shows a switch building one entry in its DHCP Binding Table. In this simple network, the DHCP client on the right leases IP address 10.1.1.11 from the DHCP server on the left. The switch's DHCP snooping feature combines the information from the DHCP messages, with information about the port (interface F0/2, assigned to VLAN 11 by the switch), and puts that in the DHCP Binding Table.



Interface	VLAN	MAC	IP
F0/2	11	0200.1111.1111	10.1.1.11

Figure 6-12 *Legitimate DHCP Client with DHCP Binding Entry Built by DHCP Snooping*

Because of this DHCP binding table entry, DHCP snooping would now prevent another client on another switch port from claiming to be using that same IP address (10.1.1.11) or the same MAC address (2000.1111.1111). (Many DHCP client attacks will use the same IP address or MAC address as a legitimate host.)

Note that beyond firewall-like rules of filtering based on logic, DHCP snooping can also be configured to rate limit the number of DHCP messages on an interface. For instance, by rate limiting incoming DHCP messages on untrusted interfaces, DHCP snooping can help prevent a DoS attack designed to overload the legitimate DHCP server, or to consume all the available DHCP IP address space.

Summarizing DHCP Snooping Features

DHCP snooping can help reduce risk, particularly because DHCP is such a vital part of most networks. The following list summarizes some of the key points about DHCP snooping for easier exam study:

Key Topic

Trusted ports: Trusted ports allow all incoming DHCP messages.

Untrusted ports, server messages: Untrusted ports discard all incoming messages that are considered server messages.

Untrusted ports, client messages: Untrusted ports apply more complex logic for messages considered client messages. They check whether each incoming DHCP message conflicts with existing DHCP binding table information and, if so, discard the DHCP message. If the message has no conflicts, the switch allows the message through, which typically results in the addition of new DHCP Binding Table entries.

Rate limiting: Optionally limits the number of received DHCP messages per second, per port.

Switch Stacking and Chassis Aggregation

Cisco offers several options that allow customers to configure their Cisco switches to act cooperatively to appear as one switch, rather than as multiple switches. This final major section of the chapter discusses two major branches of these technologies: switch stacking, which is more typical of access layer switches, and chassis aggregation, more commonly found on distribution and core switches.

Traditional Access Switching Without Stacking

Imagine for a moment that you are in charge of ordering all the gear for a new campus, with several multistory office buildings. You take a tour of the space, look at drawings of the space, and start thinking about where all the people and computers will be. At some point, you get to the point of thinking about how many Ethernet ports you need in each wiring closet.

Imagine for one wiring closet you need 150 ports today, and you want to build enough switch port capacity to 200 ports. What size switches do you buy? Do you get one switch for the wiring closet, with at least 200 ports in the switch? (The books do not discuss various switch models very much, but yes, you can buy LAN switches with hundreds of ports in one switch.) Or do you buy a pair of switches with at least 100 ports each? Or eight or nine switches with 24 access ports each?

There are pros and cons for using smaller numbers of large switches, and vice versa. To meet those needs, vendors such as Cisco offer switches with a wide range of port densities. However, a switch feature called *switch stacking* gives you some of the benefits of both worlds.

To appreciate the benefits of switch stacking, imagine a typical LAN design like the one shown in Figure 6-13. The figure shows the conceptual design, with two distribution switches and four access layer switches.

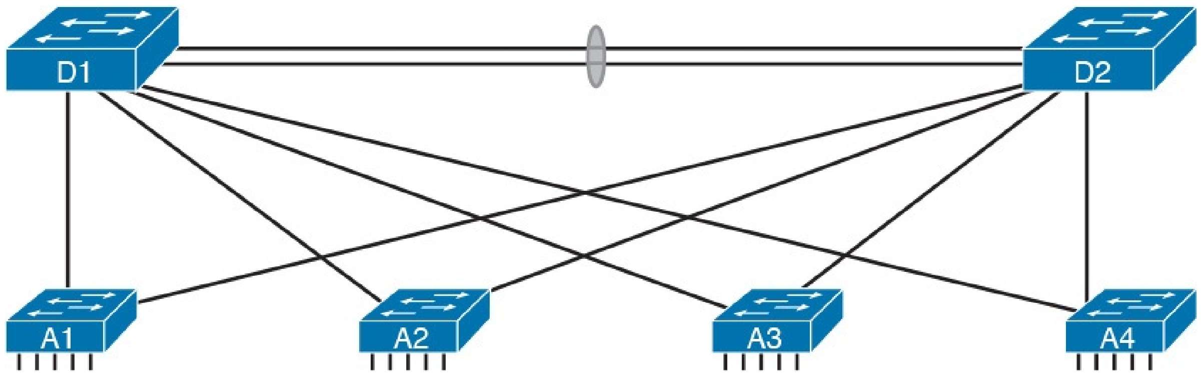


Figure 6-13 *Typical Campus Design: Access Switches and Two Distribution Switches*

For later comparison, let me emphasize a few points here. Access switches A1 through A4 all operate as separate devices. The network engineer must configure each. They each have an IP address for management. They each run CDP, STP, and maybe VTP. They each have a MAC address table, and they each forward Ethernet frames based on that MAC address table. Each switch probably has very similar configuration, but that configuration is separate, and all the functions are separate.

Now picture those same four access layer switches physically, not in Figure 6-13, but as you would imagine them in a wiring closet, even in the same rack. In this case, imagine all four access switches sit in the same rack in the same closet. All the wiring on that floor of the building runs back to the wiring closet, and each cable is patched into some port in one of these four switches. Each switch might be one rack unit (RU) tall (1.75 inches), and they all sit one on top of the other.

Switch Stacking of Access Layer Switches

The scenario described so far is literally a stack of switches one above the other. Switch stacking technology allows the network engineer to make that stack of physical switches act like one switch. For instance, if a switch stack was made from the four switches in Figure 6-13, the following would apply:

Key Topic

- The stack would have a single management IP address.
- The engineer would connect with Telnet or SSH to one switch (with that one management IP address), not multiple switches.
- One configuration file would include all interfaces in all four physical switches.
- STP, CDP, VTP would run on one switch, not multiple switches.
- The switch ports would appear as if all are on the same switch.

- There would be one MAC address table, and it would reference all ports on all physical switches.

The list could keep going much longer for all possible switch features, but the point is that switch stacking makes the switches act as if they are simply parts of a single larger switch.

To make that happen, the switches must be connected together with a special network. The network does not use standard Ethernet ports. Instead, the switches have special hardware ports called *stacking ports*. With the Cisco FlexStack and FlexStack-Plus stacking technology, a stacking module must be inserted into each switch, and then connected with a stacking cable.

Note

Cisco has created a few switch stacking technologies over the years, so to avoid having to refer to them all, note that this section describes Cisco's FlexStack and FlexStack Plus options. These stacking technologies are supported to different degrees in the popular 2960-S, 2960-X, and 2960-XR switch families.

The stacking cables together make a ring between the switches as shown in Figure 6-14. That is, the switches connect in series, with the last switch connecting again to the first. Using full duplex on each link, the stacking modules and cables create two paths to forward data between the physical switches in the stack. The switches use these connections to communicate between the switches to forward frames and to perform other overhead functions.

Key Topic

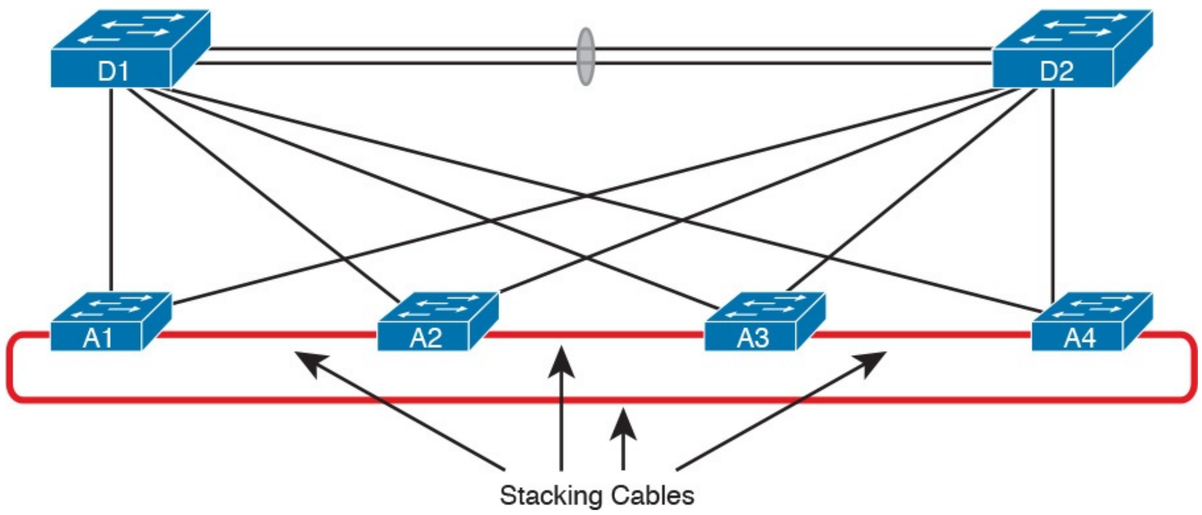


Figure 6-14 *Stacking Cables Between Access Switches in the Same Rack*

Note that each stacking module has two ports with which to connect to another switch's stacking module. For instance, if the four switches were all 2960XR switches, each would need one stacking module, and four cables total to connect the four switches as shown. Figure 6-15 shows the same idea in Figure 6-14, but as a photo that shows the stacking cables on the left side of the figure.

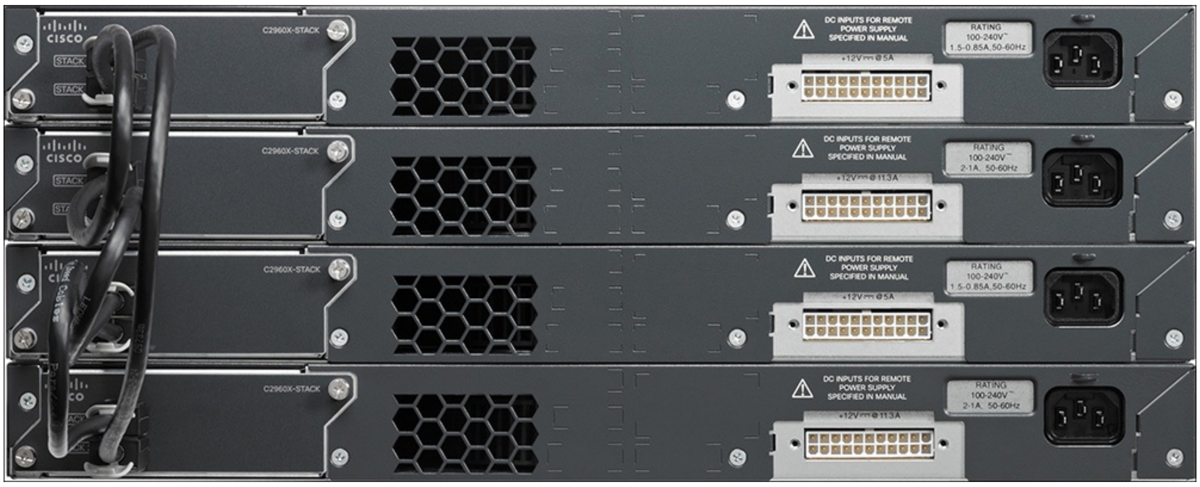


Figure 6-15 *Photo of Four 2960X Switches Cabled on the Left with Stacking Cables*

You should think of switch stacks as literally a stack of switches in the same rack. The stacking cables are short, with the expectation that the switches sit together in the same room and rack. For instance, Cisco offers stacking cables of .5, 1, and 3 meters long for the FlexStack and FlexStack-Plus stacking technology discussed in more depth at the end of this section.

Switch Stack Operation as a Single Logical Switch

With a switch stack, the switches together act as one *logical switch*. This term (logical switch) is meant to emphasize that there are obviously physical switches, but they act together as one switch.

To make it all work, one switch acts as a *stack master* to control the rest of the switches. The links created by the stacking cables allow the physical switches to communicate, but the stack master is in control of the work. For instance, if you number the physical switches as 1, 2, 3, and 4, a frame might arrive on switch 4 and need to exit a link on switch 3. If switch 1 were the stack master, switches 1, 3, and 4 would all need to communicate over the stack links to forward that frame. But switch 1, as stack master, would do the matching of the MAC address table to choose where to forward the frame.

Figure 6-16 focuses on the LAN design impact of how a switch stack acts like one logical switch. The figure shows the design with no changes to the cabling between the four access switches and the distribution switches.

Formerly, each separate access switch had two links to the distribution layer: one connected to each distribution switch (see Figure 6-13). That cabling is unchanged. However, acting as one logical switch, the switch stack now operates as if it is one switch, with four uplinks to each distribution switch. Add a little configuration to put each set of four links into an EtherChannel, and you have the design shown in Figure 6-16.

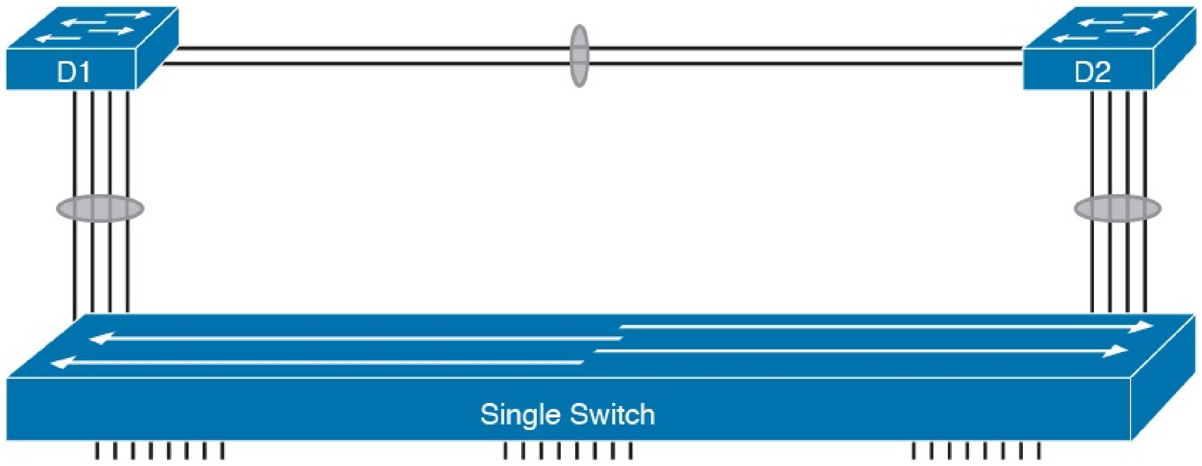


Figure 6-16 *Stack Acts Like One Switch*

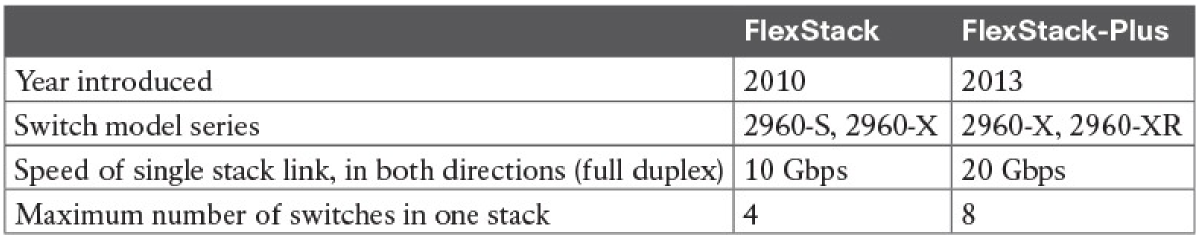
The stack also simplifies operations. Imagine for instance the scope of an STP topology for a VLAN that has access ports in all four of the physical access switches in this example. That Spanning Tree would include all six switches. With the switch stack acting as one logical switch, that same VLAN now has only three switches in the STP topology, and is much easier to understand and predict.

Cisco FlexStack and FlexStack-Plus

Just to put a finishing touch on the idea of a switch stack, this closing topic examines a few particulars of Cisco's FlexStack and FlexStack-Plus stacking options.

Cisco's stacking technologies require that Cisco plan to include stacking as a feature in a product, given that it requires specific hardware. Cisco has a long history of building new model series of switches with model numbers that begin with 2960. Per Cisco's documentation, Cisco created one stacking technology, called FlexStack, as part of the introduction of the 2960-S model series. Cisco later enhanced FlexStack with FlexStack-Plus, adding support with products in the 2960-X and 2960-XR model series. For switch stacking to support future designs, the stacking hardware tends to increase over time as well, as seen in the comparisons between FlexStack and FlexStack-Plus in Table 6-3.

Key Topic



	FlexStack	FlexStack-Plus
Year introduced	2010	2013
Switch model series	2960-S, 2960-X	2960-X, 2960-XR
Speed of single stack link, in both directions (full duplex)	10 Gbps	20 Gbps
Maximum number of switches in one stack	4	8

Table 6-3 Comparisons of Cisco’s FlexStack and FlexStack-Plus Options

Chassis Aggregation

The term *chassis aggregation* refers to another Cisco technology used to make multiple switches operate as a single switch. From a big picture perspective, switch stacking is more often used and offered by Cisco in switches meant for the access layer. Chassis aggregation is meant for more powerful switches that sit in the distribution and core layers. Summarizing some of the key differences, chassis aggregation

- Typically is used for higher-end switches used as distribution or core switches
- Does not require special hardware adapters, instead using Ethernet interfaces
- Aggregates two switches
- Arguably is more complex but also more functional

The big idea of chassis aggregation is the same as for a switch stack: make multiple switches act like one switch, which gives you some availability and design advantages. But much of the driving force behind chassis aggregation is about high-availability design for LANs. This section works through a few of those thoughts to give you the big ideas about the thinking behind high availability for the core and distribution layer.

Note

This section looks at the general ideas of chassis aggregation, but for further reading about a specific implementation, search at Cisco.com for Cisco’s Virtual Switching System (VSS) that is supported on 6500 and 6800 series switches.

High Availability with a Distribution/Core Switch

Even without chassis aggregation, the distribution and core switches need to have high availability. The next few pages look at how the switches built for use as distribution and core switches can help improve availability, even without chassis aggregation.

If you were to look around a medium to large enterprise campus LAN, you would typically find many more access switches than distribution and core switches. For instance, you might have four access switches per floor, with ten floors in a building, for 40 access switches. That same building probably has only a pair of distribution switches.

And why two distribution switches instead of one? Because if the design used only one distribution switch, and it failed, none of the devices in the building could reach the rest of the network. So, if two distribution switches are good, why not four? Or eight? One reason is cost, another complexity. Done right, a pair of distribution switches for a building can provide the right balance of high availability and low cost/complexity.

The availability features of typical distribution and core switches allow network designers to create a great availability design with just two distribution or core switches. Cisco makes typical distribution/core switches with more redundancy. For instance, Figure 6-17 shows a representation of a typical chassis-based Cisco switch. It has slots that can be used for line cards —that is, cards with Ethernet ports. It has dual supervisor cards that do frame and packet forwarding. And it has two power supplies, each of which can be connected to power feeds from different electrical substations if desired.

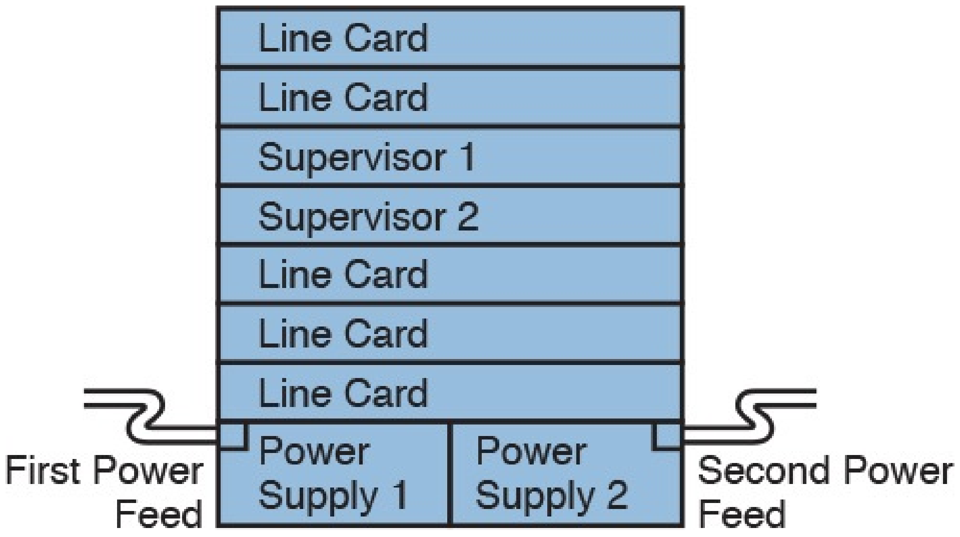


Figure 6-17 *Common Line-Card Arrangement in a Modular Cisco Distribution/Core Switch*

Now imagine two distribution switches sitting beside each other in a wiring closet as shown in Figure 6-18. A design would usually connect the two switches with an EtherChannel. For better availability, the EtherChannel could use ports from different line cards, so that if one line card failed due to some hardware problem, the EtherChannel would still work.

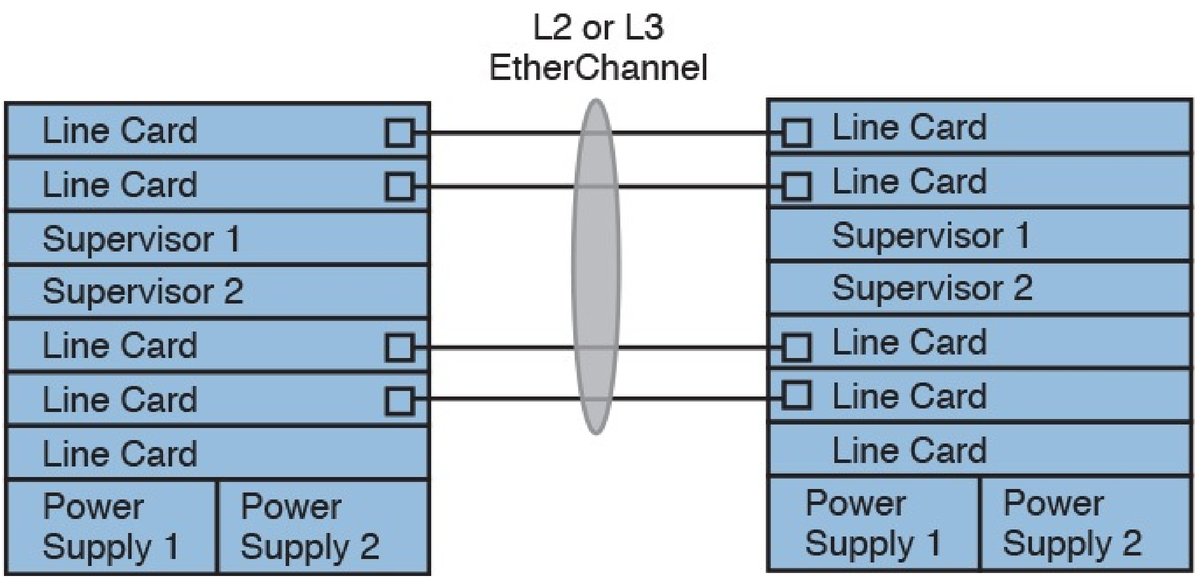


Figure 6-18 *Using EtherChannel and Different Line Cards*

Improving Design and Availability with Chassis Aggregation

Next, consider the effect of adding chassis aggregation to a pair of distribution switches. In terms of effect, the two switches act as one switch, much like switch stacking. The particulars of how chassis aggregation achieves that differs.

Figure 6-19 shows a comparison. On the left, the two distribution switches act independently, and on the right, the two distribution switches are aggregated. In both cases, each distribution switch connects with a single Layer 2 link to the access layer switches A1 and A2, which act independently—that is, they do not use switch stacking. So, the only difference between the left and right examples is that on the right the distribution switches use switch aggregation.

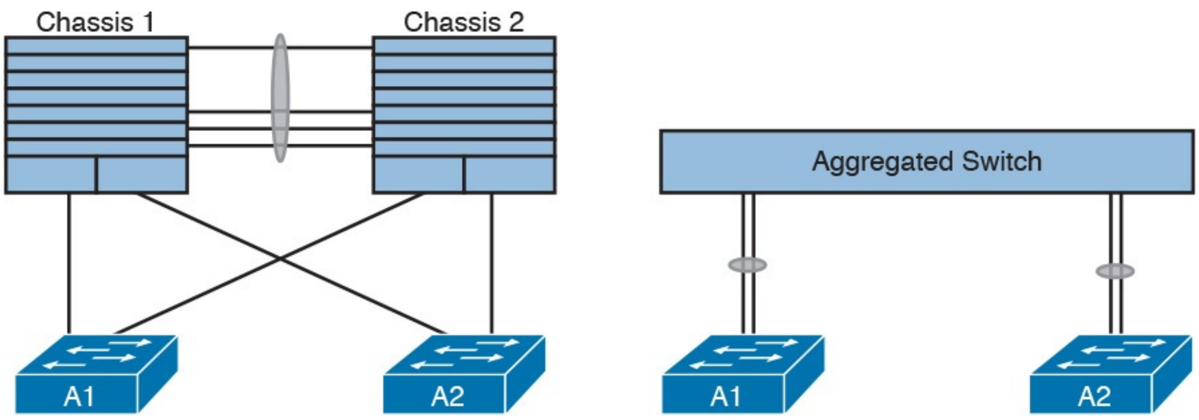


Figure 6-19 *One Design Advantage of Aggregated Distribution Switches*

The right side of the figure shows the aggregated switch that appears as one switch to the access layer switches. In fact, even though the uplinks connect into two different switches, they can be configured as an EtherChannel through a feature called *Multichassis EtherChannel* (MEC).

The following list describes some of the advantages of using switch aggregation. Note that many of the benefits should sound familiar from the switch stacking discussion. The one difference in this list has to do with the active/active data plane.

Key Topic

- **Multichassis EtherChannel (MEC):** Uses the EtherChannel between the two physical switches.
- **Active/Standby Control Plane:** Simpler operation for control plane because the pair acts as one switch for control plane protocols: STP, VTP, EtherChannel, ARP, routing protocols.
- **Active/Active data plane:** Takes advantage of forwarding power of supervisors on both switches, with active Layer 2 and Layer 3 forwarding the supervisors of both switches. The switches synchronize their MAC and routing tables to support that process.
- **Single switch management:** Simpler operation of management protocols by running management protocols (Telnet, SSH, SNMP) on the active switch; configuration is synchronized automatically with the standby switch.

Finally, using chassis aggregation and switch stacking together in the same network has some great design advantages. Look back to Figure 6-13 at the beginning of this section. It showed two distribution switches and four access switches, all acting independently, with one uplink from each access switch to each distribution switch. If you enable switch stacking for the four access switches, and enable chassis aggregation for the two distribution switches, you end up with a topology as shown in Figure 6-20.

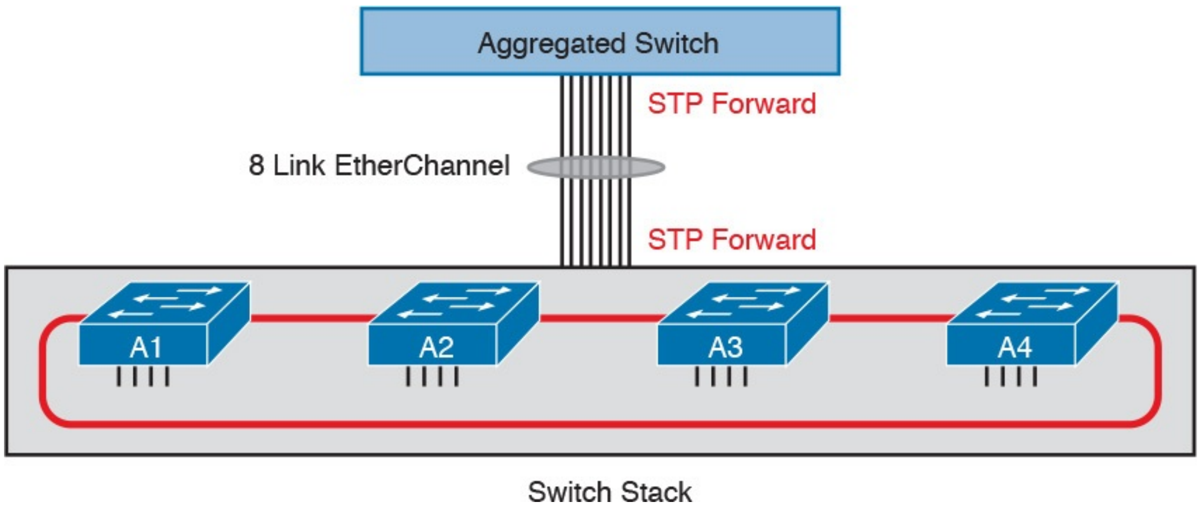


Figure 6-20 *Making Six Switches Act like Two*

Chapter Review

One key to doing well on the exams is to perform repetitive spaced review sessions. Review this chapter's material using either the tools in the book, DVD, or interactive tools for the same material found on the book's companion website. Refer to the "Your Study Plan" element for more details. Table 6-4 outlines the key review elements and where you can find them. To better track your study progress, record when you completed these activities in the second column.

Review Element	Review Date(s)	Resource Used
Review key topics		Book, DVD/website
Review key terms		Book, DVD/website
Answer DIKTA questions		Book, PCPT
Review memory tables		Book, App

Table 6-4 Chapter Review Tracking

Review All the Key Topics

Key Topic

Key Topic Element	Description	Page Number
Figure 6-3	Protocols used by IEEE 802.1x	146
Table 6-2	Comparisons of TACACS+ and RADIUS	148
Figure 6-8	Concept: Trusted and untrusted ports for DHCP snooping	151
Figure 6-11	Summary of DHCP snooping filtering actions	153
List	Key points about DHCP snooping	154
List	Common Cisco switch stack features	156
Figure 6-14	Switch stacking cabling	157
Table 6-3	Comparisons of FlexStack and FlexStack-Plus	158
List	Key features of switch aggregation	161

Table 6-5 Key Topics for Chapter 6

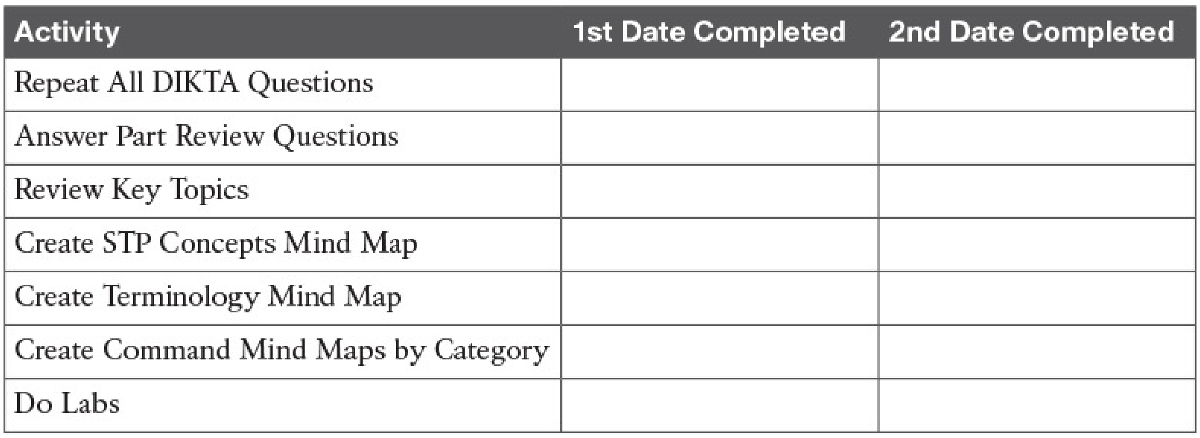
Key Terms You Should Know

FlexStack
FlexStack-Plus
stacking module
stacking cable
switch stacking
chassis aggregation
Multichassis EtherChannel

trusted port
untrusted port
DHCP Binding Table
DHCP snooping
AAA server
Extensible Authentication Protocol (EAP)
EAP over LAN (EAPoL)
supplicant
authenticator

Part I Review

Keep track of your part review progress with the checklist shown in Table P1-1. Details about each task follow the table.



Activity	1st Date Completed	2nd Date Completed
Repeat All DIKTA Questions		
Answer Part Review Questions		
Review Key Topics		
Create STP Concepts Mind Map		
Create Terminology Mind Map		
Create Command Mind Maps by Category		
Do Labs		

Table P1-1 Part I Part Review Checklist

Repeat All DIKTA Questions

For this task, answer the "Do I Know This Already?" questions again for the chapters in this part of the book using the PCPT software. See the section "How to View Only DIKTA Questions by Chapter or Part" in the Introduction to this book to learn how to make the PCPT software show you DIKTA questions for this part only.

Answer Part Review Questions

For this task, answer the Part Review questions for this part of the book using the PCPT software. Refer to the Introduction to this book, in the section "How to View Part Review Questions," for more details.

Review Key Topics

Review all Key Topics in all chapters in this part, either by browsing the chapters or by using the Key Topics application on the DVD or companion website.

Create STP Concepts Mind Map

Spanning Tree Protocol (STP) defines a lot of ideas that you might find hard to mentally organize. Create a mind map to help your brain organize the various STP concepts. Some suggestions as to how to organize the concepts:

> **Rules:** This section might include any of the rules a switch uses when making choices. For instance, the rules switches use for

choosing a root switch.

Roles: STP defines both roles and states; an example of a role is the root port role.

States: For example, forwarding and blocking.

Within each of these sections, break down the details based on 802.1D STP and 802.1w RSTP.

Create Terminology Mind Map

The chapters in this part weave in and out of different topics. Without looking back at the chapters or your notes, create a mind map with all the terminology you can recall from Part I of the book. Your job is as follows:

- Think of every term that you can remember from Part I of the book.
- Organize the terms into these divisions: VLANs, VLAN trunks, STP, VTP, AAA, 802.1x, DHCP snooping, switch stacking/aggregation.
- After you have written every term you can remember into the mind map, review the Key Terms list at the end of Chapters 1 through 6. Add any terms you forgot to your mind map.

Create Command Mind Maps by Category

Part I of this book also introduced both configuration and EXEC commands. Create one mind map (or a section of a larger mind map) for each of the categories of commands in this list:

VLANs, 802.1Q trunking, STP/RSTP, EtherChannel, VTP

For each category, think of all configuration commands and all EXEC commands (mostly **show** commands). For each category, group the configuration commands separately from the EXEC commands. Figure P1-1 shows a sample for IPv4 commands on a switch.

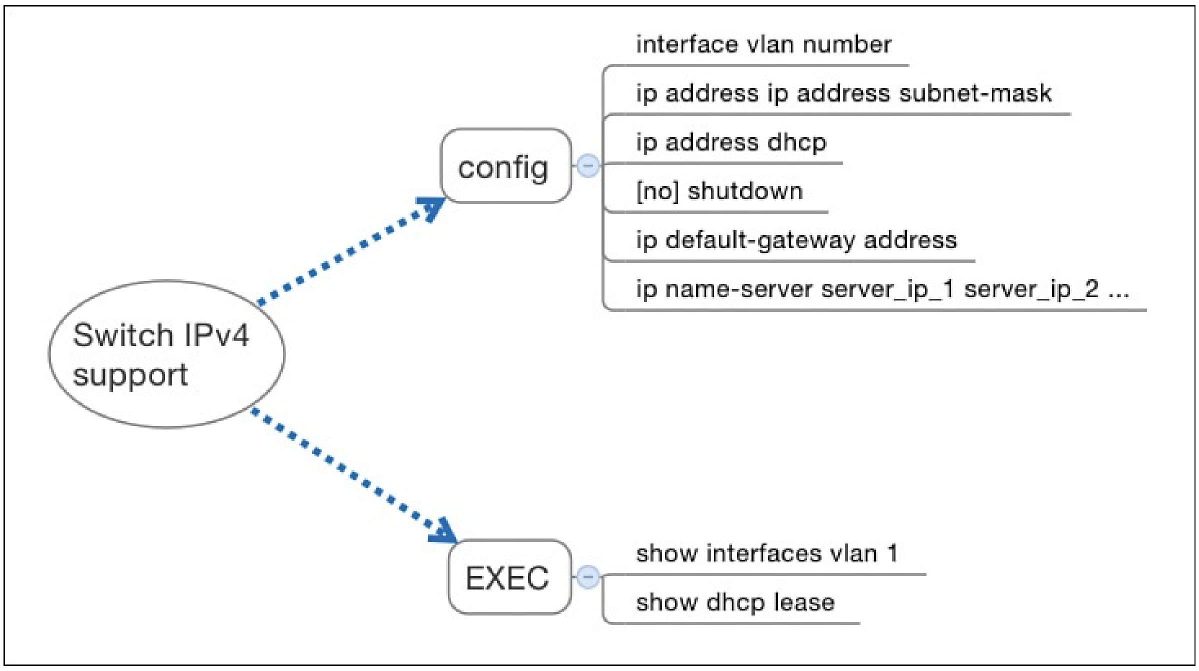


Figure P1-1 *Sample Command Mind Map*

Note

For more information on mind mapping, refer to the Introduction, in the section "About Mind Maps."

Appendix E, "Mind Map Solutions," lists sample mind map answers. If you do choose to use mind map software, rather than paper, you might want to remember where you stored your mind map files. Table P1-2 lists the mind maps for this part review and a place to record those filenames.

Map	Description	Where You Saved It
1	STP Concepts Mind Map	
2	Terminology Mind Map	
3	Command Mind Maps	

Table P1-2 Configuration Mind Maps for Part I Review

Do Labs

If you have not done so, make your choices about what lab tools you intend to use and experiment with the commands in these chapters. Re-create examples in the chapters, and try all the **show** commands; the **show** commands are very important for answering simlet questions.

Sim Lite: You can use the Pearson Network Simulator Lite included with this book to do some labs and get used to the CLI. All the labs in the ICND2 Lite product are about topics in this part of the book, so make sure and work through those labs to start learning about the CLI.

Pearson Network Simulator: If you use the full Pearson CCNA simulator, focus more on the configuration scenario and troubleshooting scenario labs associated with the topics in this part of the book. These types of labs include a larger set of topics, and work well as Part Review activities. (See the Introduction for some details about how to find which labs are about topics in this part of the book.)

Config Labs: In your idle moments, review and repeat any of the Config Labs for this book part in the author's blog; launch from http://blog.certskills.com/ccna and navigate to **Hands-On > Config Lab**.