Another way to improve the user experience at the console is to control timeouts from the console. By default, the switch automatically disconnects console and vty (Telnet and SSH) users after 5 minutes of inactivity. The **exec-timeout** *minutes seconds* line subcommand lets you set the length of that inactivity timer, with the special value of 0 minutes and 0 seconds meaning "never time out."

Example 8-8 shows the syntax for these two commands, both on the console line. Note that both can be applied to the vty lines as well, for the same reasons.

Example 8-8 *Defining Console Inactivity Timeouts and When to Display Log Messages*

```
line console 0
 login
 password cisco
 exec-timeout 0 0
 logging synchronous
```

NOTE This concludes the first half of this chapter. If you have not yet tried any commands on a router or switch, now would be a good time to pause from your reading and try some. If you have real gear, or the Pearson Simulator, do some labs about navigating the CLI, setting passwords, and other basic administration. If not, watch the videos from the DVD on CLI navigation and route configuration. Also, try a few labs from the ICND1 Simulator Lite on the DVD. Even if you do a lab on something you have not seen yet, you can get a little better idea about how to move around with the command-line interface.

LAN Switch Configuration and Operation

Cisco switches work very well when received from the factory, without any configuration added. Cisco switches leave the factory with default settings, with all interfaces enabled (a default configuration of **no shutdown**) and with autonegotiation enabled for ports that can use it (a default configuration of **duplex auto** and **speed auto**). All interfaces default to be part of VLAN 1 (**switchport access vlan 1**). All you have to do with a new Cisco switch is make all the physical connections—Ethernet cables and power cord—and the switch starts working.

In most enterprise networks, you will want the switch to operate with some different settings as compared with the factory defaults. The second half of this chapter discusses some of those settings, with Chapter 9 ("Implementing Ethernet Virtual LANs") discussing more. (Also note that the details in this section differ from the configuration on a router.) In particular, this section covers the following:

- IP for remote access
- Interface configuration (including speed and duplex)
- Port security
- Securing unused switch interfaces

Enabling IP for Remote Access

To allow Telnet or SSH access to the switch, and to allow other IP-based management protocols (for example, Simple Network Management Protocol) to function as intended, the switch needs an IP address. The IP address has nothing to do with how switches forward Ethernet frames; it simply exists to support overhead management traffic.

A switch's IP configuration works like a PC with a single Ethernet interface. For perspective, a PC has a CPU, with the operating system running on the CPU. It has an Ethernet network interface card (NIC). The OS configuration includes an IP address associated with the NIC, either configured or learned dynamically with DHCP. To support IP, the switch has the equivalent settings.

A switch uses concepts similar to a host, except that the switch can use a virtual NIC. Like a PC, a switch has a real CPU, running an OS (called IOS). The switch then uses a NIC-like concept called a *switched virtual interface (SVI)*, or more commonly, a *VLAN interface*, that acts like the switch's own NIC for connecting into a LAN to send IP packets. Like a host, the switch configuration assigns IP settings, like an IP address, to this VLAN interface, as seen in Figure 8-5.

Key Topic

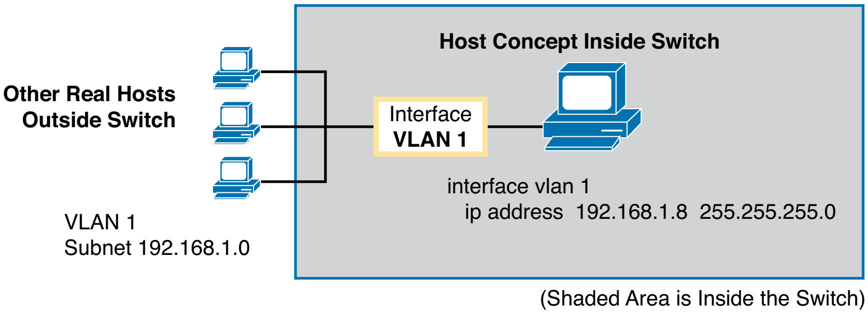

Host Concept Inside Switch
Other Real Hosts Outside Switch
Interface VLAN 1
interface vlan 1
ip address 192.168.1.8 255.255.255.0
VLAN 1
Subnet 192.168.1.0
(Shaded Area is Inside the Switch)

Figure 8-5 *Switch Virtual Interface (SVI) Concept Inside a Switch*

A typical Layer 2 Cisco LAN switch can use only one VLAN interface at a time, but the network engineer can choose which VLAN interface, putting the switch's management traffic into a particular VLAN. For example, Figure 8-6 shows a switch with some physical ports in two different VLANs (1 and 2). The network engineer needs to choose whether the switch IP address, used to access and manage the switch, should have an IP address in subnet 192.168.1.0 (in VLAN 1), or in subnet 192.168.2.0 (in VLAN 2).

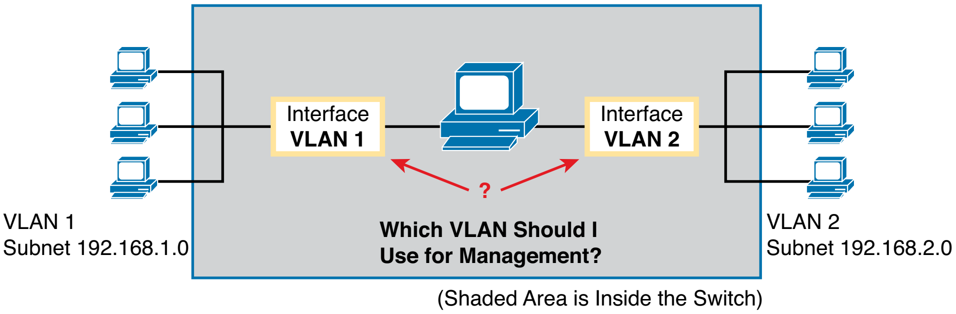


Figure 8-6 *Choosing One VLAN on Which to Configure a Switch IP Address*

NOTE Some Cisco switches, called *Layer 2 switches*, forward Ethernet frames as discussed in depth in Chapter 6, "Building Ethernet LANs with Switches." Other Cisco switches, called *multilayer switches* or *Layer 3 switches*, can also route IP packets using the Layer 3 logic normally used by routers. Layer 3 switches configure IP addresses on more than one VLAN interface at a time. This chapter assumes all switches are Layer 2 switches. Chapter 9 further defines the differences between these types of LAN switches.

Configuring IPv4 on a Switch

A switch configures its IPv4 address and mask on this special NIC-like *VLAN interface*. The following steps list the commands used to configure IPv4 on a switch, assuming that the IP address is configured to be in VLAN 1, with Example 8-9 that follows showing an example configuration.

Step 1. Enter VLAN 1 configuration mode using the **interface vlan 1** global configuration command.

Step 2. Assign an IP address and mask using the **ip address** *ip-address mask* interface subcommand.

Step 3. If not already enabled, enable the VLAN 1 interface using the **no shutdown** interface subcommand.

Step 4. Add the **ip default-gateway** *ip-address* global command to configure the default gateway.

Step 5. (Optional) Add the **ip name-server** *ip-address1 ip-address2 . . .* global command to configure the switch to use DNS to resolve names into their matching IP address.

Example 8-9 *Switch Static IP Address Configuration*

```
Emma# configure terminal
Emma(config)# interface vlan 1
Emma(config-if)# ip address 192.168.1.200 255.255.255.0
Emma(config-if)# no shutdown
00:25:07: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:25:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
  state to up
Emma(config-if)# exit
Emma(config)# ip default-gateway 192.168.1.1
```

On a side note, this example shows a particularly important and common command: the **[no] shutdown** command. To administratively enable an interface on a switch, use the **no shutdown** interface subcommand; to disable an interface, use the **shutdown** interface subcommand. The messages shown in Example 8-9, immediately following the **no shutdown** command, are syslog messages generated by the switch stating that the switch did indeed enable the interface.

The switch can also use DHCP to dynamically learn its IPv4 settings. Basically, all you have to do is tell the switch to use DHCP on the interface, and enable the interface. Assuming that DHCP works in this network, the switch will learn all its settings. The following list details the steps, again assuming the use of interface VLAN 1, with Example 8-10 that follows showing an example.

Step 1. Enter VLAN 1 configuration mode using the **interface vlan 1** global configuration command, and enable the interface using the **no shutdown** command as necessary.

Step 2. Assign an IP address and mask using the **ip address dhcp** interface subcommand.

Example 8-10 *Switch Dynamic IP Address Configuration with DHCP*

```
Emma# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Emma(config)# interface vlan 1
Emma(config-if)# ip address dhcp
Emma(config-if)# no shutdown
Emma(config-if)# ^Z
Emma#
00:38:20: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:38:21: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
```

Verifying IPv4 on a Switch

The switch IPv4 configuration can be checked in several places. First, you can always look at the current configuration using the **show running-config** command. Second, you can look at the IP address and mask information using the **show interface vlan** *x* command, which shows detailed status information about the VLAN interface in VLAN *x*. Finally, if using DHCP, use the **show dhcp lease** command to see the (temporarily) leased IP address and other parameters. (Note that the switch does not store the DHCP-learned IP configuration in the running-config file.) Example 8-11 shows sample output from these commands to match the configuration in Example 8-10.

Example 8-11 *Verifying DHCP-learned Information on a Switch*

```
Emma# show dhcp lease
Temp IP addr: 192.168.1.101  for peer on Interface: Vlan1
Temp  sub net mask: 255.255.255.0
   DHCP Lease server: 192.168.1.1, state: 3 Bound
   DHCP transaction id: 1966
   Lease: 86400 secs,  Renewal: 43200 secs,  Rebind: 75600 secs
Temp default-gateway addr: 192.168.1.1
   Next timer fires after: 11:59:45
   Retry count: 0   Client-ID: cisco-0019.e86a.6fc0-Vl1
   Hostname: Emma
Emma# show interfaces vlan 1
Vlan1 is up, line protocol is up
  Hardware is EtherSVI, address is 0019.e86a.6fc0 (bia 0019.e86a.6fc0)
  Internet address is 192.168.1.101/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
! lines omitted for brevity
Emma# show ip default-gateway
192.168.1.1
```

The output of the **show interfaces vlan 1** command lists two very important details related to switch IP addressing. First, this **show** command lists the interface status of the VLAN 1 interface—in this case, “up and up.” If the VLAN 1 interface is not up, the switch cannot use its IP address to send and receive traffic. Notably, if you forget to issue the **no shutdown** command, the VLAN 1 interface remains in its default shutdown state and is listed as “administratively down” in the **show** command output.

Second, note that the output lists the interface's IP address on the third line. If you statically configure the IP address, as in Example 8-9, the IP address will always be listed. However, if you use DHCP, and DHCP fails, the **show interfaces vlan** *x* command will not list an IP address here. When DHCP works, you can see the IP address with this command, but it does not remind you whether the address is either statically configured or DHCP leased.

Configuring Switch Interfaces

IOS uses the term *interface* to refer to physical ports used to forward data to and from other devices. Each interface can be configured with several settings, each of which might differ from interface to interface.

IOS uses interface subcommands to configure these settings. For example, interfaces can be configured to use the **duplex** and **speed** interface subcommands to configure those settings statically, or an interface can use autonegotiation (the default). Example 8-12 shows how to configure duplex and speed, as well as the **description** command, which is simply a text description that can be configured by the administrator.

Example 8-12 *Interface Configuration Basics*

```
Emma# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Emma(config)# interface FastEthernet 0/1
Emma(config-if)# duplex full
Emma(config-if)# speed 100
Emma(config-if)# description Server1 connects here
Emma(config-if)# exit
Emma(config)# interface range FastEthernet 0/11 - 20
Emma(config-if-range)# description end-users connect_here
Emma(config-if-range)# ^Z
Emma#
Emma# show interfaces status
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1     Server1 connects h notconnect   1            full    100 10/100BaseTX
Fa0/2                        notconnect   1            auto   auto 10/100BaseTX
Fa0/3                        notconnect   1            auto   auto 10/100BaseTX
Fa0/4                        connected    1          a-full  a-100 10/100BaseTX
Fa0/5                        notconnect   1            auto   auto 10/100BaseTX
Fa0/6                        connected    1          a-full  a-100 10/100BaseTX
Fa0/7                        notconnect   1            auto   auto 10/100BaseTX
Fa0/8                        notconnect   1            auto   auto 10/100BaseTX
Fa0/9                        notconnect   1            auto   auto 10/100BaseTX
Fa0/10                       notconnect   1            auto   auto 10/100BaseTX
Fa0/11    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/12    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/13    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/14    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/15    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/16    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/17    end-users connect  notconnect   1            auto   auto 10/100BaseTX
```

```
Fa0/18    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/19    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/20    end-users connect  notconnect   1            auto   auto 10/100BaseTX
Fa0/21                       notconnect   1            auto   auto 10/100BaseTX
Fa0/22                       notconnect   1            auto   auto 10/100BaseTX
Fa0/23                       notconnect   1            auto   auto 10/100BaseTX
Fa0/24                       notconnect   1            auto   auto 10/100BaseTX
Gi0/1                        notconnect   1            auto   auto 10/100/1000BaseTX
Gi0/2                        notconnect   1            auto   auto 10/100/1000BaseTX
```

You can see some of the details of interface configuration with both the **show running-config** command (not shown in the example) and the handy **show interfaces status** command. This command lists a single line for each interface, the first part of the interface description, and the speed and duplex settings. Several of the early entries in the output purposefully show some differences, as follows:

FastEthernet 0/1 (Fa0/1): This output lists the configured speed of 100 and duplex full; however, it lists a status of notconnect. The notconnect status means that the physical link is not currently working, including reasons like no cable being connected, the other device being powered off, or the other device putting the port in a shutdown state. In this case, no cable had been installed when the output was gathered.

FastEthernet 0/2 (Fa0/2): This port also has no cable installed yet, but it uses all default configuration. So, the highlighted output shows this interface with the default setting of auto (meaning autonegotiate).

FastEthernet 0/4 (Fa0/4): Like Fa0/2, this port has all default configuration, but was cabled to another device that is up, causing the status to be listed as "connect." This device also completed the autonegotiation process, so the output lists the resulting speed and duplex (**a-full** and **a-100**), in which the **a-** refers to the fact that these values were autonegotiated.

Also, note that for the sake of efficiency, you can configure a command on a range of interfaces at the same time using the **interface range** command. In the example, the **interface range FastEthernet 0/11 - 20** command tells IOS that the next subcommand(s) apply to interfaces Fa0/11 through Fa0/20.

NOTE Configuring both the speed and duplex on a Cisco switch interface disables autonegotiation.

Port Security

If the network engineer knows what devices should be cabled and connected to particular interfaces on a switch, the engineer can use *port security* to restrict that interface so that only the expected devices can use it. This reduces exposure to attacks in which the attacker connects a laptop to some unused switch port. When that inappropriate device attempts to send frames to the switch interface, the switch can take different actions, ranging from simply issuing informational messages to effectively shutting down the interface.

Port security identifies devices based on the source MAC address of Ethernet frames the devices send. For example, in Figure 8-7, PC1 sends a frame, with PC1's MAC address as the source address. SW1's F0/1 interface can be configured with port security, and if so, SW1 would think about PC1's MAC address and whether PC1 was allowed to send frames into port F0/1.

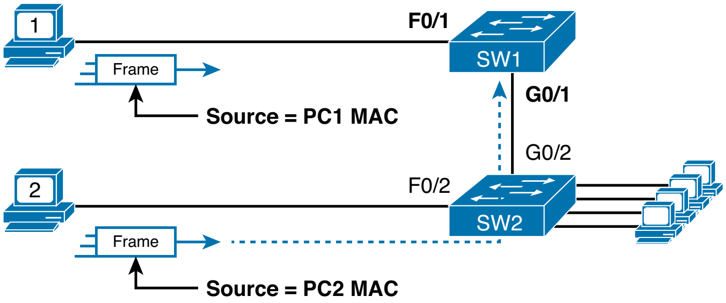


Figure 8-7 *Source MAC Addresses in Frames as They Enter a Switch*

Port security also has no restrictions on whether the frame came from a local device or it was forwarded through other switches. For example, switch SW1 could use port security on its G0/1 interface, checking the source MAC address of the frame from PC2, when forwarded up to SW1 from SW2.

Port security has several flexible options, but all operate with the same core concepts. First, switches enable port security per port, with different settings available per port. Each port has a maximum number of allowed MAC addresses, meaning that for all frames entering that port, only that number of *different* source MAC addresses can be used in different incoming frames before port security thinks a violation has occurred. When a frame with a new source MAC address arrives, pushing the number of MAC addresses past the allowed maximum, a port security violation occurs. At that point, the switch takes action—by default, discarding all future incoming traffic on that port.

The following list summarizes these ideas common to all variations of port security:

Key Topic

- Define a maximum number of source MAC addresses allowed for all frames coming in the interface.
- Watch all incoming frames, and keep a list of all source MAC addresses, plus a counter of the number of different source MAC addresses.
- When adding a new source MAC address to the list, if the number of MAC addresses pushes past the configured maximum, a port security violation has occurred. The switch takes action (the default action is to shutdown the interface).

While those rules define the basics, port security allows other options as well, including letting you configure the specific MAC address(es) allowed to send frames in an interface. For example, in Figure 8-7, switch SW1 connects through interface F0/1 to PC1, so the port security configuration could list PC1's MAC address as the specific allowed MAC address. But predefining MAC addresses for port security is optional: You can predefine all MAC addresses, none, or a subset of the MAC addresses.

You might like the idea of predefining the MAC addresses for port security, but finding the MAC address of each device can be a bother. Port security provides an easy way to discover the MAC addresses used off each port using a feature called *sticky secure MAC addresses*. With this feature, port security learns the MAC addresses off each port and stores those in the port security configuration (in the running-config file). This feature helps reduce the big effort of finding out the MAC address of each device.

As you can see, port security has a lot of detailed options. The next few sections walk you through these options to pull the ideas together.

Configuring Port Security

Port security configuration involves several steps. First, you need to disable the negotiation of a feature that is not discussed until Chapter 9: whether the port is an access or trunk port. For now, accept that port security requires a port to be configured to either be an access port or a trunking port. The rest of the commands enable port security, set the maximum allowed MAC addresses per port, and configure the actual MAC addresses, as detailed in this list:

Key Topic

Step 1. Make the switch interface either a static access or trunk interface, using the **switchport mode access** or the **switchport mode trunk** interface subcommands, respectively.

Step 2. Enable port security using the **switchport port-security** interface subcommand.

Step 3. (Optional) Override the default maximum number of allowed MAC addresses associated with the interface (1) by using the **switchport port-security maximum** *number* interface subcommand.

Step 4. (Optional) Override the default action to take upon a security violation (shutdown) using the **switchport port-security violation** {**protect** | **restrict** | **shutdown**} interface subcommand.

Step 5. (Optional) Predefine any allowed source MAC address(es) for this interface, using the **switchport port-security mac-address** *mac-address* command. Use the command multiple times to define more than one MAC address.

Step 6. (Optional) Tell the switch to "sticky learn" dynamically learned MAC addresses with the **switchport port-security mac-address sticky** interface subcommand.

Figure 8-8 and Example 8-13 show four examples of port security, each with different details just to show the different options.

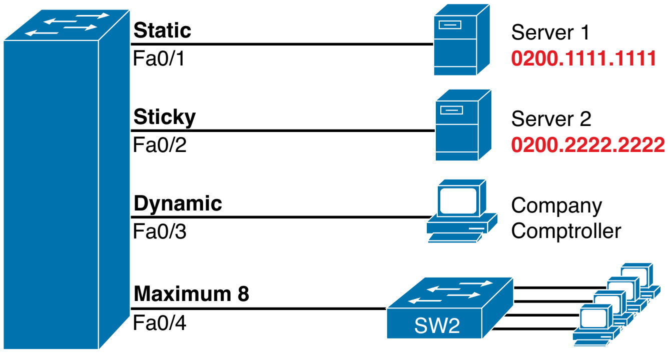


Figure 8-8 *Port Security Configuration Example*

Example 8-13 *Variations on Port Security Configuration*

```
SW1# show running-config
(Lines omitted for brevity)

interface FastEthernet0/1
 switchport mode access
 switchport port-security
 switchport port-security mac-address 0200.1111.1111
!
```

```
interface FastEthernet0/2
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
!
interface FastEthernet0/3
 switchport mode access
 switchport port-security
!
interface FastEthernet0/4
 switchport mode access
 switchport port-security
 switchport port-security maximum 8
```

First, scan the configuration for all four interfaces in Example 8-13, focusing on the first two interface subcommands. Note that all four interfaces in the example use the same first two interface subcommands, matching the first two configuration steps noted before Figure 8-8. The **switchport port-security** command enables port security, with all defaults, with the **switchport mode access** command meeting the requirement to configure the port as either an access or trunk port.

Next, scan all four interfaces again, and note that the configuration differs on each interface after those first two interface subcommands. Each interface simply shows a different example for perspective.

The first interface, FastEthernet 0/1, adds one optional port security subcommand: **switchport port-security mac-address 0200.1111.1111**, which defines a specific source MAC address. With the default maximum source address setting of 1, only frames with source MAC 0200.1111.1111 will be allowed in this port. When a frame with a source other than 0200.1111.1111 enters F0/1, the switch will take the default violation action and disable the interface.

As a second example, FastEthernet 0/2 uses the same logic as FastEthernet 0/1, except that it uses the sticky learning feature instead of predefining a MAC address with the **switchport port-security mac-address sticky** command. The end of upcoming Example 8-14 shows the running config file that lists the sticky-learned MAC address in this case.

NOTE Port security does not save the configuration of the sticky addresses, so use the **copy running-config startup-config** command if desired.

The other two interfaces do not predefine MAC addresses, nor do they sticky-learn the MAC addresses. The only difference between these two interfaces' port security configuration is that FastEthernet 0/4 supports eight MAC addresses, because it connects to another switch and should receive frames with multiple source MAC addresses. Interface F0/3 uses the default maximum of one MAC address.

Verifying Port Security

Example 8-14 lists the output of two examples of the **show port-security interface** command. This command lists the configuration settings for port security on an interface, plus it lists several important facts about the current operation of port security, including information about any security violations. The two commands in the example show interfaces F0/1 and F0/2, based on Example 8-13's configuration.

Example 8-14 *Using Port Security to Define Correct MAC Addresses of Particular Interfaces*

```
SW1# show port-security interface fastEthernet 0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0013.197b.5004:1
Security Violation Count   : 1

SW1# show port-security interface fastEthernet 0/2
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : 0200.2222.2222:1
Security Violation Count   : 0

SW1# show running-config
(Lines omitted for brevity)
interface FastEthernet0/2
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0200.2222.2222
```

The first two commands in Example 8-14 confirm that a security violation has occurred on FastEthernet 0/1, but no violations have occurred on FastEthernet 0/2. The **show port-security interface fastethernet 0/1** command shows that the interface is in a *secure-shutdown* state, which means that the interface has been disabled because of port security. In this case, another device connected to port F0/1, sending a frame with a source MAC address other than 0200.1111.1111, is causing a violation. However, port Fa0/2, which used sticky learning, simply learned the MAC address used by Server 2.

8

The bottom of Example 8-14, as compared to the configuration in Example 8-13, shows the changes in the running-config because of sticky learning, with the **switchport port-security mac-address sticky 0200.2222.2222** interface subcommand.

Port Security Actions

Finally, the switch can be configured to use one of three actions when a violation occurs. All three options cause the switch to discard the offending frame, but some of the options make the switch take additional actions. The actions include the sending of syslog messages to the console, sending SNMP trap messages to the network management station, and disabling the interface. Table 8-3 lists the options of the **switchport port-security violation {protect | restrict | shutdown}** command and their meanings.

Table 8-3 Actions When Port Security Violation Occurs

Option on the switchport port-security violation Command	Protect	Restrict	Shutdown*
Discards offending traffic	Yes	Yes	Yes
Sends log and SNMP messages	No	Yes	Yes
Disables the interface, discarding all traffic	No	No	Yes

***shutdown** is the default setting.

Note that the shutdown option does not actually add the **shutdown** subcommand to the interface configuration. Instead, IOS puts the interface in an *error disabled* (err-disabled) state, which makes the switch stop all inbound and outbound frames. To recover from this state, someone must manually disable the interface with the **shutdown** interface command and then enable the interface with the **no shutdown** command.

Securing Unused Switch Interfaces

The default settings on Cisco switches work great if you want to buy a switch, unbox it, plug it in, and have it immediately work without any other effort. Those same defaults have an unfortunate side effect for worse security. With all default configuration, unused interfaces might be used by an attacker to gain access to the LAN. So, Cisco makes some general recommendations to override the default interface settings to make the unused ports more secure, as follows:

- Administratively disable the interface using the **shutdown** interface subcommand.
- Prevent VLAN trunking by making the port a nontrunking interface using the **switchport mode access** interface subcommand.
- Assign the port to an unused VLAN using the **switchport access vlan** *number* interface subcommand.
- Set the native VLAN to not be VLAN 1, but to instead be an unused VLAN, using the **switchport trunk native vlan** *vlan-id* interface subcommand. (The native VLAN is discussed in Chapter 9.)

Frankly, if you just shutdown the interface, the security exposure goes away, but the other tasks prevent any immediate problems if someone else comes around and enables the interface by configuring a **no shutdown** command.

Review Activities

Chapter Summary

- The first step in securing a switch is to secure access to the CLI. Securing the CLI includes protecting access to enable mode, because from enable mode an attacker could reload the switch or change the configuration.
- Cisco switches protect enable mode for any user with the *enable password*. The user, in user mode, types the **enable** EXEC command and is prompted for this enable password. If the user types the correct password, IOS moves the user to enable mode.
- The console and vty password configuration uses the same two subcommands in console and vty line configuration modes, respectively. The **login** command tells IOS to use simple password security, and the **password** *password_value* command defines the password. IOS protects enable mode using the enable secret password, configured using the global command **enable secret** *password_value*.
- The migration from using the password-only login method to using locally configured usernames and passwords requires only some small configuration changes. The switch needs one or more **username** *name* **password** *password* global configuration commands to define the usernames and passwords.
- Cisco switches and routers support an alternative way to keep track of valid usernames and passwords by using an external AAA server. 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.
- To support SSH, Cisco switches require the base configuration used to support Telnet login with usernames, plus additional configuration. First, the switch already runs an SSH server by default, accepting incoming SSH connections from both SSH version 1 and version 2 clients. In addition, the switch needs a cryptography key, used to encrypt the data.
- To prevent password vulnerability in a printed version of the configuration file, or in a backup copy of the configuration file stored on a server, you can encrypt some passwords using the **service password-encryption** global configuration command.
- The **banner** global configuration command can be used to configure all three types of banners:
  - **The message of the day (MOTD):** Shown before the login prompt. For temporary messages that might change from time to time, such as “Router1 down for maintenance at midnight.”
  - **The login banner:** Shown before the login prompt but after the MOTD banner. For permanent messages, such as “Unauthorized Access Prohibited.”
  - **The Exec banner:** Shown after the login prompt. Used to supply information that should be hidden from unauthorized users.
- A switch configures its IPv4 address and mask on this special NIC-like *VLAN interface*. The following steps list the commands used to configure IPv4 on a switch, assuming the IP address is configured to be in VLAN 1:

  Step 1. Enter VLAN 1 configuration mode using the **interface vlan 1** global configuration command.

  Step 2. Assign an IP address and mask using the **ip address** *ip-address mask* interface subcommand.

  Step 3. If not already enabled, enable the VLAN 1 interface using the **no shutdown** interface subcommand.

Step 4. Add the **ip default-gateway** *ip-address* global command to configure the default gateway.

Step 5. (Optional) Add the **ip name-server** *ip-address1 ip-address2…* global command to configure the switch to use DNS to resolve names into their matching IP addresses.

- To administratively enable an interface on a switch, use the **no shutdown** interface subcommand. To disable an interface, use the **shutdown** interface subcommand.
- If the network engineer knows what devices should be cabled and connected to particular interfaces on a switch, the engineer can use *port security* to restrict that interface so that only the expected devices can use it. This reduces exposure to attacks in which the attacker connects a laptop to some unused switch port. When that inappropriate device attempts to send frames to the switch interface, the switch can take different actions, ranging from simply issuing informational messages to effectively shutting down the interface.

Review Questions

Answer these review questions. You can find the answers at the bottom of the last page of the chapter. For thorough explanations, see DVD Appendix C, "Answers to Review Questions."

1. Imagine that you have configured the **enable secret** command, followed by the **enable password** command, from the console. You log out of the switch and log back in at the console. Which command defines the password that you had to enter to access privileged mode?

A. **enable password**

B. **enable secret**

C. Neither

D. The **password** command, if it's configured

2. An engineer had formerly configured a Cisco 2960 switch to allow Telnet access so that the switch expected a password of **mypassword** from the Telnet user. The engineer then changed the configuration to support Secure Shell. Which of the following commands could have been part of the new configuration? (Choose two answers.)

A. A **username** *name* **password** *password* vty mode subcommand

B. A **username** *name* **password** *password* global configuration command

C. A **login local** vty mode subcommand

D. A **transport input ssh** global configuration command

3. The following command was copied and pasted into configuration mode when a user was telnetted into a Cisco switch:

```
banner login this is the login banner
```

Which of the following is true about what occurs the next time a user logs in from the console?

A. No banner text is displayed.

B. The banner text "his is" is displayed.

C. The banner text "this is the login banner" is displayed.

D. The banner text "Login banner configured, no text defined" is displayed.

4. Which of the following is required when configuring port security with sticky learning?

A. Setting the maximum number of allowed MAC addresses on the interface with the **switchport port-security maximum** interface subcommand

B. Enabling port security with the **switchport port-security** interface subcommand

C. Defining the specific allowed MAC addresses using the **switchport port-security mac-address** interface subcommand

D. All the other answers list required commands

5. An engineer's desktop PC connects to a switch at the main site. A router at the main site connects to each branch office through a serial link, with one small router and switch at each branch. Which of the following commands must be configured on the branch office switches, in the listed configuration mode, to allow the engineer to telnet to the branch office switches? (Choose three answers.)

A. The **ip address** command in VLAN configuration mode

B. The **ip address** command in global configuration mode

C. The **ip default-gateway** command in VLAN configuration mode

D. The **ip default-gateway** command in global configuration mode

E. The **password** command in console line configuration mode

F. The **password** command in vty line configuration mode

6. Which of the following describes a way to disable IEEE standard autonegotiation on a 10/100 port on a Cisco switch?

A. Configure the **negotiate disable** interface subcommand

B. Configure the **no negotiate** interface subcommand

C. Configure the **speed 100** interface subcommand

D. Configure the **duplex half** interface subcommand

E. Configure the **duplex full** interface subcommand

F. Configure the **speed 100** and **duplex full** interface subcommands

7. In which of the following modes of the CLI could you configure the duplex setting for interface Fast Ethernet 0/5?

A. User mode

B. Enable mode

C. Global configuration mode

D. VLAN mode

E. Interface configuration mode

Review All the Key Topics

Review the most important topics from this chapter, noted with the Key Topic icon. Table 8-4 lists these key topics and shows where each is discussed.

NOTE There is no need to memorize any configuration step list referenced as a key topic; these lists are just study aids.

Key Topic

Table 8-4 Key Topics for Chapter 8

Key Topic Element	Description	Page Number
Example 8-1	Example showing basic password configuration	176
Figure 8-2	Configuration steps to use local usernames	179
List	Configuration steps for SSH support on a switch	180
List	Key points about **enable secret** and **enable password**	184
Table 8-2	List of commands related to the command history buffer	187
Figure 8-5	Conceptual diagram of a switch VLAN interface	189
List	Configuration checklist for a switch's IP address and default gateway configuration	190
List	Configuration checklist for a switch to learn IP settings as a DHCP client	190
List	Key features of all variations of port security	194
List	Port security configuration checklist	195
Table 8-3	Port security actions and the results of each action	198
List	Suggested security actions for unused switch ports	198

Complete the Tables and Lists from Memory

Print a copy of DVD Appendix M, "Memory Tables," or at least the section for this chapter, and complete the tables and lists from memory. DVD Appendix N, "Memory Tables Answer Key," includes completed tables and lists for you to check your work.

Definitions of Key Terms

After your first reading of the chapter, try to define these key terms, but do not be concerned about getting them all correct at that time. Chapter 30 directs you in how to use these terms for late-stage preparation for the exam.

Telnet, SSH, local username, VLAN interface, port security

Command References

Tables 8-5 though 8-9 list the configuration commands used in this chapter, by general topic. Table 8-10, at the very end of the chapter, lists the EXEC commands from this chapter.

Table 8-5 Console, Telnet, and SSH Login Commands

Command	Mode/Purpose/Description
line console 0	Changes the context to console configuration mode.
line vty *1st-vty last-vty*	Changes the context to vty configuration mode for the range of vty lines listed in the command.
login	Console and vty configuration mode. Tells IOS to prompt for a password.
password *pass-value*	Console and vty configuration mode. Lists the password required if the **login** command (with no other parameters) is configured.
login local	Console and vty configuration mode. Tells IOS to prompt for a username and password, to be checked against locally configured **username** global configuration commands on this switch or router.
username *name* **secret** *pass-value*	Global command. Defines one of possibly multiple usernames and associated passwords, used for user authentication. Used when the **login local** line configuration command has been used.
crypto key generate rsa	Global command. Creates and stores (in a hidden location in flash memory) the keys required by SSH.
transport input {**telnet** | **ssh** | **all** | **none**}	vty line configuration mode. Defines whether Telnet and/or SSH access is allowed into this switch. Both values can be configured on one command to allow both Telnet and SSH access (the default).
service password-encryption	Global command that (weakly) encrypts passwords defined by the **username password**, **enable password**, and **login** commands.

Table 8-6 Switch IPv4 Configuration

Command	Mode/Purpose/Description
interface vlan *number*	Changes the context to VLAN interface mode. For VLAN 1, allows the configuration of the switch's IP address.
ip address *ip-address subnet-mask*	VLAN interface mode. Statically configures the switch's IP address and mask.
ip address dhcp	VLAN interface mode. Configures the switch as a DHCP client to discover its IP address, mask, and default gateway.

Command	Mode/Purpose/Description
ip default-gateway *address*	Global command. Configures the switch's default gateway IP address. Not required if the switch uses DHCP.
ip name-server *server-ip-1 server-ip-2 …*	Global command. Configures the IP address(es) of DNS servers, so any commands when logged into the switch will use the DNS for name resolution.

Table 8-7 Switch Interface Configuration

Command	Mode/Purpose/Description
interface *type port-number*	Changes context to interface mode. The type is typically FastEthernet or GigabitEthernet. The possible port numbers vary depending on the model of switch—for example, Fa0/1, Fa0/2, and so on.
interface range *type port-range*	Changes the context to interface mode for a range of consecutively numbered interfaces. The subcommands that follow then apply to all interfaces in the range.
shutdown **no shutdown**	Interface mode. Disables or enables the interface, respectively.
speed {**10** | **100** | **1000** | **auto**}	Interface mode. Manually sets the speed to the listed speed or, with the **auto** setting, automatically negotiates the speed.
duplex {**auto** | **full** | **half**}	Interface mode. Manually sets the duplex to half or full, or to autonegotiate the duplex setting.
description *text*	Interface mode. Lists any information text that the engineer wants to track for the interface, such as the expected device on the other end of the cable.

Table 8-8 Port Security

Command	Mode/Purpose/Description
switchport mode {access | trunk | negotiate}	Interface configuration mode command that tells the switch to always be an access port, or always be a trunk port, or to negotiate which to be.
switchport port-security mac-address *mac-address*	Interface configuration mode command that statically adds a specific MAC address as an allowed MAC address on the interface.
switchport port-security mac-address sticky	Interface subcommand that tells the switch to learn MAC addresses on the interface and add them to the configuration for the interface as secure MAC addresses.
switchport port-security maximum *value*	Interface subcommand that sets the maximum number of static secure MAC addresses that can be assigned to a single interface.
switchport port-security violation {**protect** | **restrict** | **shutdown**}	Interface subcommand that tells the switch what to do if an inappropriate MAC address tries to access the network through a secure switch port.

Table 8-9 Other Switch Configuration

Command	Mode/Purpose/Description
hostname *name*	Global command. Sets this switch's host name, which is also used as the first part of the switch's command prompt.
enable secret *pass-value*	Global command. Sets this switch's password that is required for any user to reach enable mode.
history size *length*	Line config mode. Defines the number of commands held in the history buffer, for later recall, for users of those lines.
logging synchronous	Console or vty mode. Tells IOS to send log messages to the user at natural break points between commands, rather than in the middle of a line of output.
[no] logging console	Global command that disables or enables the display of log messages to the console.
exec-timeout *minutes [seconds]*	Console or vty mode. Sets the inactivity timeout, so that after the defined period of no action, IOS closes the current user login session.
switchport access vlan *vlan-number*	Interface subcommand that defines the VLAN in which the interface resides.
banner [motd | exec | login] *delimiter banner-text delimiter*	Global command that defines a banner that is displayed at different times when users log in to the switch or router.

Table 8-10 Chapter 8 EXEC Command Reference

Command	Purpose
show running-config	Lists the currently used configuration.
show running-config | begin line vty	Pipes (sends) the command output to the **begin** command, which only lists output beginning with the first line that contains the text "line vty."
show mac address-table dynamic	Lists the dynamically learned entries in the switch's address (forwarding) table.
show dhcp lease	Lists any information the switch acquires as a DHCP client. This includes IP address, subnet mask, and default gateway information.
show crypto key mypubkey rsa	Lists the public and shared key created for use with SSH using the **crypto key generate rsa** global configuration command.
show ip ssh	Lists status information for the SSH server, including the SSH version.
show ssh	Lists status information for current SSH connections into and out of the local switch.
show interfaces status	Lists one output line per interface, noting the description, operating state, and settings for duplex and speed on each interface.
show interfaces vlan 1	Lists the interface status, the switch's IP address and mask, and much more.

Command	Purpose
show ip default-gateway	Lists the interface status, the switch's IP address and mask, and much more.
show port-security interface *type number*	Lists an interface's port security configuration settings and security operational status.
terminal history size *x*	Changes the length of the history buffer for the current user only, only for the current login to the switch.
show history	Lists the commands in the current history buffer.

Answers to Review Questions:

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

This page intentionally left blank

Chapter 9

Implementing Ethernet Virtual LANs

At their heart, Ethernet switches receive Ethernet frames, make decisions, and then forward (switch) those Ethernet frames. That core logic revolves around MAC addresses, the interface in which the frame arrives, and the interface(s) out which the switch forwards the frame. Several switch features have some impact on an individual switch's decisions about where to forward frames, but of all the topics in this book, virtual LANs (VLAN) easily have the biggest impact on those choices.

This chapter examines the concepts and configuration of VLANs. The first major section of the chapter explains the core concepts. These concepts include how VLANs work on a single switch, how to use VLAN trunking to create VLANs that span across multiple switches, and how to forward traffic between VLANs using a router. The second major section shows how to configure VLANs and VLAN trunks: how to statically assign interfaces to a VLAN.

This chapter covers the following exam topics:

Operation of IP Data Networks

Predict the data flow between two hosts across a network.

LAN Switching Technologies

Identify basic switching concepts and the operation of Cisco switches.

Broadcast Domains

CAM Table

Describe how VLANs create logically separate networks and the need for routing between them.

Explain network segmentation and basic traffic management concepts

Configure and verify VLANs

Configure and verify trunking on Cisco switches

DTP

Troubleshooting

Troubleshoot and Resolve VLAN problems

Identify that VLANs are configured

Port membership correct

IP address configured

Troubleshoot and Resolve trunking problems on Cisco switches

Correct trunk states

Correct encapsulation configured

Correct vlans allowed

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 9-1.

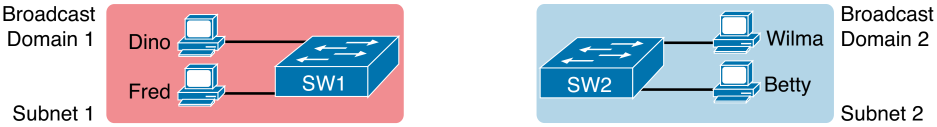


Figure 9-1 *Creating Two Broadcast Domains with Two Physical Switches and No VLANs*

With support for VLANs, a single switch can accomplish the same goals of the design in Figure 9-1—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 9-2, 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

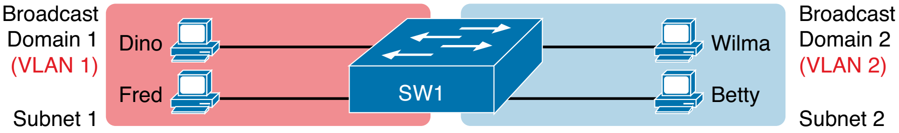


Figure 9-2 *Creating Two Broadcast Domains Using One Switch and VLANs*

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 9-3 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.

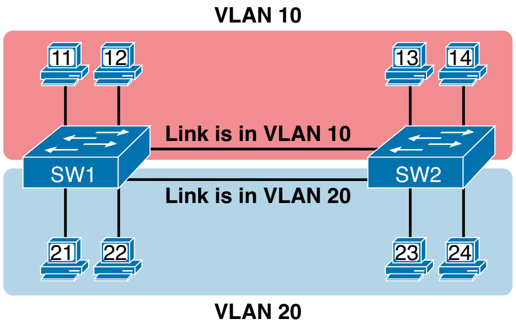


Figure 9-3 *Multiswitch VLAN Without VLAN Trunking*

The design in Figure 9-3 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 9-3 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 9-4 shows the idea, with a single physical link between the two switches.

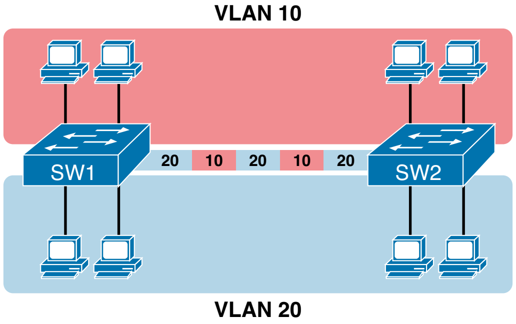


Figure 9-4 *Multiswitch VLAN with Trunking*

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 9-5 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.

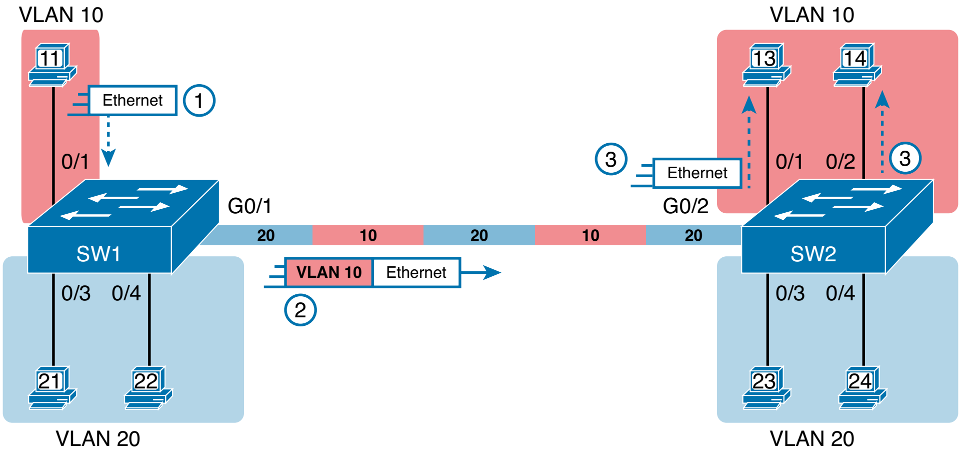


Figure 9-5 *VLAN Trunking Between Two Switches*

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 noticing 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 the 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 9-6. 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, while 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].)

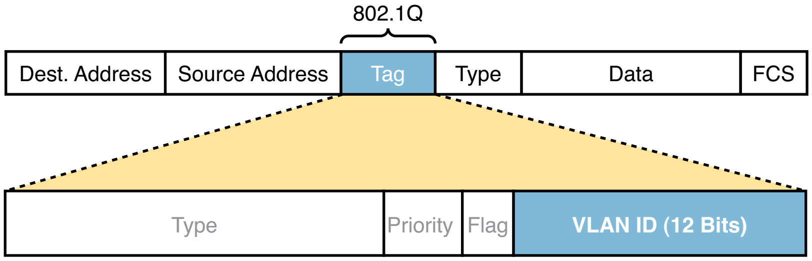


Figure 9-6 *802.1Q Trunking*

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 1005 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 come categories of LAN switches. All the Ethernet switch functions described in this book so far use the details and logic defined by OSI Layer 2 protocols. For example, Chapter 6, "Building Ethernet LANs with Switches," 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 so far in this book, some LAN switches have even more functions. LAN switches that forward data based on Layer 2 logic, as discussed so far in this book, 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 9-7, 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).

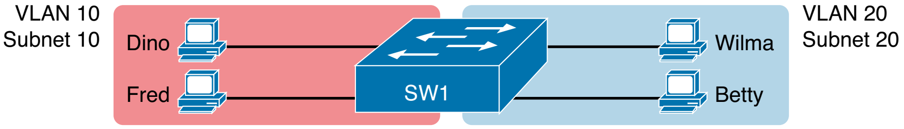


Figure 9-7 *Routing Between VLANs on Two Physically Separated Switches*

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.

Layer 2 switches will not forward data between two VLANs. 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 PDU (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 9-8 shows a router that can route packets between subnets 10 and 20. The figure shows the same Layer 2 switch as shown in Figure 9-7, with the same PCs and with 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.

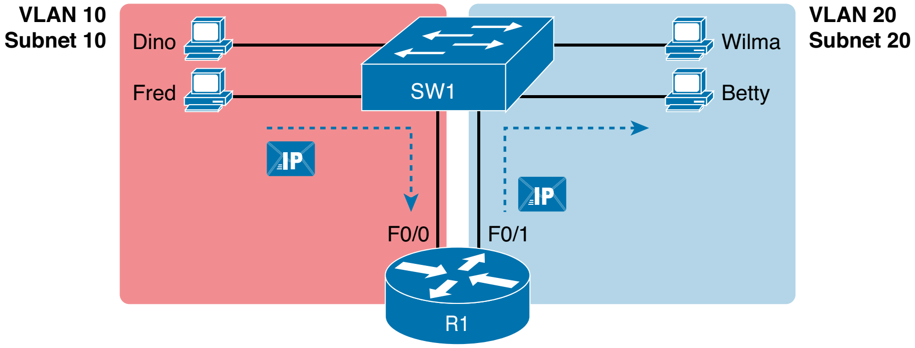


Figure 9-8 *Routing Between Two VLANs on Two Physical Interfaces*

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 9-8 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 9-9 shows the same design idea as Figure 9-8, 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

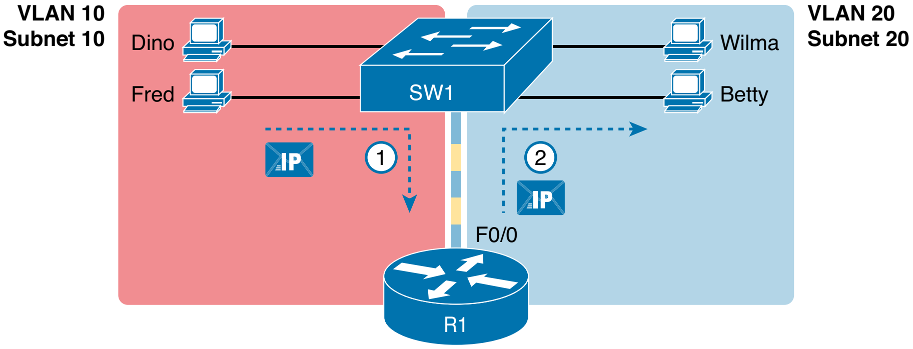


Figure 9-9 *Routing Between Two VLANs Using a Trunk on the Router*

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 9-8 and 9-9 as "routing packets between VLANs." You can use that phrase, and people know what you mean. However, for exam preparation purposes, 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 different 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 9-9, 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 9-8, with a separate Layer 2 switch and Layer 3 router, and then image all those features happening inside one device, you have the general idea of what a Layer 3 switch does. Figure 9-10 shows that exact concept, repeating many details of Figure 9-8, but with an overlay that shows the one Layer 3 switch doing the Layer 2 switch functions and the separate Layer 3 routing function.

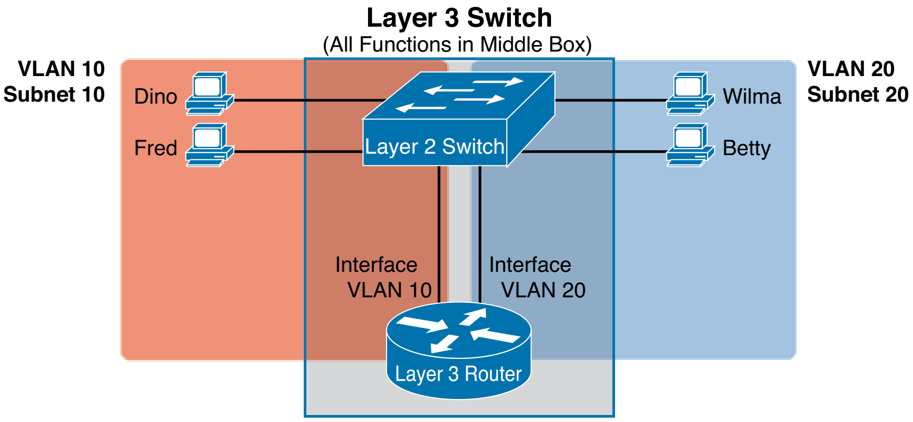


Figure 9-10 *Multilayer Switch: Layer 2 Switching with Layer 3 Routing in One Device*

This chapter introduces the core concepts of routing IP packets between VLANs (or more accurately, between the subnets on the VLANs). Chapter 16, “Configuring IPv4 Addresses and Routes,” 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 every enterprise network does—you need to add some configuration.

This chapter separates the VLAN configuration details into two major sections. The first looks at how to configure access interfaces, which 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. Additionally, 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”:

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

A. From configuration mode, use the **vlan** *vlan-id* global configuration command to create the VLAN and to move the user into VLAN configuration mode.

B. (Optional) Use the **name** *name* VLAN subcommand to list a name for the VLAN. If not configured, the VLAN name is VLANZZZZ, where *ZZZZ* is the 4-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** command to move into interface configuration mode for each desired interface.

B. Use the **switchport access vlan** *id-number* interface subcommand to specify the VLAN number associated with that interface.

C. (Optional) To disable trunking on that same interface, so that the interface does not negotiate to become a trunk, use the **switchport mode access** interface subcommand.

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.

VLAN Configuration Example 1: Full VLAN Configuration

Example 9-1 shows the configuration process of adding a new VLAN and assigning access interfaces to that VLAN. Figure 9-11 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.

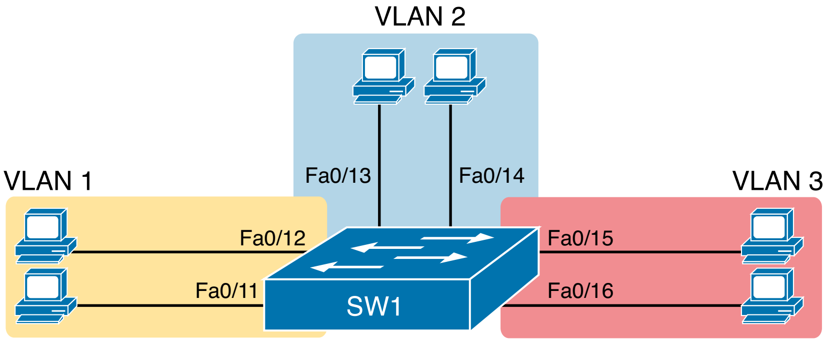


Figure 9-11 *Network with One Switch and Three VLANs*

Example 9-1 *Configuring VLANs and Assigning VLANs to Interfaces*

```
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)# 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  -      -      -        -    -        0      0
```

```
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 example surrounding Figure 9-11 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 9-1, 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.

VLAN Configuration Example 2: Shorter VLAN Configuration

Example 9-1 shows several of the optional configuration commands, with a side effect of being a bit longer than is required. Example 9-2 shows a much briefer alternative configuration, picking up the story where Example 9-1 ended and showing the addition of VLAN 3 (as seen in Figure 9-11). Note that SW1 does not know about VLAN 3 at the beginning of this example.

Example 9-2 *Shorter VLAN Configuration Example (VLAN 3)*

```
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 9-2 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 (VTP)

Before showing more configuration examples, you also need to know something about an older 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.

This book does not discuss VTP as an end to itself. However, VTP has some small impact on how every Cisco Catalyst switch works, even if you do not try and use VTP. This brief section introduces enough details of VTP so that you can see these small differences in VTP that cannot be avoided.

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.

This book attempts to ignore VTP as much as is possible. To that end, all examples in this book use switches that have either been set 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.
- The **show running-config** command does not list any **vlan** commands.

If possible, switch to VTP transparent mode and ignore VTP for your switch configuration practice for the CCENT and CCNA exam.

NOTE If you experiment with VTP settings on a real lab switch, be very careful. If that switch connects to other switches, which in turn connect to switches used in the production LAN, it is possible to cause problems by overwriting the VLAN configuration in other switches. Be careful and never experiment with VTP settings on a switch unless it, and the other switches connected to it, have absolutely no physical links connected to the production LAN.

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 9-1.

Key Topic

Table 9-1 Trunking Administrative Mode Options with the **switchport mode** Command

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

For example, consider the two switches shown in Figure 9-12. This figure shows an expansion of the network of Figure 9-11, with a trunk to a new switch (SW2) and with parts of VLANs 1 and 3 on portsattached 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.

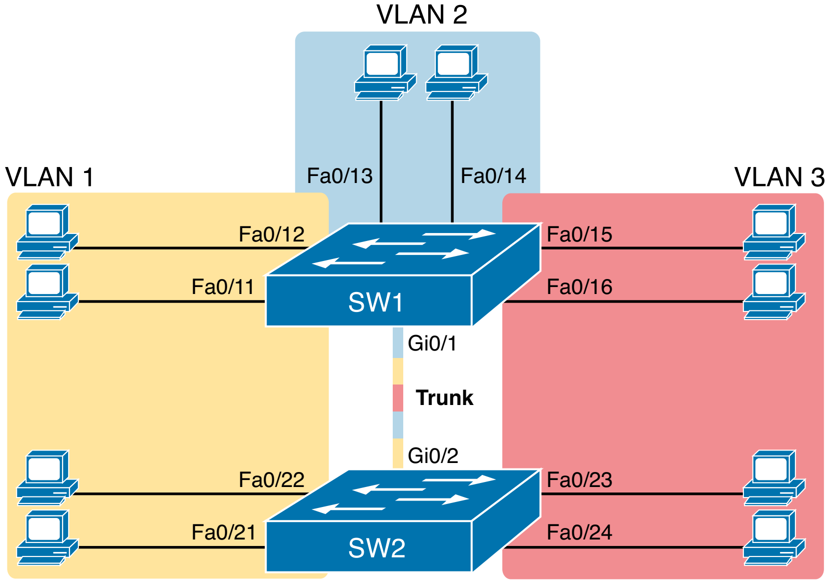


Figure 9-12 *Network with Two Switches and Three VLANs*

Example 9-3 begins by showing the two switches in Figure 9-12 with the default configuration so that the two switches do not trunk.

Example 9-3 *Initial (Default) State: Not Trunking Between SW1 and SW2*

```
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 9-3. 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 list 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 9-4, 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 9-4 *SW1 Changes from Dynamic Auto to Dynamic Desirable*

```
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#
01:43:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed
state to down
01:43:49: %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        Native 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  -      -      -        -    -        0      0

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

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

To verify that trunking is working now, the middle of Example 9-4 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 interfaces trunk** 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 9-2 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

Table 9-2 Expected Trunking Operational Mode Based on the Configured Administrative Modes

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.

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.

Controlling Which VLANs Can Be Supported on a Trunk

The *allowed VLAN list* feature provides a mechanism for engineers to administratively disable a VLAN from a trunk. By default, switches include all possible VLANs (1–4094) in each trunk's allowed VLAN list. However, the engineer can then limit the VLANs allowed on the trunk by using the following interface subcommand:

```
switchport trunk allowed vlan {add | all | except | remove} vlan-list
```

This command provides a way to easily add and remove VLANs from the list. For example, the **add** option permits the switch to add VLANs to the existing allowed VLAN list, and the **remove** option permits the switch to remove VLANs from the existing list. The **all** option means all VLANs, so you can use it to reset the switch to its original default setting (permitting VLANs 1–4094 on the trunk). The **except** option is rather tricky: It adds all VLANs to the list that are not part of the command. For example, the **switchport trunk allowed vlan except 100-200** interface subcommand adds VLANs 1 through 99 and 201 through 4094 to the existing allowed VLAN list on that trunk.

In addition to the allowed VLAN list, a switch has other reasons to prevent a particular VLAN's traffic from crossing a trunk. All five reasons are summarized in the following list:

Key Topic

- A VLAN has been removed from the trunk's *allowed VLAN* list.
- A VLAN does not exist in the switch's configuration (as seen with the **show vlan** command).
- A VLAN does exist, but has been administratively disabled (**shutdown**).
- A VLAN has been automatically pruned by VTP.
- A VLAN's STP instance has placed the trunk interface into a blocking state.

NOTE The last two reasons in the list are outside the scope of this book, but are mentioned here for completeness.

While this section has already discussed the first reason—the allowed VLAN list—next consider the next two reasons in the list. If a switch does not know that a VLAN exists—for example, if the switch does not have a **vlan** *vlan-id* command configured, as confirmed by the output of the **show vlan** command—the switch will not forward frames in that VLAN over any interface. Additionally, a VLAN can exist in a switch's configuration, but also be administratively shut down either by using the **shutdown vlan** *vlan-id* global configuration command, or using the **shutdown** command in VLAN configuration mode. When disabled, a switch will no longer forward frames in that VLAN, even over trunks. So, switches do not forward frames in nonexistent VLANs or a shutdown VLAN over any of the switch's trunks.

This book has a motive for listing the reasons for limiting VLANs on a trunk: The **show interfaces trunk** command lists VLAN ID ranges as well, based on these same reasons. This command includes a progression of three lists of the VLANs supported over a trunk. These three lists are as follows:

- VLANs allowed on the trunk, 1–4094 by default
- VLANs from the first group that are also configured and active (not shut down)
- VLANs from the second group that are not VTP pruned and not STP blocked

To get an idea of these three lists inside the output of the **show interfaces trunk** command, Example 9-5 shows how VLANs might be disallowed on a trunk for various reasons. The command output is taken from SW1 in Figure 9-12, after the completion of the configuration as shown in all the earlier examples in this chapter. In other words, VLANs 1 through 3 exist in SW1's configuration, and are not shut down. Trunking is operational between SW1 and SW2. Then, during the example, the following items are configured on SW1:

Step 1. VLAN 4 is configured.

Step 2. VLAN 2 is shut down.

Step 3. VLAN 3 is removed from the trunk's allowed VLAN list.

Example 9-5 *Allowed VLAN List and the List of Active VLANs*

```
! The three lists of VLANs in the next command list allowed VLANs (1-4094),
! Allowed and active VLANs (1-3), and allowed/active/not pruned/STP forwarding
! VLANs (1-3)
SW1# show interfaces trunk

Port        Mode         Encapsulation  Status        Native 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

! Next, the switch is configured with new VLAN 4; VLAN 2 is shutdown;
! and VLAN 3 is removed from the allowed VLAN list on the trunk.
SW1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SW1(config)# vlan 4
SW1(config-vlan)# vlan 2
SW1(config-vlan)# shutdown
SW1(config-vlan)# interface gi0/1
SW1(config-if)# switchport trunk allowed vlan remove 3
SW1(config-if)# ^Z

! The three lists of VLANs in the next command list allowed VLANs (1-2, 4-4094),
! allowed and active VLANs (1,4), and allowed/active/not pruned/STP forwarding
! VLANs (1,4)
SW1# show interfaces trunk

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

! VLAN 3 is omitted next, because it was removed from the allowed VLAN list.
Port        Vlans allowed on trunk
Gi0/1       1-2,4-4094

! VLAN 2 is omitted below because it is shutdown. VLANs 5-4094 are omitted below
! because SW1 does not have them configured.
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
```

Review Activities

Chapter Summary

- A LAN includes all devices in the same broadcast domain.
- Without VLANs, a switch considers all its interfaces to be in the same broadcast domain.
- 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 (VLANs).
- The following list summarizes the most common reasons for choosing to create smaller broadcast domains (VLANs):
  - 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
- 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.
- When using VLANs in networks that have multiple interconnected switches, the switches must 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.
- Cisco has supported two different trunking protocols over the years: Inter-Switch Link (ISL) and IEEE 802.1Q.
- 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.
- If you create a campus LAN that contains many VLANs, you typically still need all devices to be capable of sending data to all other devices. Layer 2 switches will not forward data between two VLANs. Traditionally, a router is used to route packets between VLANs.
- 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 to do both Layer 2 switching and Layer 3 routing.
- The configuration steps for access interfaces are as follows:

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

  1. From configuration mode, use the **vlan** *vlan-id* global configuration command to create the VLAN and to move the user into VLAN configuration mode.

2. (Optional) Use the **name** *name* VLAN subcommand to list a name for the VLAN. If not configured, the VLAN name is VLANZZZZ, where *ZZZZ* is the 4-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:

1. Use the **interface** command to move into interface configuration mode for each desired interface.
2. Use the **switchport access vlan** *id-number* interface subcommand to specify the VLAN number associated with that interface.
3. (Optional) To disable trunking on that same interface, ensuring that the interface is an access interface, use the **switchport mode access** interface subcommand.

- VLAN Trunking Protocol (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.

Review Questions

Answer these review questions. You can find the answers at the bottom of the last page of the chapter. For thorough explanations, see DVD Appendix C, "Answers to Review Questions."

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 can't 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. For an 802.1Q trunk between two Ethernet switches, which answer most accurately defines which frames do not include an 802.1Q header?

A. Frames in the native VLAN (only one)

B. Frames in extended VLANs

C. Frames in VLAN 1 (not configurable)

D. Frames in all native VLANs (multiple allowed)

5. 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. Trunking turned **on**

B. **dynamic auto**

C. **dynamic desirable**

D. **access**

E. None of the other answers are correct.

6. 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.

7. 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**

Review All the Key Topics

Review the most important topics from this chapter, noted with the Key Topic icon. Table 9-3 lists these key topics and where each is discussed.

Key Topic

Table 9-3 Key Topics for Chapter 9

Key Topic Element	Description	Page Number
Figure 9-2	Basic VLAN concept	209
List	Reasons for using VLANs	210
Figure 9-5	Diagram of VLAN trunking	212
Figure 9-6	802.1Q header	212
Figure 9-9	Routing between VLANs with router-on-a-stick	215
Figure 9-10	Routing between VLANs with Layer 3 switch	216
List	Configuration checklist for configuring VLANs and assigning to interfaces	216
Table 9-1	Options of the **switchport mode** command	222
Table 9-2	Expected trunking results based on the configuration of the **switchport mode** command	225
List	Reasons why a trunk does not pass traffic for a VLAN	226

Complete the Tables and Lists from Memory

Print a copy of Appendix M, "Memory Tables," (found on the DVD) or at least the section for this chapter, and complete the tables and lists from memory. Appendix N, "Memory Tables Answer Key," also on the DVD, includes completed tables and lists to check your work.

Definitions of Key Terms

After your first reading of the chapter, try to define these key terms, but do not be concerned about getting them all correct at that time. Chapter 30 directs you in how to use these terms for late-stage preparation for the exam.

802.1Q, trunk, trunking administrative mode, trunking operational mode, VLAN, VTP, VTP transparent mode, Layer 3 switch, access interface, trunk interface

Command Reference to Check Your Memory

While you should not necessarily memorize the information in the tables in this section, this section does include a reference for the configuration and EXEC commands covered in this chapter. Practically speaking, you should memorize the commands as a side effect of reading the chapter and doing all the activities in this exam preparation section. To check and see how well you have memorized the commands as a side effect of your other studies, cover the left side of the table with a piece of paper, read the descriptions in the right side, and see whether you remember the command.

Table 9-4 Chapter 9 Configuration Command Reference

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 trunk allowed vlan {**add** | **all** | **except** | **remove**} *vlan-list*	Interface subcommand that defines the list of allowed VLANs
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

Table 9-5 Chapter 9 EXEC 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

Answers to Review Questions::

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

This page intentionally left blank

Chapter 10

Troubleshooting Ethernet LANs

This chapter focuses on the processes of verification and troubleshooting. *Verification* refers to the process of confirming whether a network is working as designed. *Troubleshooting* refers to the follow-on process that occurs when the network is not working as designed, by trying to determine the real reason why the network is not working correctly, so that it can be fixed.

Over the years, the CCENT and CCNA exams have been asking more and more verification and troubleshooting questions. Each of these questions requires you to apply networking knowledge to unique problems, rather than just being ready to answer questions about lists of facts that you've memorized.

To help you prepare to answer troubleshooting questions, this book, as well as the ICND2 book, devotes different book elements (both full chapters and sections of chapters) to troubleshooting. These book elements do not just list the configuration, and they do not just list example output from different **show** commands. Instead, these elements discuss how to use different commands to verify what should be happening, and if not, how to find the root cause of the problem.

This chapter discusses a wide number of topics, many of which have already been discussed in Chapters 6, 7, 8, and 9. First, this chapter begins with some perspectives on troubleshooting networking problems, because it is the first book element that focuses on troubleshooting. At that point, this chapter looks at four key technical topics that matter to verifying and troubleshooting Ethernet LANs, as follows:

- Analyzing LAN Topology Using CDP
- Analyzing Switch Interface Status
- Predicting Where Switches Will Forward Frames
- Analyzing VLANs and VLAN Trunks

This chapter covers the following exam topics:

LAN Switching Technologies

- Configure and verify trunking on Cisco switches
  - DTP
  - Autonegotiation

Network Device Security

- Configure and verify Switch Port Security features such as
  - Static / dynamic
  - Violation modes
    - Err disable
    - Shutdown
    - Protect restrict

Troubleshooting

- Troubleshoot and Resolve VLAN problems
  - identify that VLANs are configured
  - port membership correct
  - IP address configured
- Troubleshoot and Resolve trunking problems on Cisco switches
  - correct trunk states
  - correct encapsulation configured
  - correct VLANs allowed
- Troubleshoot and Resolve Layer 1 problems
  - Framing
  - CRC
  - Runts
  - Giants
  - Dropped packets
  - Late collision
  - Input / Output errors

Foundation Topics

Perspectives on Network Verification and Troubleshooting

NOTE The information in this section is a means to help you learn troubleshooting skills. However, the specific processes and comments in this section, up to the next major heading ("Analyzing LAN Topology Using Cisco Discovery Protocol"), do not cover any specific exam objective for any of the CCENT or CCNA exams.

You need several skills to be ready to answer the more challenging questions on today's CCENT and CCNA exams. However, the required skills differ when comparing the different types of questions. This section starts with some perspectives on the various question types, followed by some general comments on troubleshooting.

First, as a reminder, the Introduction to this book briefly describes a couple of different types of exam questions mentioned in this chapter: Sim, Simlet, and multiple choice (MC).

Sim and Simlet questions use a simulator, where you can use the CLI of simulated routers and switches. Sim questions require you to find a configuration problem and solve the problem by fixing or completing the configuration. Simlet questions require you to verify the current operation of the network and then answer MC questions about the current operation. MC questions simply ask a question, with multiple answers (choices) for the correct answer.

NOTE Refer to www.cisco.com/web/learning/wwtraining/certprog/training/cert_exam_tutorial.html for a tutorial about the various types of CCENT and CCNA exam questions.

Preparing to Use an Organized Troubleshooting Process

On exam day, you have one goal: answer enough questions correctly to pass the exam. However, before the exam, you should use a thorough and organized thought process. You can learn a lot by thinking through the troubleshooting process as you prepare for the exam so that you can be better prepared to attack problems quickly on exam day.

To that end, this book includes many suggested troubleshooting processes. The troubleshooting processes are not ends unto themselves, so you do not need to memorize them for the exams. They are learning tools, with the ultimate goal being to help you correctly and quickly find the answers to the more challenging questions on the exams.

This section gives an overview of a general troubleshooting process. As you progress through this book, the process will be mentioned occasionally as it relates to other technology areas, such as IP routing. The three major steps in this book's organized troubleshooting process are as follows:

Step 1. **Analyzing/predicting normal operation:** Predict the details of what should happen if the network is working correctly, based on documentation, configuration, and **show** and **debug** command output.

Step 2. **Problem isolation:** Determine how far along the expected path the frame/packet goes before it cannot be forwarded any farther, again based on documentation, configuration, and **show** and **debug** command output.

Step 3. **Root cause analysis:** Identify the underlying causes of the problems identified in the preceding step—specifically, the causes that have a specific action with which the problem can be fixed.

Following this process requires a wide variety of learned skills. You need to remember the theory of how networks should work, as well as how to interpret the **show** command output that confirms how the devices are currently behaving. This process requires the use of testing tools, such as **ping** and **traceroute**, to isolate the problem. Finally, this approach requires the ability to think broadly about everything that could affect a single component.

For example, consider the following problem based on the network in Figure 10-1. PC1 and PC2 supposedly sit in the same VLAN (10). At one time, the **ping 10.1.1.2** command on PC1 worked; now it does not.

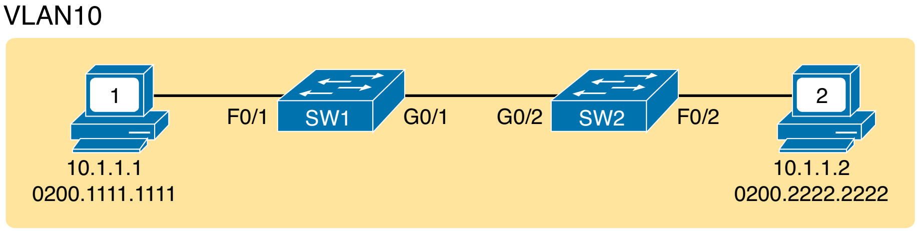


Figure 10-1 *Example Network with a ping Problem*

So, how do you attack this problem? If you doubt whether the figure is even correct, you could look at **show** command output to confirm the network topology. After it is confirmed, you could predict its normal working behavior based on your knowledge of LAN switching. As a result, you could predict where a frame sent by PC1 to PC2 should flow. To isolate the problem, you could look in the switch MAC tables to confirm the interfaces out which the frame should be forwarded, possibly then finding that the interface connected to PC2 has failed.

If you did conclude that an interface had failed, you still do not know the root cause: What caused the interface to fail? What could you do to fix that underlying problem? In that particular case, you would then need to broaden your thinking to any and all reasons why an interface might fail—from an unplugged cable, to electrical interference, to port security disabling the interface. **show** commands can either confirm that a specific root cause is the problem or at least give some hints as to the root cause.

The first example problem uses a simple LAN, with one subnet and no need for IP routing. However, many exam questions will include multiple IP subnets, with routers that must route IP packets between the hosts. In these cases, the troubleshooting process often begins with some analysis of how the Layer 3 routing process works when forwarding IP packets.

For example, the user of PC1 in Figure 10-2 can usually connect to the web server on the right by entering www.example.com in PC1's web browser. However, that web-browsing attempt fails right now. The user calls the help desk, and the problem is assigned to a network engineer to solve.

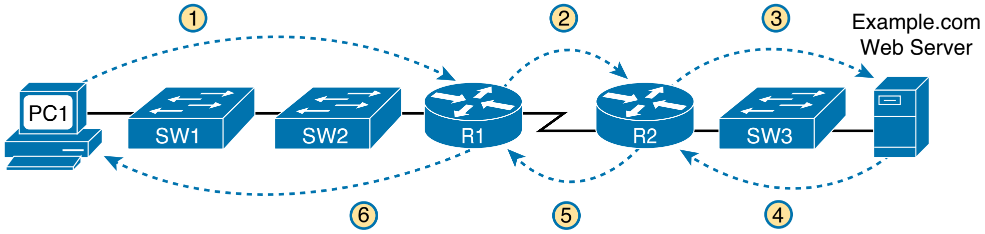


Figure 10-2 *Layer 3 Problem Isolation*

To begin the analysis, the network engineer can begin with the first tasks that would have to happen for a successful web-browsing session to occur. For example, the engineer would try to confirm that PC1 can resolve the host name (www.example.com) to the correct IP address used by the server on the right. At that point, the Layer 3 IP problem isolation process can proceed, to determine which of the six routing steps shown in the figure has failed. The routing steps shown in Figure 10-2 are as follows:

Step 1. PC1 sends the packet to its default gateway (R1) because the destination IP address (of the web server) is in a different subnet.

Step 2. R1 forwards the packet to R2 based on R1's routing table.

Step 3. R2 forwards the packet to the web server based on R2's routing table.

Step 4. The web server sends a packet back toward PC1 based on the web server's default gateway setting (R2).

Step 5. R2 forwards the packet destined for PC1 by forwarding the packet to R1 according to R2's routing table.

Step 6. R1 forwards the packet to PC1 based on R1's routing table.

Many engineers break down network problems as in this list, analyzing the Layer 3 path through the network, hop by hop, in both directions. This process helps you take the first attempt at problem isolation. When the analysis shows which hop in the layer path fails, you can then look further at those details. And if in this case the Layer 3 problem isolation process discovers that Step 1, 3, 4, or 6 fails, the root cause might be related to Ethernet.

For example, imagine that the Layer 3 analysis determined that PC1 cannot even send a packet to its default gateway (R1), meaning that Step 1 in Figure 10-2 fails. To further isolate the problem and find the root causes, the engineer would need to determine the following:

- The MAC address of PC1 and of R1's LAN interface
- The switch interfaces used on SW1 and SW2
- The interface status of each switch interface
- The VLANs that should be used
- The expected forwarding behavior of a frame sent by PC1 to R1 as the destination MAC address

By gathering and analyzing these facts, the engineer can most likely isolate the problem's root cause and fix it.

Troubleshooting as Covered in This Book

In the current version of the ICND1 and ICND2 exams, Cisco spreads troubleshooting topics across both exams. However, in the current versions of the exams (100-101 ICND1 and 200-101 ICND2), more of the troubleshooting sits in the ICND2 exam, with less in the ICND1 exam. As a result, this book has one chapter devoted to troubleshooting (this chapter), with some other smaller troubleshooting topics spread throughout different chapters. The companion *Cisco CCNA Routing and Switching ICND2 200-101 Official Cert Guide*, Academic Edition has many more troubleshooting elements by comparison.

The rest of this chapter examines troubleshooting related to Ethernet LANs, with four major topics. Of these, only the first topic, about the Cisco Discovery Protocol (CDP), presents completely new material. The other three topics discuss familiar topics, but with a troubleshooting approach. The topics include the following:

- **Cisco Discovery Protocol (CDP):** Used to confirm the documentation, and learn about the network topology, to predict normal operation of the network.
- **Examining interface status:** Interfaces must be in a working state before a switch will forward frames on the interface. You must determine whether an interface is working, as well as determine the potential root causes for a failed switch interface.
- **Analyzing where frames will be forwarded:** You must know how to analyze a switch's MAC address table and how to then predict how a switch will forward a particular frame.
- **Analyzing VLANs and VLAN trunking:** Keeping a Layer 2 switch focus, this last section looks at what can go wrong with VLANs and VLAN trunks.

Analyzing LAN Topology Using Cisco Discovery Protocol

The proprietary Cisco Discovery Protocol (CDP) discovers basic information about neighboring routers and switches without needing to know the passwords for the neighboring devices. To discover information, routers and switches send CDP messages out each of their interfaces. The messages essentially announce information about the device that sent the CDP message. Devices that support CDP learn information about others by listening for the advertisements sent by other devices.

As is so often the case, Cisco created CDP as a proprietary solution to meet a need for Cisco customers. Since that time, the IEEE has standardized the Link Layer Discovery Protocol (LLDP), which serves the same role. However, most enterprises that use Cisco routers and switches use CDP, with LLDP as an option, so this chapter focuses solely on CDP instead of LLDP.

From a troubleshooting perspective, CDP can be used to either confirm or fix the documentation shown in a network diagram, or even discover the devices and interfaces used in a network. Confirming that the network is actually cabled to match the network diagram is a good step to take before trying to predict the normal flow of data in a network.

10

On media that support multicasts at the data link layer (like Ethernet), CDP uses multicast frames; on other media, CDP sends a copy of the CDP update to any known data link addresses. So, any CDP-supporting device that shares a physical medium with another CDP-supporting device can learn about the other device.

CDP discovers several useful details from the neighboring Cisco devices:

Key Topic

- **Device identifier:** Typically the host name
- **Address list:** Network and data link addresses
- **Port identifier:** The interface on the remote router or switch on the other end of the link that sent the CDP advertisement
- **Capabilities list:** Information on what type of device it is (for example, a router or a switch)
- **Platform:** The model and OS level running on the device

Examining Information Learned by CDP

CDP has **show** commands that list information about neighbors, **show** commands that list information about how CDP is working, and configuration commands to disable and enable CDP. Table 10-1 lists the three **show** commands that list the most important CDP information.

Key Topic

Table 10-1 show cdp Commands That List Information About Neighbors

Command	Description
show cdp neighbors [*type number*]	Lists one summary line of information about each neighbor, or just the neighbor found on a specific interface if an interface was listed.
show cdp neighbors detail	Lists one large set (approximately 15 lines) of information, one set for every neighbor.
show cdp entry *name*	Lists the same information as the **show cdp neighbors detail** command, but only for the named neighbor (case sensitive).

> **NOTE** Cisco routers and switches support the same CDP commands, with the same parameters and same types of output.

The next example shows the power of the information in CDP commands. The example uses the network shown in Figure 10-3, with Example 10-1 that follows listing the output of several **show cdp** commands.

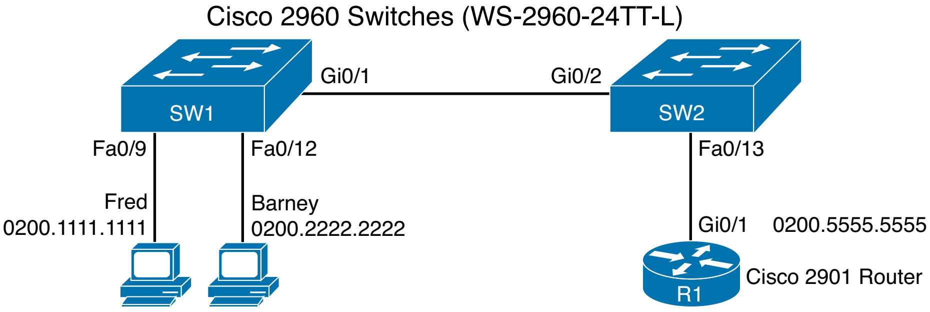


Figure 10-3 *Small Network Used in CDP Examples*

Example 10-1 show cdp *Command Examples: SW2*

```
! The show cdp neighbors command lists SW2's local interface, and both R1's
! and SW1's interfaces  (in the "port" column), along with other details.
!
SW2# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW1              Gig 0/2           170              S I   WS-C2960- Gig 0/1
R1               Fas 0/13          136             R S I  CISCO2901 Gig 0/1

SW2# show cdp neighbors detail
-------------------------
Device ID: SW1
Entry address(es):
  IP address: 172.16.1.1
Platform: cisco WS-C2960-24TT-L,  Capabilities: Switch IGMP
Interface: GigabitEthernet0/2,  Port ID (outgoing port): GigabitEthernet0/1
Holdtime : 161 sec
```

```
Version :
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(1)SE3, RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Wed 30-May-12 14:26 by prod_rel_team

advertisement version: 2
Protocol Hello:  OUI=0x00000C, Protocol ID=0x0112; payload len=27,
value=00000000FFFFFFFF010221FF000000000000018339D7B0E80FF0000
VTP Management Domain: ‘’
Native VLAN: 1
Duplex: full
Management address(es):
  IP address: 172.16.1.1

! This is a comment from the author: next lines are about R1.
-------------------------
Device ID: R1
Entry address(es):
  IP address: 10.1.1.9
Platform: Cisco CISCO2901/K9,  Capabilities: Router Switch IGMP
Interface: FastEthernet0/13,  Port ID (outgoing port): GigabitEthernet0/1
Holdtime : 127 sec

Version :
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.2(4)M1, RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 26-Jul-12 20:54 by prod_rel_team

advertisement version: 2
VTP Management Domain: ''
Duplex: full
Management address(es):
```

The example begins with the **show cdp neighbors** command, which lists one line per neighbor. Each line lists the most important topology information: the neighbor’s hostname (Device ID), the local device’s interface, and the neighboring device’s interface (under the Port heading). For example, SW2’s **show cdp neighbors** command lists an entry for SW1, with SW2’s local interface of Gi0/2 and SW1’s interface of Gi0/1 (see Figure 10-3 for reference). This command also lists the platform, identifying the specific model of the neighboring router or switch. So, even using this basic information, you could either construct a figure like Figure 10-3 or confirm that the details in the figure are correct.

The **show cdp neighbors detail** command lists additional details, such as the full name of the model of switch (WS-2960-24TT-L) and the IP address configured on the neighboring device.

NOTE The **show cdp entry** *name* command lists the exact same details shown in the output of the **show cdp neighbors detail** command, but for only the one neighbor listed in the command.

As you can see, you can sit on one device and discover a lot of information about a neighboring device—a fact that actually creates a security exposure. Cisco recommends that CDP be disabled on any interface that might not have a need for CDP. For switches, any switch port connected to another switch, a router, or to an IP phone should use CDP.

CDP can be disabled globally and per-interface. Per-interface, the **no cdp enable** and **cdp enable** interface subcommands toggle CDP off and on, respectively. Alternatively, the **no cdp run** and **cdp run** global commands toggle CDP off and on (respectively) for the entire switch.

Examining the Status of the CDP Protocols

CDP defines protocol messages that flow between devices. Cisco switches include a few commands that list statistics and other status information about how the CDP protocols are working, as summarized in Table 10-2 for easy reference.

Table 10-2 Commands Used to Verify CDP Operations

Command	Description
show cdp	States whether CDP is enabled globally, and lists the default update and holdtime timers.
show cdp interface [*type number*]	States whether CDP is enabled on each interface, or a single interface if the interface is listed, and states update and holdtime timers on those interfaces.
show cdp traffic	Lists global statistics for the number of CDP advertisements sent and received.

Example 10-2 lists sample output from each of the commands in Table 10-2, based on switch SW2 in Figure 10-3.

Example 10-2 **show cdp** *Commands That Show CDP Status*

```
SW2# show cdp
Global CDP information:
    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled

SW2# show cdp interface FastEthernet0/13
FastEthernet0/13 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
SW2# show cdp traffic
CDP counters :
    Total packets output: 304, Input: 305
    Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
    No memory: 0, Invalid packet: 0,
    CDP version 1 advertisements output: 0, Input: 0
    CDP version 2 advertisements output: 304, Input: 305
```

Analyzing Switch Interface Status

This section begins the third of five major sections in this chapter by looking at switch interfaces. That process begins with finding out whether each switch interface works. Unsurprisingly, Cisco switches do not use interfaces at all unless the interface is first considered to be in a functional or working state. Additionally, the switch interface might be in a working state, but intermittent problems might still be occurring.

This section begins by looking at the Cisco switch interface status codes and what they mean so that you can know whether an interface is working. The rest of this section then looks at those more unusual cases in which the interface is working, but not working well.

Interface Status Codes and Reasons for Nonworking States

Cisco switches actually use two different sets of interface status codes—one set of two codes (words) that use the same conventions as do router interface status codes, and another set with a single code (word). Both sets of status codes can determine whether an interface is working.

The switch **show interfaces** and **show interfaces description** commands list the two-code status just like routers. The two codes are named the *line status* and *protocol status*. They *generally* refer to whether Layer 1 is working (line status) and whether Layer 2 is working (protocol status), respectively. LAN switch interfaces typically show an interface with both codes with the same value, either "up" or "down."

NOTE This book refers to these two status codes in shorthand by just listing the two codes with a slash between them, such as "up/up."

The **show interfaces status** command lists a different single interface status code. This single interface status code corresponds to different combinations of the traditional two-code interface status codes and can be easily correlated to those codes. For example, the **show interfaces status** command lists a "connected" state for working interfaces. It corresponds to the up/up state seen with the **show interfaces** and **show interfaces description** commands.

Any interface state other than *connected* or *up/up* means that the switch will not forward or receive frames on the interface. Each nonworking interface state has a small set of root causes. Also, note that the exams could easily ask a question that showed only one or the other type of status code, so be prepared to see both types of status codes on the exams, and know the meanings of both. Table 10-3 lists the code combinations and some root causes that could have caused a particular interface status.

Key Topic

Table 10-3 LAN Switch Interface Status Codes

Line Status	Protocol Status	Interface Status	Typical Root Cause
Administratively Down	Down	disabled	The interface is configured with the **shutdown** command.
Down	Down	notconnect	No cable; bad cable; wrong cable pinouts; the speeds are mismatched on the two connected devices; the device on the other end of the cable is (a) powered off, (b) **shutdown**, or (c) error disabled.
Up	Down	notconnect	An interface up/down state is not expected on LAN switch physical interfaces.
Down	Down (err-disabled)	err-disabled	Port security has disabled the interface.
Up	Up	connected	The interface is working.

Most of the reasons for the notconnect state were covered earlier in this book. For example, using incorrect cabling pinouts, instead of the correct pinouts explained in Chapter 2, "Fundamentals of Ethernet LANs," causes a problem. However, one topic can be particularly difficult to troubleshoot—the possibility for both speed and duplex mismatches, as explained in the next section.

As you can see in the table, having a bad cable is just one of many reasons for the down/down state (or notconnect, per the **show interfaces status** command). Interestingly, the Cisco CCENT and CCNA exams do not focus much on cabling itself. However, for some examples of the root causes of cabling problems:

- The installation of any equipment that uses electricity, even non-IT equipment, can interfere with the transmission on the cabling, and make the link fail.
- The cable could be damaged, for example, if it lies under carpet. If the user's chair keeps squashing the cable, eventually the electrical signal can degrade.
- While optical cables do not suffer from EMI, someone can try to be helpful and move a fiber-optic cable out of the way—bending it too much. A bend into too tight a shape can prevent the cable from transmitting bits (called *macrobending*).

For the other interface states listed in Table 10-3, only the up/up (connected) state needs more discussion. An interface can be in a working state, and it might really be working—or it might be working in a degraded state. The next few topics discuss how to examine an up/up interface to find out whether it is working well or having problems.

Interface Speed and Duplex Issues

Many UTP-based Ethernet interfaces support multiple speeds, either full- or half-duplex, and support IEEE standard autonegotiation (as discussed in Chapter 6's section "Autonegotiation"). These same interfaces can also be configured to use a specific speed using the **speed** {**10** | **100** | **1000**} interface subcommand, and a specific duplex using the **duplex** {**half** | **full**} interface subcommand. With both configured, a switch or router disables the IEEE-standard autonegotiation process on that interface.

The **show interfaces** and **show interfaces status** commands list both the actual speed and duplex settings on an interface, as demonstrated in Example 10-3.

Key Topic

Example 10-3 *Displaying Speed and Duplex Settings on Switch Interfaces*

```
SW1# show interfaces status

Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1                        notconnect   1            auto   auto 10/100BaseTX
Fa0/2                        notconnect   1            auto   auto 10/100BaseTX
Fa0/3                        notconnect   1            auto   auto 10/100BaseTX
Fa0/4                        connected    1          a-full  a-100 10/100BaseTX
Fa0/5                        connected    1          a-full  a-100 10/100BaseTX
Fa0/6                        notconnect   1            auto   auto 10/100BaseTX
Fa0/7                        notconnect   1            auto   auto 10/100BaseTX
Fa0/8                        notconnect   1            auto   auto 10/100BaseTX
Fa0/9                        notconnect   1            auto   auto 10/100BaseTX
Fa0/10                       notconnect   1            auto   auto 10/100BaseTX
Fa0/11                       connected    1          a-full     10 10/100BaseTX
Fa0/12                       connected    1            half    100 10/100BaseTX
Fa0/13                       connected    1          a-full  a-100 10/100BaseTX
Fa0/14                       disabled     1            auto   auto 10/100BaseTX
```

```
Fa0/15                       notconnect   3            auto   auto 10/100BaseTX
Fa0/16                       notconnect   3            auto   auto 10/100BaseTX
Fa0/17                       connected    1          a-full  a-100 10/100BaseTX
Fa0/18                       notconnect   1            auto   auto 10/100BaseTX
Fa0/19                       notconnect   1            auto   auto 10/100BaseTX
Fa0/20                       notconnect   1            auto   auto 10/100BaseTX
Fa0/21                       notconnect   1            auto   auto 10/100BaseTX
Fa0/22                       notconnect   1            auto   auto 10/100BaseTX
Fa0/23                       notconnect   1            auto   auto 10/100BaseTX
Fa0/24                       notconnect   1            auto   auto 10/100BaseTX
Gi0/1                        connected    trunk        full   1000 10/100/1000BaseTX
Gi0/2                        notconnect   1            auto   auto 10/100/1000BaseTX

SW1# show interfaces fa0/13
FastEthernet0/13 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 0019.e86a.6f8d (bia 0019.e86a.6f8d)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mbps, media type is 10/100BaseTX
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:05, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     85022 packets input, 10008976 bytes, 0 no buffer
     Received 284 broadcasts (0 multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 281 multicast, 0 pause input
     0 input packets with dribble condition detected
     95226 packets output, 10849674 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

Although both commands in the example can be useful, only the **show interfaces status** command implies how the switch determined the speed and duplex settings. The command output lists autonegotiated settings with a prefix of **a-**. For example, **a-full** means full-duplex as autonegotiated, whereas **full** means full-duplex but as manually configured. The example shades the command output that implies that the switch's Fa0/12 interface's speed and duplex were not found through autonegotiation, but Fa0/13 did use autonegotiation. Note that the **show interfaces fa0/13** command (without the **status** option) simply lists the speed and duplex for interface Fast Ethernet 0/13, with nothing implying that the values were learned through autonegotiation.

When the IEEE autonegotiation process works on both devices, both devices agree to the fastest speed supported by both devices. Additionally, the devices use full-duplex if it is supported by both devices, or half-duplex if it is not. However, when one device has disabled autonegotiation, and the other device uses autonegotiation, the device using autonegotiation chooses the default duplex setting based on the current speed. The defaults are as follows:

Key Topic

- If the speed is not known through any means, use 10 Mbps, half-duplex.
- If the switch successfully senses the speed without IEEE autonegotiation, by just looking at the signal on the cable:
    - If the speed is 10 or 100 Mbps, default to use half-duplex.
    - If the speed is 11,000 Mbps, default to use full-duplex.

NOTE Ethernet interfaces using speeds faster than 1 Gbps always use full-duplex.

While autonegotiation works well, these defaults allow for the possibility of a difficult-to-troubleshoot problem called a *duplex mismatch*. Chapter 6's section "Autonegotiation" explains how both devices could use the same speed, so the devices would consider the link to be up, but one side would use half-duplex, and the other side would use full-duplex.

The next example shows a specific case that causes a duplex mismatch. In Figure 10-4, imagine that SW2's Gi0/2 interface was configured with the **speed 100** and **duplex full** commands (these settings are not recommended on a Gigabit-capable interface, by the way). On Cisco switches, configuring both the **speed** and **duplex** commands disables IEEE autonegotiation on that port. If SW1's Gi0/1 interface tries to use autonegotiation, SW1 would also use a speed of 100 Mbps, but default to use half-duplex. Example 10-4 shows the results of this specific case on SW1.

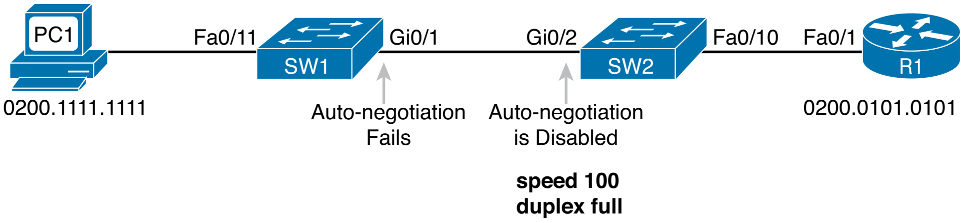


Figure 10-4 *Conditions to Create a Duplex Mismatch Between SW1 and SW2*

Example 10-4 *Confirming Duplex Mismatch on Switch SW1*

```
SW1# show interfaces gi0/1 status

Port      Name               Status       Vlan       Duplex  Speed Type
Gi0/1                        connected    trunk      a-half  a-100 10/100/1000BaseTX
```

First, focusing on the command output, the command confirms SW1's speed and duplex. It also lists a prefix of **a-** in the output, implying autonegotiation. Even with SW1 using autonegotiation defaults, the command still notes the values as being learned through autonegotiation.

Finding a duplex mismatch can be much more difficult than finding a speed mismatch, because *if the duplex settings do not match on the ends of an Ethernet segment, the switch interface will still be in a connected (up/up) state*. In this case, the interface works, but it might work poorly, with poor performance, and with symptoms of intermittent problems. The reason is that the device using half-duplex uses carrier sense multiple access collision detect (CSMA/CD) logic, waiting to send when receiving a frame, believing collisions occur when they physically do not—and actually stopping sending a frame because the switch thinks a collision occurred. With enough traffic load, the interface could be in a connect state, but it's extremely inefficient for passing traffic.

To identify duplex mismatch problems, check the duplex setting on each end of the link and watch for incrementing collision and late collision counters, as explained in the next section.

Common Layer 1 Problems on Working Interfaces

When the interface reaches the connect (up/up) state, the switch considers the interface to be working. The switch, of course, tries to use the interface, and at the same time, the switch keeps various interface counters. These interface counters can help identify problems that can occur even though the interface is in a connect state. This section explains some of the related concepts and a few of the most common problems.

Whenever the physical transmission has problems, the receiving device might receive a frame whose bits have changed values. These frames do not pass the error detection logic as implemented in the FCS field in the Ethernet trailer, as covered in Chapter 2. The receiving device discards the frame and counts it as some kind of *input error*. Cisco switches list this error as a CRC error, as highlighted in Example 10-5. (Cyclic redundancy check [CRC] is a term related to how the FCS math detects an error.)

Example 10-5 *Interface Counters for Layer 1 Problems*

```
SW1# show interfaces fa0/13
! lines omitted for brevity
     Received 284 broadcasts (0 multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 281 multicast, 0 pause input
     0 input packets with dribble condition detected
     95226 packets output, 10849674 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

The number of input errors, and the number of CRC errors, are just a few of the counters in the output of the **show interfaces** command. The challenge is to decide which counters you need to think about, which ones show that a problem is happening, and which ones are normal and of no concern.

The example highlights several of the counters as examples so that you can start to understand which ones point to problems and which ones are just counting normal events that are not problems. The following list shows a short description of each highlighted counter, in the order shown in the example:

Runts: Frames that did not meet the minimum frame size requirement (64 bytes, including the 18-byte destination MAC, source MAC, Type, and FCS). Can be caused by collisions.

Giants: Frames that exceed the maximum frame size requirement (1518 bytes, including the 18-byte destination MAC, source MAC, Type, and FCS).

Input Errors: A total of many counters, including runts, giants, no buffer, CRC, frame, overrun, and ignored counts.

CRC: Received frames that did not pass the FCS math; can be caused by collisions.

Frame: Received frames that have an illegal format, for example, ending with a partial byte; can be caused by collisions.

Packets Output: Total number of packets (frames) forwarded out the interface.

Output Errors: Total number of packets (frames) that the switch port tried to transmit, but for which some problem occurred.

Collisions: Counter of all collisions that occur when the interface is transmitting a frame.

Late Collisions: The subset of all collisions that happen after the 64th byte of the frame has been transmitted. (In a properly working Ethernet LAN, collisions should occur within the first 64 bytes; late collisions today often point to a duplex mismatch.)

Note that many of these counters occur as part of the CSMA/CD process used when half-duplex is enabled. Collisions occur as a normal part of the half-duplex logic imposed by CSMA/CD, so a switch interface with an increasing collisions counter might not even have a problem. However, one problem, called late collisions, points to the classic duplex mismatch problem.

If a LAN design follows cabling guidelines, all collisions should occur by the end of the 64th byte of any frame. When a switch has already sent 64 bytes of a frame, and the switch receives a frame on that same interface, the switch senses a collision. In this case, the collision is a late collision, and the switch increments the late collision counter in addition to the usual CSMA/CD actions to send a jam signal, wait a random time, and try again.

With a duplex mismatch, like the mismatch between SW1 and SW2 in Figure 10-4, the half-duplex interface will likely see the late collisions counter increment. Why? The half-duplex interface sends a frame (SW1), but the full-duplex neighbor (SW2) sends at any time, even after the 64th byte of the frame sent by the half-duplex switch. So, just keep repeating the **show interfaces** command, and if you see the late collisions counter incrementing on a half-duplex interface, you might have a duplex mismatch problem.

A working interface (in an up/up state) can still suffer from issues related to the physical cabling as well. The cabling problems might not be bad enough to cause a complete failure, but the transmission failures result in some frames failing to pass successfully over the cable. For example, excessive interference on the cable can cause the various input error counters to keep growing larger, especially the CRC counter. In particular, if the CRC errors grow, but the collisions counters do not, the problem might simply be interference on the cable. (The switch counts each collided frame as one form of input error as well.)

Predicting Where Switches Will Forward Frames

This section begins the fourth of five major sections in this chapter. This section looks at a key part of the troubleshooting process for Ethernet LANs: predicting where frames should go in the LAN so that you can compare what should happen versus what is actually happening in a LAN.

Predicting the Contents of the MAC Address Table

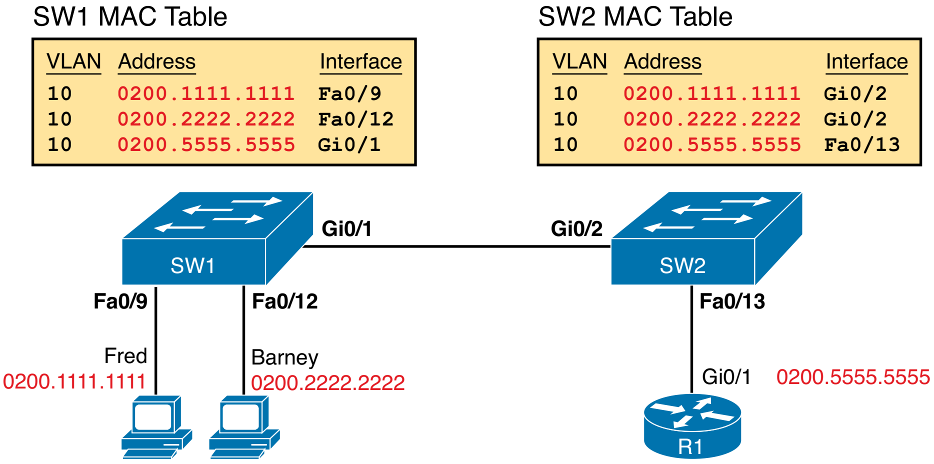
As explained in Chapter 6, "Building Ethernet LANs with Switches," switches learn MAC addresses and then use the entries in the MAC address table to make a forwarding/filtering decision for each frame. To know exactly how a particular switch will forward an Ethernet frame, you need to examine the MAC address table on a Cisco switch.

The **show mac address-table** EXEC command displays the contents of a switch's MAC address table. This command lists all MAC addresses currently known by the switch. The output includes some static overhead MAC addresses used by the switch and any statically configured MAC addresses, such as those configured with the port security feature. The command also lists all dynamically learned MAC addresses. If you want to see only the dynamically learned MAC address table entries, simply use the **show mac address-table dynamic** EXEC command.

> **NOTE** Some older switch IOS versions only support the older version of this command: **show mac-address-table.**

The more formal troubleshooting process begins with a mental process where you predict where frames should flow in the LAN. As an exercise, go back and review Figure 10-3 and try to create a MAC address table on paper for each switch. Include the MAC addresses for both PCs, as well as the Gi0/1 MAC address for R1. (Assume that all three are assigned to VLAN 10.) Then predict which interfaces would be used to forward a frame sent by Fred, Barney, and R1 to every other device.

The MAC table entries you predict in this case define where you think frames will flow. Even though this sample network in Figure 10-3 shows only one physical path through the Ethernet LAN, the exercise should be worthwhile, because it forces you to correlate what you'd expect to see in the MAC address table with how the switches forward frames. Figure 10-5 shows the resulting MAC table entries for PCs Fred and Barney, as well as for Router R1.



SW1 MAC Table

VLAN	Address	Interface
10	0200.1111.1111	Fa0/9
10	0200.2222.2222	Fa0/12
10	0200.5555.5555	Gi0/1

SW2 MAC Table

VLAN	Address	Interface
10	0200.1111.1111	Gi0/2
10	0200.2222.2222	Gi0/2
10	0200.5555.5555	Fa0/13

Figure 10-5 *Predictions for MAC Table Entries on SW1 and SW2*

While Figure 10-5 shows the concepts, Example 10-6 lists the same facts but in the form of the **show mac address-table dynamic** command on the switches. This command lists all dynamically learned MAC table entries on a switch, for all VLANs.

Example 10-6 *Examining SW1 and SW2 Dynamic MAC Address Table Entries*

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


Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  10    0200.1111.1111    DYNAMIC     Fa0/9
  10    0200.2222.2222    DYNAMIC     Fa0/12
  10    0200.5555.5555    DYNAMIC     Gi0/1
```

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

```
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  10    0200.1111.1111    DYNAMIC     Gi0/2
  10    0200.2222.2222    DYNAMIC     Gi0/2
  10    0200.5555.5555    DYNAMIC     Fa0/13
```

When predicting the MAC address table entries, you need to imagine a frame sent by a device to another device on the other side of the LAN and then determine *which switch ports the frame would enter* as it passes through the LAN. For example, if Barney sends a frame to Router R1, the frame would enter SW1's Fa0/12 interface, so SW1 has a MAC table entry that lists Barney's 0200.2222.2222 MAC address with Fa0/12. SW1 would forward Barney's frame to SW2, arriving on SW2's Gi0/2 interface, so SW2's MAC table lists Barney's MAC address (0200.2222.2222) with interface Gi0/2.

After you predict the expected contents of the MAC address tables, you can then examine what is actually happening on the switches, as described in the next section.

Analyzing the Forwarding Path

Troubleshooting revolves around three big ideas: predicting what should happen, determining what is happening that is different than what should happen, and figuring out why that different behavior is happening. This next section discusses how to look at what is actually happening in a VLAN based on those MAC address tables, first using a summary of switch forwarding logic and then showing an example.

The following list summarizes switch forwarding logic including the LAN switching features discussed in this book:

Key Topic

Step 1. Process functions on the incoming interface, if the interface is currently in an up/up (connected) state, as follows:

A. If configured, apply port security logic to filter the frame as appropriate.

B. If the port is an access port, determine the interface's access VLAN.

C. If the port is a trunk, determine the frame's tagged VLAN.

Step 2. Make a forwarding decision. Look for the frame's destination MAC address in the MAC address table, but only for entries in the VLAN identified in Step 1. If the destination MAC is...

A. **Found (unicast),** forward the frame out the only interface listed in the matched address table entry.

B. **Not found (unicast),** flood the frame out all other access ports (except the incoming port) in that same VLAN, plus out trunks that have not restricted the VLAN from that trunk (as discussed in Chapter 9, "Implementing Ethernet Virtual LANs," as related to the **show interfaces trunk** command).

C. **Broadcast,** flood the frame, with the same rules as the previous step.

For an example of this process, consider a frame sent by Barney to its default gateway, R1 (0200.5555.5555). Using the steps just listed, the following occurs:

Step 1. Input interface processing:

A. The port does not happen to have port security enabled.

B. SW1 receives the frame on its Fa0/12 interface, an access port in VLAN 10.

Step 2. Make a forwarding decision: SW1 looks in its MAC address table for entries in VLAN 10:

A. SW1 finds an entry (known unicast) for 0200.5555.5555, associated with VLAN 10, outgoing interface Gi0/1, so SW1 forwards the frame only out interface Gi0/1. (This link is a VLAN trunk, so SW1 adds a VLAN 10 tag to the 802.1Q trunking header.)

At this point, the frame with source 0200.2222.2222 (Barney) and destination 0200.5555.5555 (R1) is on its way to SW2. You can then apply the same logic for SW2, as follows:

Step 1. Input interface processing:

A. The port does not happen to have port security enabled.

B. SW2 receives the frame on its Gi0/2 interface, a trunk; the frame lists a tag of VLAN 10. (SW2 will remove the 802.1Q header as well.)

Step 2. Make a forwarding decision: SW2 looks for its MAC table for entries in VLAN 10:

A. SW2 finds an entry (known unicast) for 0200.5555.5555, associated with VLAN 10, outgoing interface Fa0/13, so SW2 forwards the frame only out interface Fa0/13.

At this point, the frame should be on its way, over the Ethernet cable between SW2 and R1.

Port Security and Filtering

When tracing the path a frame takes through LAN switches, different kinds of filters can discard frames, even when all the interfaces are up. For example, LAN switches can use filters called access control lists (ACL) that filter based on the source and destination MAC address, discarding some frames. Additionally, routers can filter IP packets using IP ACLs. (This book does not discuss ACLs for LAN switches, but it does discuss IP ACLs for routers in Chapter 22, “Basic IPv4 Access Control Lists,” and Chapter 23, “Advanced IPv4 ACLs and Device Security.”)

Additionally, port security, as discussed in Chapter 8, “Configuring Ethernet Switching,” also filters frames. In some cases, you can easily tell that port security has taken action, because port security shuts down the interface. However, in other cases, port security leaves the interface up, but simply discards the offending traffic. From a troubleshooting perspective, a port security configuration that leaves the interface up, but still discards frames, requires the network engineer to look closely at port security status, rather than just looking at interfaces and the MAC address table.

As a reminder, port security allows three violation modes (**shutdown**, **protect**, and **restrict**), but only the default setting of **shutdown** causes the switch to err-disable the interface.

For example, consider a case in which someone takes a working network and adds port security to filter frames sent by Barney. Use Figure 10-3 or 10-5, both of which show the same topology. Barney sends frames into SW1’s Fa0/12 port, which is now configured with port security. The port security configuration considers frames with Barney’s source MAC address as a violation, and it uses a violation mode set to **protect**.

What happens? SW1 now discards all frames sourced by Barney's MAC address. But SW1 does not disable any interfaces. The **show interfaces** or **show interfaces status** command on SW1 shows no changes to the interface status, and no evidence of what happened. You would need to look further at port security (**show port-security interface**) to find evidence that port security was discarding the frames sent by Barney.

The MAC address table gives some hints that port security might be enabled. Because port security manages the MAC addresses, any MAC addresses associated with a port on which port security is enabled show up as static MAC addresses. As a result, the **show mac address-table dynamic** command does not list MAC addresses off these interfaces on which port security is enabled. However, the **show mac address-table** and **show mac address-table static** commands do list these static MAC addresses.

Analyzing VLANs and VLAN Trunks

A switch's forwarding process, as discussed earlier in the section "Analyzing the Forwarding Path," depends 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 of the five major sections in this chapter focuses on VLAN and VLAN trunking issues, and specifically issues that impact the frame switching process. The four potential issues are as follows:

Step 1. Identify all access interfaces and their assigned access VLANs and reassign into the correct VLANs as needed.

Step 2. Determine whether the VLANs both exist (configured or learned with 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. Ensure that for any links that should use trunking, one switch does not think it is trunking, while the other switch does not think it is trunking because of an unfortunate choice of configuration settings.

Ensuring That the Right Access Interfaces Are in the Right VLANs

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 10-4 can be particularly helpful in this process.

Key Topic

Table 10-4 Commands That Can Find Access Ports and VLANs

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

EXEC Command	Description
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

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 PC, 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 Not Being Defined

Switches do not forward frames for VLANs that are (a) not configured or (b) configured but disabled (shutdown). This section summarizes the best ways to confirm that a switch knows that a particular VLAN exists, and if it exists, determines the state of the VLAN.

First, on the issue of whether a VLAN is defined, 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. This book purposefully ignores VTP as much as possible, so 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 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, the problem might be that the VLAN simply needs to be defined. If so, follow the VLAN configuration process as covered in detail in Chapter 9.

Access VLANs Being Disabled

For any existing VLANs, also verify that 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 shutdown. 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 10-7 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 10-7 *Enabling and Disabling VLANs on a Switch*

```
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)#
```

Check the Allowed VLAN List on Both Ends of a Trunk

The next item, and the one that follows, both occur when an engineer makes some poor configuration choices on a VLAN trunk. In real life, you should instead just configure the trunk correctly, as outlined in Chapter 9's section "VLAN Trunking Configuration" and the section that follows it, "Controlling Which VLANs Can Be Supported on a Trunk." But for the exams, you should be ready to notice a couple of oddities that happen with some unfortunate configuration choices on trunks.

First, it is possible to configure a different allowed VLAN list on the opposite ends of a VLAN trunk. When mismatched, the trunk cannot pass traffic for that VLAN.

Figure 10-6 shows an example. Both switches have defined VLANs 1 through 10, so both by default include VLANs 1 through 10 in their allowed VLAN list. However, SW2 has been configured with a **switchport trunk allowed vlan remove 10** command, removing VLAN 10 from SW2's G0/2 allowed list. In this case, SW1, which still allows VLAN 10, acts as normal, tagging and forwarding frames in VLAN 10 (Step 1 in the figure). SW2 simply discards any VLAN 10 frames received on that trunk (Step 2), because SW2 does not allow VLAN 10 traffic on that trunk.

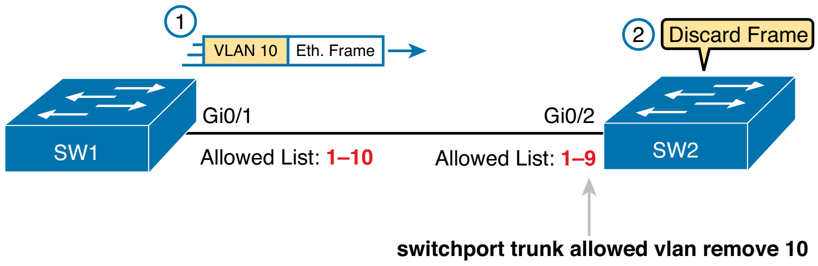


Figure 10-6 *Mismatched VLAN-Allowed Lists on a Trunk*

And to emphasize the point, you cannot see this problem from just one side of the trunk or the other. The **show interfaces trunk** command output on both sides looks completely normal. You can only notice the problem by comparing the allowed lists on both ends of the trunk.

To compare the lists, you need to look at the second of three lists of VLANs listed by the **show interfaces trunk** command, as highlighted in the example output in Example 10-8. The highlighted text shows the second section, which lists VLANs that meet these criteria: the VLANs that exist on the switch, that are not shutdown, and that are not removed from the allowed list.

Example 10-8 *Second Set of VLANs: Existing, Not Shut Down, and Allowed*

```
SW2# show interfaces trunk

Port        Mode         Encapsulation  Status        Native vlan
Gi0/2       desirable    802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/2       1-4094

Port        Vlans allowed and active in management domain
Gi0/2       1-9

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/2       1-9
```

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), as well as the fact that both switches operate as "static access" ports. Example 10-9 highlights those parts of the output from this command.

Example 10-9 *Operational Trunking State*

```
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 10-7 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 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.

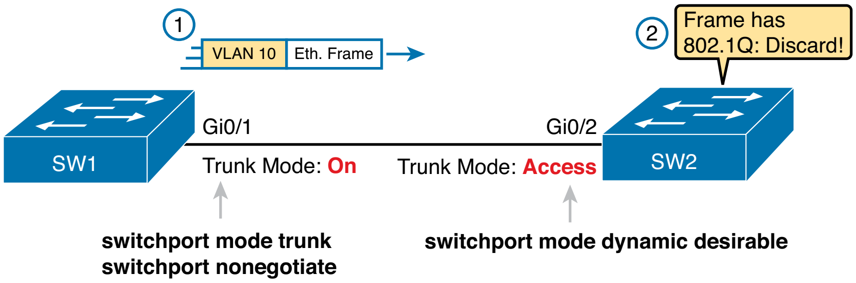


Figure 10-7 *Mismatched Trunking Operational States*

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 (Step 1). 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.

First, 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.

Review Activities

Chapter Summary

- The proprietary Cisco Discovery Protocol (CDP) discovers basic information about neighboring routers and switches without needing to know the passwords for the neighboring devices.
- To discover information, routers and switches send CDP messages out each of their interfaces. The messages essentially announce information about the device that sent the CDP message. Devices that support CDP learn information about others by listening for the advertisements sent by other devices.
- From a troubleshooting perspective, CDP can be used to either confirm or fix the documentation shown in a network diagram, or even discover the devices and interfaces used in a network.
- CDP discovers several useful details from the neighboring Cisco devices:
  - **Device identifier:** Typically the hostname
  - **Address list:** Network and data-link addresses
  - **Port identifier:** The interface on the remote router or switch on the other end of the link that sent the CDP advertisement
  - **Capabilities list:** Information about what type of device it is (for example, a router or a switch)
  - **Platform:** The model and OS level running in the device
- The switch **show interfaces** and **show interfaces description** commands list the two-code status just like routers. The two codes are named the *line status* and *protocol status*.
- The **show interfaces status** command lists a different single interface status code. This single interface status code corresponds to different combinations of the traditional two-code interface status codes and can be easily correlated to those codes.
- Any interface state other than *connected* or *up/up* means that the switch will not forward or receive frames on the interface.
- The following list shows a short description of each counter displayed in the output of a **show interfaces** command.
  - **Runts:** Frames that did not meet the minimum frame size requirement (64 bytes, including the 18-byte destination MAC, source MAC, Type, and FCS). May be caused by collisions.
  - **Giants:** Frames that exceed the maximum frame size requirement (1518 bytes, including the 18-byte destination MAC, source MAC, Type, and FCS).
  - **Input Errors:** A total of many counters, including runts, giants, no buffer, CRC, frame, overrun, and ignored counts.
  - **CRC:** Received frames that did not pass the FCS math; may be caused by collisions.
  - **Frame:** Received frames that have an illegal format (for example, ending with a partial byte); may be caused by collisions.
  - **Output Errors:** The total number of packets (frames) that the switch port tried to transmit but for which some problem occurred.
  - **Packets Output:** The total number of packets (frames) forwarded out the interface.
  - **Collisions:** The counter of all collisions that occur when the interface is transmitting a frame.
  - **Late Collisions:** The subset of all collisions that happen after the 64th byte of the frame has been transmitted. (In a properly working Ethernet LAN, collisions should occur within the first 64 bytes; late collisions today often point to a duplex mismatch.)

Review Questions

Answer these review questions. You can find the answers at the bottom of the last page of the chapter. For thorough explanations, see DVD Appendix C, "Answers to Review Questions."

1. Imagine that a switch connects through an Ethernet cable to a router, and the router's host name is Hannah. Which of the following commands could tell you information about the IOS version on Hannah without establishing a Telnet connection to Hannah? (Choose two answers.)

   A. **show neighbors Hannah**

   B. **show cdp**

   C. **show cdp neighbors**

   D. **show cdp neighbors Hannah**

   E. **show cdp entry Hannah**

   F. **show cdp neighbors detail**

2. A switch is cabled to a router whose host name is Hannah. Which of the following CDP commands could identify Hannah's model of hardware? (Choose two answers.)

   A. **show neighbors**

   B. **show neighbors Hannah**

   C. **show cdp**

   D. **show cdp interface**

   E. **show cdp neighbors**

   F. **show cdp entry Hannah**

3. The output of the **show interfaces status** command on a 2960 switch shows interface Fa0/1 in a "disabled" state. Which of the following is true about interface Fa0/1? (Choose three answers.)

   A. The interface is configured with the **shutdown** command.

   B. The **show interfaces fa0/1** command will list the interface with two status codes of administratively down and line protocol down.

   C. The **show interfaces fa0/1** command will list the interface with two status codes of up and down.

   D. The interface cannot currently be used to forward frames.

   E. The interface can currently be used to forward frames.

4. Switch SW1 uses its Gigabit 0/1 interface to connect to switch SW2's Gigabit 0/2 interface. SW2's Gi0/2 interface is configured with the **speed 1000** and **duplex full** commands. SW1 uses all defaults for interface configuration commands on its Gi0/1 interface. Which of the following are true about the link after it comes up? (Choose two answers.)

   A. The link works at 1000 Mbps (1 Gbps).

   B. SW1 attempts to run at 10 Mbps because SW2 has effectively disabled IEEE standard autonegotiation.

   C. The link runs at 1 Gbps, but SW1 uses half-duplex and SW2 uses full-duplex.

   D. Both switches use full-duplex.

5. The following line of output was taken from a **show interfaces fa0/1** command:

```
Full-duplex, 100Mbps, media type is 10/100BaseTX
```

Which of the following are true about the interface? (Choose two answers.)

A. The speed was definitely configured with the **speed 100** interface subcommand.

B. The speed might have been configured with the **speed 100** interface subcommand.

C. The duplex was definitely configured with the **duplex full** interface subcommand.

D. The duplex might have been configured with the **duplex full** interface subcommand.

6. Which of the following commands list the MAC address table entries for MAC addresses configured by port security? (Choose two answers.)

A. **show mac address-table dynamic**

B. **show mac address-table**

C. **show mac address-table static**

D. **show mac address-table port-security**

7. On a Cisco Catalyst switch, you issue a **show mac address-table** command. Which of the following answers list information you would likely see in most lines of output? (Choose two answers.)

A. A MAC address

B. An IP address

C. A VLAN ID

D. Type (broadcast, multicast, or unicast)

8. Layer 2 switches SW1 and SW2 connect through a link, with port G0/1 on SW1 and port G0/2 on SW2. The network engineer wants to use 802.1Q trunking on this link. The **show interfaces g0/1 switchport** command on SW1 shows the output listed here:

```
SW1# show interfaces gigabit0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
```

Which of the following must be true on switch SW2's G0/2 port?

A. The operational state per the **show interfaces switchport** command must be "trunk."

B. The administrative state per the **show interfaces switchport** command must be "trunk."

C. SW2 must use the **switchport mode trunk** configuration command on G0/2, or the link will not use trunking.

D. SW2 can use the **switchport mode dynamic auto** configuration command as one option to make the link use trunking.

Review All the Key Topics

Review the most important topics from this chapter, noted with the Key Topic icon. Table 10-5 lists these key topics and where each is discussed.

Key Topic

Table 10-5 Key Topics for Chapter 10

Key Topic Element	Description	Page Number
List	Information gathered by CDP	239
Table 10-1	Three CDP **show** commands that list information about neighbors	240
Table 10-3	Two types of interface state terms and their meanings	243
Example 10-3	Example that shows how to find the speed and duplex settings, as well as whether they were learned through autonegotiation	244
List	Defaults for IEEE autonegotiation	246
List	Summary of switch forwarding steps	250
Table 10-4	Commands that identify access VLANs assigned to ports	252

Complete the Tables and Lists from Memory

Print a copy of DVD Appendix M, "Memory Tables," or at least the section for this chapter, and complete the tables and lists from memory. DVD Appendix N, "Memory Tables Answer Key," includes completed tables and lists for you to check your work.

Definitions of Key Terms

After your first reading of the chapter, try to define these key terms, but do not be concerned about getting them all correct at that time. Chapter 30 directs you in how to use these terms for late-stage preparation for the exam.

CDP neighbor, up and up, connected, error disabled, problem isolation, root cause, duplex mismatch

Command References

Tables 10-6 and 10-7 list only commands specifically mentioned in this chapter, but the command references at the end of Chapters 8 and 9 also cover some related commands. Table 10-6 lists and briefly describes the configuration commands used in this chapter.

Table 10-6 Commands for Catalyst 2960 Switch Configuration

Command	Description
shutdown **no shutdown**	Interface subcommands that administratively disable and enable an interface, respectively.
switchport port-security violation {**protect** | **restrict** | **shutdown**}	Interface subcommand that tells the switch what to do if an inappropriate MAC address tries to access the network through a secure switch port.
cdp run **no cdp run**	Global commands that enable and disable, respectively, CDP for the entire switch or router.

Command	Description
cdp enable **no cdp enable**	Interface subcommands that enable and disable, respectively, CDP for a particular interface.
speed {**auto** | **10** | **100** | **1000**}	Interface subcommand that manually sets the interface speed.
duplex {**auto** | **full** | **half**}	Interface subcommand that manually sets the interface duplex.

Table 10-7 lists and briefly describes the EXEC commands used in this chapter.

Table 10-7 Chapter 10 EXEC Command Reference

Command	Description
show mac address-table [**dynamic** | **static**] [**address** *hw-addr*] [**interface** *interface-id*] [**vlan** *vlan-id*]	Displays the MAC address table. The security option displays information about the restricted or static settings.
show port-security [**interface** *interface-id*] [**address**]	Displays information about security options configured on an interface.
show cdp neighbors [*type number*]	Lists one summary line of information about each neighbor, or just the neighbor found on a specific interface if an interface was listed.
show cdp neighbors detail	Lists one large set of information (approximately 15 lines) for every neighbor.
show cdp entry *name*	Displays the same information as the **show cdp neighbors detail** command, but only for the named neighbor.
show cdp	States whether CDP is enabled globally, and lists the default update and holdtime timers.
show cdp interface [*type number*]	States whether CDP is enabled on each interface, or a single interface if the interface is listed, and states update and holdtime timers on those interfaces.
show cdp traffic	Displays global statistics for the number of CDP advertisements sent and received.
show interfaces [*type number*]	Displays detailed information about interface status, settings, and counters.
show interfaces description	Displays one line of information per interface, with a two-item status (similar to the **show interfaces** command status), and includes any description that is configured on the interfaces.
show interfaces [*type number*] **status**	Displays summary information about interface status and settings, including actual speed and duplex, a single-item status code, and whether the interface was autonegotiated.
show interfaces [*type number*] **switchport**	Displays a large variety of configuration settings and current operational status, including VLAN trunking details, access and voice VLAN, and native VLAN.

Command	Description
show interfaces [*type number*] **trunk**	Lists information about the currently operational trunks (or just for the trunk listed in the command) and the VLANs supported on those trunks.
show vlan brief, **show vlan**	Lists each VLAN and all interfaces assigned to that VLAN but does not include trunks.
show vlan id *num*	Lists both access and trunk ports in the VLAN.
show vtp status	Lists the current VTP status, including the current mode.

Answers to Review Questions:

1 E and F **2** E and F **3** A, B, and D **4** A and D **5** B and D **6** B and C **7** A and C **8** D