APPENDIX B

ICND2 Exam Updates

Over time, reader feedback allows Cisco Press to gauge which exam topics give our readers the most problems. In addition, Cisco may make small changes in the breadth of exam topics or in emphasis of certain topics. To assist readers with those topics, the author creates new materials to clarify and expand on those challenging exam topics.

You can check for an updated version of this appendix at http://www.ciscopress.com/title/9781587143731.

The document you are reading is Version 3.0 of this appendix. The original version (1.0) contained no technical topics. Version 2.0 included details about the Rapid Spanning Tree Protocol (RSTP). This version, Version 3.0, added more content to several other topics: Static Routes, FHRP, and OSPFv2.

Table B-1 lists the major topic headings in this chapter, along with a suggestion about the best place in your reading plan to go ahead and read that part of this appendix.

Key Topic

Table B-1 Topics and When to Best Read Them

Topic	Chapter
Rapid STP (IEEE 802.1w) Concepts	2
Static IPv4 Routes	5
Troubleshooting IPv4 Routing with ACLs	5
First Hop Redundancy Protocols	6
OSPF Default Routes and Interface Configuration	8
EIGRP Issues on Multipoint and Physical Interfaces	14

Rapid STP (IEEE 802.1w) Concepts

As described in detail in Chapters 1 and 2, the IEEE defines STP in the 802.1D IEEE standard. The IEEE improved the 802.1D protocol with the definition of RSTP, as defined in standard 802.1w.

Key Topic

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

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

RSTP can be deployed alongside traditional 802.1D STP switches, with RSTP features working in switches that support it, and traditional 802.1D STP features working in the switches that support only STP.

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

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

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

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

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

Table B-2 Port Roles in 802.1w RSTP

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

RSTP and the Alternate (Root) Port

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

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

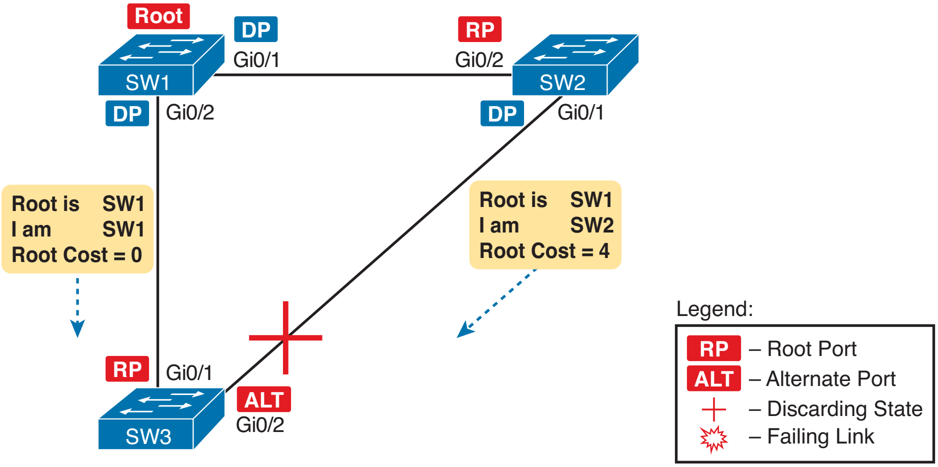

Figure B-1 *Example of SW3 Making G0/2 Become an Alternate Port*

An alternate port basically works like the second-best option for root port. The alternate port can take over for the former root port, often very rapidly, without requiring a wait in other interim RSTP states. For instance, when the root port fails, or when hellos stop arriving on the original root port, the switch moves the original root port to a disabled role and transitions to a discarding state (the equivalent of STP's blocking state). Without waiting on any timers, the best alternate port then becomes the new root port. That new root port also does not need to spend time in other states, like learning state, instead moving immediately to forwarding state.

Figure B-2 shows an example of RSTP convergence in which the link between SW1 and SW3 fails. The figure begins with Step 1 as the event that causes the link to fail.

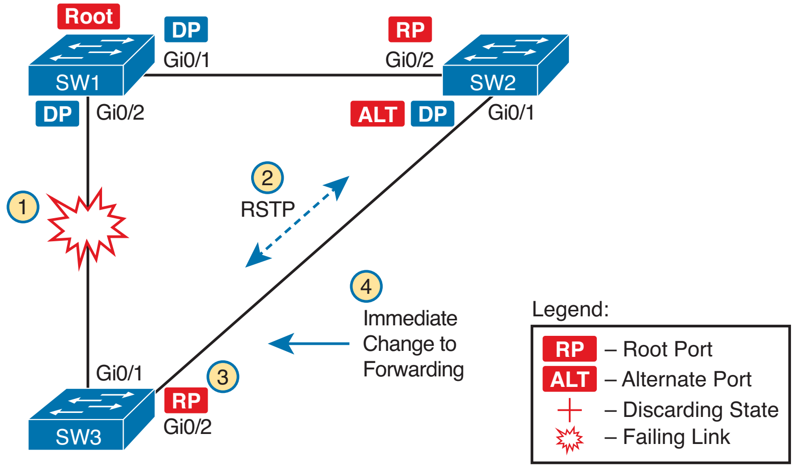

Figure B-2 *Convergence Events with SW3 G0/1 Failure*

Following the steps in the figure:

Step 1. The link between SW1 and SW3 fails.

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

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

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

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

RSTP States and Processes

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

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

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

Table B-3 shows the list of STP and RSTP states for comparison purposes.

Table B-3 Port States Compared: 802.1D STP and 802.1w RSTP

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

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

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

RSTP Backup (Designated) Ports

To complete the discussion, next consider the idea of a backup for a designated port. This concept, called a *backup port*, can be a bit confusing at first, because it only happens in designs that are a little unlikely today. The reason is that a design must use hubs, which then allows the possibility that one switch connects more than one port to the same collision domain.

Figure B-3 shows an example. SW3 and SW4 both connect to the same hub. SW4's port F0/1 happens to win the election as designated port (DP). The other port on SW4 that connects to the same collision domain, F0/2, acts as a backup port.

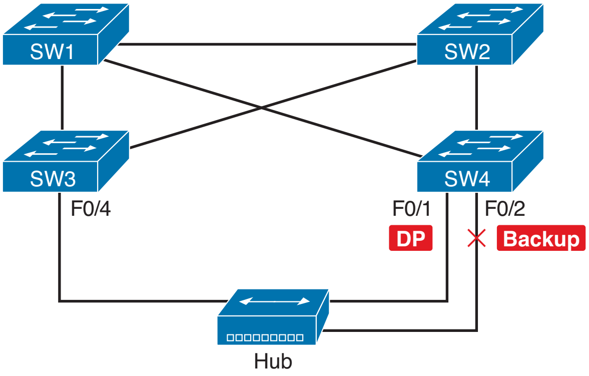

Figure B-3 *RSTP Backup Port Example*

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

RSTP Port Types

The final concept to mention before moving on to the RSTP configuration and verification topics relates to some terms RSTP uses to refer to different types of ports and the links that connect to those ports.

To begin, consider the basic figure of Figure B-4. It shows several links between two switches. RSTP considers these links to be point-to-point links and the ports connected to them to be point-to-point ports, because the link connects exactly two devices (points).

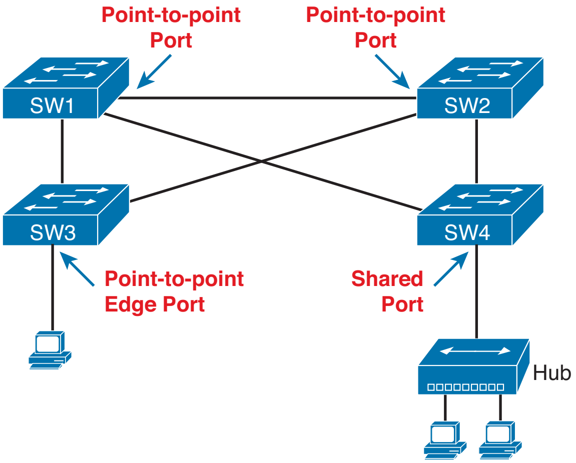

Figure B-4 *RSTP Link Types*

RSTP further classifies point-to-point ports into two categories. Point-to-point ports that connect two switches are not at the edge of the network and are simply called *point-to-*

point ports. Ports that instead connect to a single endpoint device at the edge of the network, like a PC or server, are called *point-to-point edge ports*, or simply *edge ports*. In Figure B-4, SW3's switch port connected to a PC is an edge port.

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

Identifying RSTP Through Configuration and Verification

The next few pages focus on how to configure and verify RSTP, but with emphasis on comparisons between RSTP and STP operations on Catalyst switches. The reason for focusing on the comparisons is related to the ICND2 exam topic that mentions RSTP:

> Indentify enhanced switching technologies: RSTP

Interestingly, only a few differences exist between STP and RSTP as seen in Catalyst switch configuration and verification commands. This section explains the configuration and verification of RSTP, with emphasis on how to identify RSTP features.

Identifying the STP Mode on a Catalyst Switch

Cisco Catalyst switches support an STP mode, as configured with the **spanning-tree mode** global configuration command. Based on this command's setting, the switch is either using 802.1D STP or 802.1w RSTP, as noted in Table B-4.

Key Topic

Table B-4 Cisco Catalyst STP Configuration Modes

Parameter on spanning-tree mode Command	Uses STP or RSTP?	Protocol Listed in Command Output	Description
pvst	STP	ieee	Default; per-VLAN spanning-tree instance
rapid-pvst	RSTP	rstp	Like PVST, but uses RSTP rules instead of STP for each STP instance
mst	RSTP	mst	Creates multiple RSTP instances but does not require 1 instance per each VLAN

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

Before looking at an example of the output, review the topology in Figure B-5. The remaining RSTP examples in this appendix use this topology. In the RSTP examples in this appendix, SW1 will become root, and SW3 will block on one port (G0/2), as shown.

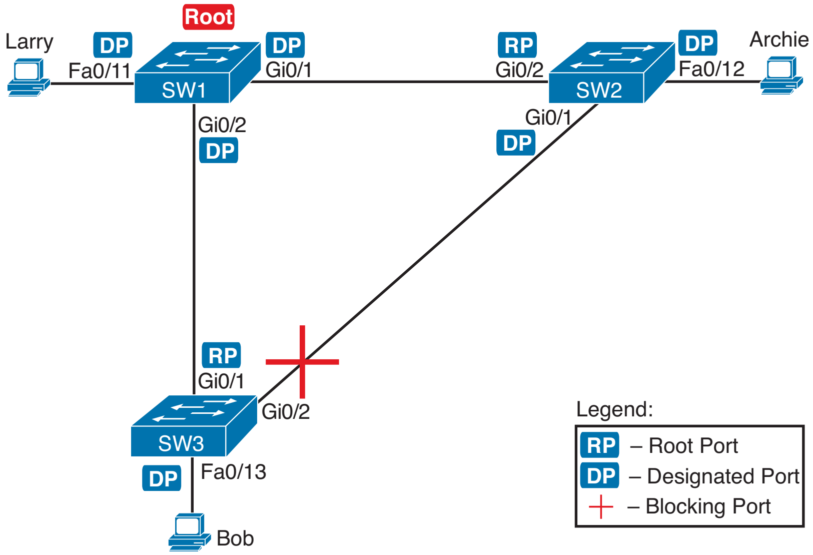

Figure B-5 *Network Topology for STP and RSTP Examples*

The first example focuses on VLAN 10, with all switches using 802.1D STP and the default setting of **spanning-tree mode pvst**. This setting creates an instance of STP per VLAN (which is the per-VLAN part of the name) and uses 802.1D STP. Each switch places the port connected to the PC into VLAN 10 and enables both PortFast and BPDU Guard. (See Chapter 2 for a review of PortFast and BPDU Guard.) Example B-1 shows a sample configuration from switch SW3, with identical interface subcommands configured on SW1's F0/11 and SW2's F0/12 ports, respectively.

Example B-1 *Sample Configuration from Switch SW3*

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

Building configuration...

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

At this point, the three switches use 802.1D STP because all use the default PVST mode. Example B-2 shows the evidence of STP's work, with only subtle and indirect clues that STP happens to be in use.

Example B-2 *Output That Confirms the Use of 802.1D STP on Switch SW3*

```
SW3# show spanning-tree vlan 10

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

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

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

SW3# show spanning-tree vlan 10 bridge

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

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

To migrate this small network to use RSTP, configure the **spanning-tree mode rapid-pvst** command. This continues the use of per-VLAN spanning-tree instances, but it applies RSTP logic to each STP instance. Example B-3 shows the output of the same two commands from Example B-2 after configuring the **spanning-tree mode rapid-pvst** command on all three switches.

Example B-3 *Output that Confirms the Use of 802.1w RSTP on Switch SW3*

```
SW3# show spanning-tree vlan 10

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

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

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

SW3# show spanning-tree vlan 10 bridge

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

Pay close attention to the differences between the 802.1D STP output in Example B-2 and the 802.1w RSTP output in Example B-3. Literally, the only difference is rstp instead of ieee in one place in the output of each of the two commands listed. In this case, rstp refers to the configuration of the **spanning-tree mode rapid-pvst** global config command, which implied the use of RSTP.

RSTP Port Roles

RSTP adds two port roles to STP: the alternate port and the backup port. Example B-4 repeats an excerpt from the **show spanning-tree vlan 10** command on switch SW3 to show an example of the alternate port role. SW3 (as seen earlier in Figure B-5) is not the root switch, with G0/1 as its root port and G0/2 port as an alternate port.

Example B-4   *Output Confirming SW3's Root Port and Alternate Port Roles*

```
SW3# show spanning-tree vlan 10
! Lines omitted for brevity
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/13              Desg FWD 19        128.13   P2p Edge
Gi0/1               Root FWD 4         128.25   P2p
Gi0/2               Altn BLK 4         128.26   P2p
```

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

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

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

For instance, just compare the output of Example B-2 and Example B-4. Example B-2 shows output for this same SW3, with the same parameters, except that all switches used PVST mode, meaning all the switches used STP. Example B-2's output still lists SW3's G0/2 as Altn, meaning alternate. Example B-4 uses the same scenario, with the same topology, except all switches use RSTP, and it also shows an alternate port.

RSTP Port States

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

For example, scan back to the most recent RSTP example (Example B-4), to the line for SW3's port G0/2. Then look for the column with heading STS, which refers to the status or state. The output shows G0/2 is listed as BLK, or blocking. In theory, because SW3 uses RSTP, the port state ought to be discarding, but the switch IOS continues to use the older notation of BLK for blocking.

Just as one more bit of evidence, the command **show spanning-tree vlan 10 interface gigabitethernet0/2 state** lists the STP or RSTP port state with the state fully spelled out. Example B-5 shows this command, taken from SW3, for interface G0/2. Note the fully spelled-out *blocking* term instead of the RSTP term *discarding*.

Example B-5 *SW3, an RSTP Switch, Continues to Use the Old Blocking Term*

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

VLAN0010             blocking
```

Port Types

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

Table B-5 RSTP Port Types

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

[1] Cisco recommends against using this combination for fear of causing loops.

You can easily find the RSTP port types in the output of several commands, including the same **show spanning-tree** command in Example B-6. Example B-6 lists output from switch SW2, with a hub added off SW2's F0/18 port (not shown in Figure B-5). The hub was added so that the output in Example B-6 lists a shared port (noted as Shr) to go along with the point-to-point ports (noted as P2p).

Example B-6 *RSTP Port Types*

```
SW2# show spanning-tree vlan 10

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

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

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

For exam prep, again note an odd fact about the highlighted output in Example B-6: The port type details appear in the output when using both STP and RSTP. For example, refer to Example B-2 again, which shows output from SW3 when using STP (when configured for PVST mode). The Type column also identifies point-to-point and edge interfaces.

Static IPv4 Routes

NOTE The best time to read this section is with Chapter 5, "Troubleshooting IPv4 Routing Part II." The *Cisco CCENT/CCNA ICND1 100-101 Official Cert Guide* introduces static IPv4 routes. This next section repeats some of that coverage, for those who might not have the ICND1 book, while adding a few details about how to troubleshoot how a router chooses which routes to add to the routing table based on the administrative distance.

Static Route Configuration

IOS allows the definition of individual static routes using the **ip route** global configuration command. Every **ip route** command defines a destination that can be matched, usually with a subnet ID and mask. The command also lists the forwarding instructions, typically listing either the outgoing interface or the next-hop router's IP address. IOS then takes that information and adds that route to the IP routing table.

As an example, Figure B-6 shows a small IP network. The diagram actually holds a subset of Figure 16-3, from Chapter 16, with some of the unrelated details removed. The figure shows only the details related to a static route on R1, for subnet 172.16.2.0/24, which sits on the far right. To create that static route on R1, R1 will configure the subnet ID and mask, and either R1's outgoing interface (S0/0/0), or R2 as the next-hop router IP address (172.16.4.2).

Key Topic

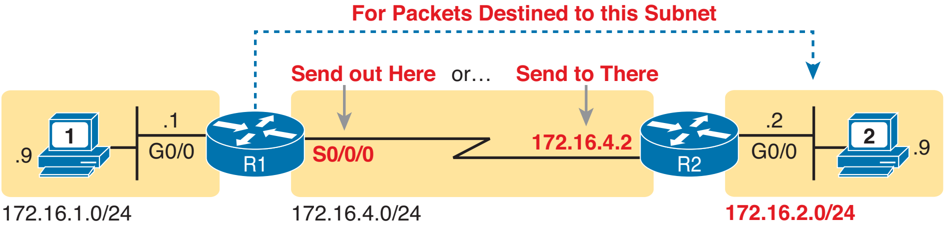

Figure B-6 *Static Route Configuration Concept*

Example B-7 shows the configuration of a couple of sample static routes. In particular, it shows routes on Router R1 in Figure B-7, for the two subnets on the right side of the figure.

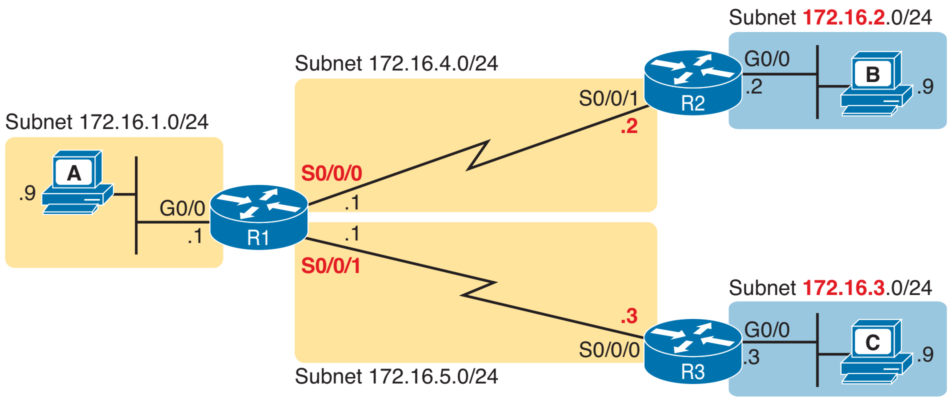

Figure B-7 *Sample Network Used in Static Route Configuration Examples*

Example B-7 *Static Routes Added to R1*

```
ip route 172.16.2.0 255.255.255.0 172.16.4.2
ip route 172.16.3.0 255.255.255.0 S0/0/1
```

The two example **ip route** commands show the two different styles. The first command shows subnet 172.16.2.0, mask 255.255.255.0, which sits on a LAN near Router R2. That same first command lists 172.16.4.2, R2's IP address, as the next-hop router. This route basically says this: To send packets to the subnet off Router R2, send them to R2.

The second route has the same kind of logic, but instead of identifying the next router by IP address, it lists the local router's outgoing interface. This route basically states: To send packets to the subnet off Router R3, send them out my own local S0/0/1 interface (which happens to connect to R3).

The routes created by these two **ip route** commands actually look a little different in the IP routing table. Both are static routes. However, the route that used the outgoing interface configuration is also noted as a connected route; this is just a quirk of the output of the **show ip route** command.

Example B-8 lists these two routes using the **show ip route static** command. This command lists the details of static routes only, but it also lists a few statistics about all IPv4 routes. For example, the example shows two lines, for the two static routes configured in Example B-7, but statistics state that this route has routes for ten subnets.

Example B-8 *Static Routes Added to R1*

```
R1# show ip route static
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
! lines omitted for brevity
Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 10 subnets, 2 masks
S        172.16.2.0/24 [1/0] via 172.16.4.2
S        172.16.3.0/24 is directly connected, Serial0/0/1
```

IOS adds and removes these static routes dynamically over time, based on whether the outgoing interface is working. For example, in this case, if R1's S0/0/1 interface fails, R1 removes the static route to 172.16.3.0/24 from the IPv4 routing table. Later, when the interface comes up again, IOS adds the route back to the routing table. Also, note that the **ip route** command also supports the **permanent** keyword, which tells IOS to leave the static route in the routing table, even when the associated interface fails.

Finally, if using static routes and not using any dynamic routing protocols at all, all routers would need to have some static routes configured. For example, at this point, in the network in Figure B-7, PC A would not be able to receive packets back from PC B, because Router R2 does not have a route for PC A's subnet. R2 would need static routes for other subnets, as would R3.

Static Routes with No Competing Routes

First, if the configured route has no competing routes, the router still checks a few rules before adding the route to its IP routing table. First, assume that the static route in question has no competition—that is, the routing protocol has not learned any routes for that exact same subnet, and no other static routes exist for that exact same subnet. Even with no competition, IOS considers the following before adding the route to its routing table:

- For **ip route** commands that list an outgoing interface, that interface must be in an up/up state.
- For **ip route** commands that list a next-hop IP address, the local router must have a route to reach that next-hop address.

For example, earlier in Example B-7, R1's command **ip route 172.16.2.0 255.255.255.0 172.16.4.2** defines a static route. Assume there were no competing routes and all links were working. Based on this route, R1 looks at its IP routing table and finds a route matching next-hop address 172.16.4.2—R1's connected route for subnet 172.16.4.0/24. As a result, R1 adds the static route to subnet 172.16.2.0/24. Later, if R1's S0/0/0 failed, R1 would remove its connected route to 172.16.4.0/24, which would then cause R1 to remove its static route to 172.16.2.0/24.

You can also configure a static route so that IOS ignores these basic checks, always putting the IP route in the routing table. To do so, just use the **permanent** keyword on the **ip route** command. For example, by adding the **permanent** keyword to the end of the two commands in Example B-7, as demonstrated in Example B-9, R1 would now add these routes, regardless of whether the two WAN links were up.

Example B-9 *Permanently Adding Static Routes to the IP Routing Table (Router R1)*

```
ip route 172.16.2.0 255.255.255.0 172.16.4.2 permanent
ip route 172.16.3.0 255.255.255.0 S0/0/1 permanent
```

Note that although the **permanent** keyword lets the router keep the route in the routing table without checking the outgoing interface or route to the next-hop address, it does not magically fix a broken route. For example, if the outgoing interface fails, the route will remain in the routing table, but the router can't forward packets because the outgoing interface is down.

Static Routes with Competing Routes

Next, consider the case in which a static route competes with other static routes or routes learned by a routing protocol. That is, the **ip route** command defines a route to a subnet, but the router also knows of other static or dynamically learned routes to reach that same subnet. In these cases, the router must first decide which routing source has the better administrative distance, with lower being better, and then use the route learned from the better source. (See Table 8-6 in Chapter 8, "Implementing OSPF for IPv4," for a list of administrative distance values.)

To see how that works, consider the example illustrated in Figure B-8, which shows a branch office with two WAN links: one very fast Gigabit Ethernet link and one rather slow (but cheap) T1. In this design, the network uses OSPFv2 over the primary link, learning a route for subnet 172.16.2.0/24. R1 also defines a static route over the backup link to that exact same subnet, so R1 must choose whether to use the static route or the OSPF-learned route.

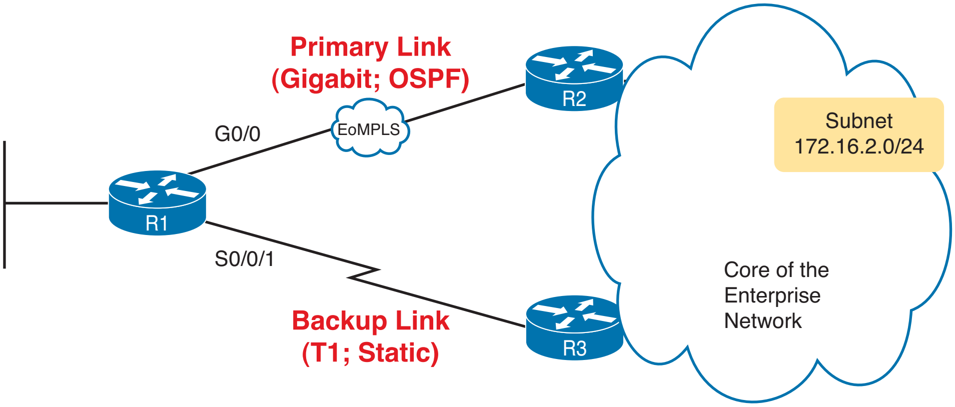

Figure B-8 *Using a Floating Static Route to Key Subnet 172.16.2.0/24*

IOS considers static routes better than OSPF-learned routes. By default, IOS gives static routes an administrative distance of 1 and OSPF routes an administrative distance of 110. Using these defaults in Figure B-8, R1 would use the lower path to reach subnet 172.16.2.0/24 in this case, which is not the intended design. Instead, the engineer prefers to use the OSPF-learned routes over the much-faster primary link, and use the static route over the backup link only as needed when the primary link fails.

To instead prefer the OSPF routes, the configuration would need to change the administrative distance settings and use what many networkers call a floating static route. A *floating static* route floats or moves into and out of the IP routing table depending on whether the better (lower) administrative distance route learned by the routing protocol happens to exist currently. Basically, the router ignores the static route during times when the better routing protocol route is known.

To implement a floating static route, just override the default administrative distance on the static route, making the value larger than the default administrative distance of the routing protocol. For example, the **ip route 172.16.2.0 255.255.255.0 172.16.5.3 130** command on R1 would do exactly that, setting the static route's administrative distance to 130. As long as the primary link stays up, and OSPF on R1 learns a route for 172.16.2.0/24, with administrative distance of 110, R1 ignores the static route.

Finally, note that while the **show ip route** command lists the administrative distance of most routes, as the first of two numbers inside two brackets, the **show ip route** subnet command plainly lists the administrative distance. Example B-10 shows a sample, matching this most recent example.

Example B-10 *Displaying the Administrative Distance of the Static Route*

```
R1# show ip route static
! Legend omitted for brevity

      172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
S        172.16.2.0/24 is directly connected, Serial0/0/1

R1# show ip route 172.16.2.0
Routing entry for 172.16.2.0/24
  Known via "static", distance 130, metric 0 (connected)
  Routing Descriptor Blocks:
  * directly connected, via Serial0/0/1
      Route metric is 0, traffic share count is 1
```

Troubleshooting IPv4 Routing with ACLs

NOTE This material about troubleshooting IP in the presence of ACLs is best read along with Chapter 5.

Access Control Lists (ACL) can filter packets as they flow through a network. This section adds a few perspectives about how ACLs impact troubleshooting, particularly commands used from the router CLI.

Pinging over Serial Links that Have ACLs

Figure B-9 illustrates a simple network topology with two routers connected to a serial link. Note that four IP ACLs exist, named A, B, C, and D, as noted by the thick arrows in the drawing. That is, ACL A is an outbound ACL on R1's S0/0/0, ACL B is an inbound ACL on R2's S0/0/1, and so on.

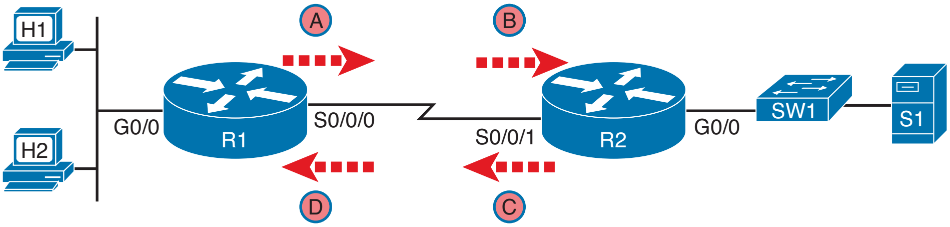

Figure B-9 *Sample Network with IP ACLs in Four Locations*

First, consider the case in which host H1 pings the IP address of host H2. Assuming that H1 and H2 are in the same subnet, router R1 does not even receive the IP packet. Even if router R1 had ACLs on its G0/0 interface, R1 has no chance of filtering the packet with an ACL.

Next, consider the case in which host H1 pings server S1's IP address. All four ACLs may filter the packets: ACLs A and B could filter the ICMP Echo Request messages going to the server, and ACLs C and D might filter the ICMP Echo Reply messages flowing back from the server to host H1.

Router ping Commands Bypass Outgoing ACL Logic

When you issue the **ping** command on the router, you must think about a quirk of how Cisco IOS works: Routers do not filter packets they create themselves. For example, imagine that a user connects to R1's CLI using SSH. At that point, the user issues a **ping** command for S1. The packets flow from R1 to S1 and back again. However, R1, having created the ICMP Echo Request messages, bypasses its own outgoing ACL logic of ACL A. The idea is that a user on R1 would not intend to filter its own packets, so R1 chooses to not filter the packets created by the **ping** command issued on R1.

Figure B-10 summarizes these ideas. Router R1 pings server S1. ACL A still exists as an outgoing ACL on Router R1. However, the figure shows ACL A as a light shade of gray to imply that it is ignored in this case.

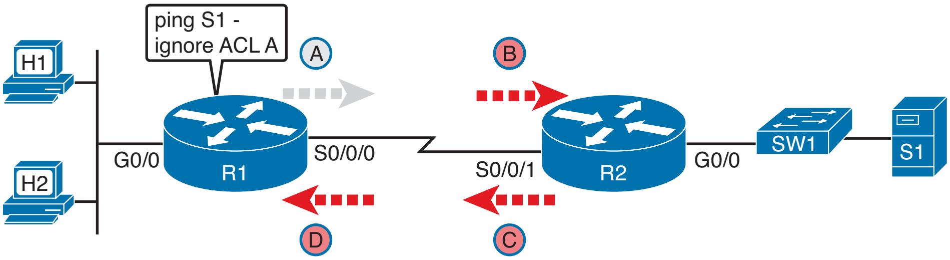

Figure B-10 *R1 Ignores Outgoing ACL for Packets Created by Its Own* **ping** *Command*

Self ping over Serial Links

Yet another scenario called the self ping points out another oddity with IOS. The term *self ping* refers to a ping of an address on the same device. For a self ping to succeed over a point-to-point serial link, the local router will send the IP packet (holding the ICMP Echo message) out the physical interface. Additionally, the physical link must be working, the local router needs the correct configuration, and the neighboring device needs the correct configuration. When the self ping succeeds, the packet holding the ICMP Echo Request actually leaves the router, with the next device sending the packet back. As a result, the router that issued the **ping** command receives its own ICMP Echo Request, at which point it must reply to the request.

Figure B-11 shows an example of a self ping of Router R1's own IP address on a point-to-point serial link. The link is up, with both ends using HDLC and the IP addresses shown in the figure. R1 begins by issuing a **ping 172.16.4.1** command. The basic flow works like this;

note that the descriptions ignore the data link headers because they are unimportant to the discussion:

Step 1. R1 builds the ICMP Echo Request, encapsulates it in an IP packet with destination address 172.16.4.1, and forwards the packet out S0/0/0.

Step 2. R2 receives the packet, sees destination address 172.16.4.1, and performs IP routing on the packet. As a result, R2 forwards the packet out S0/0/1, right back to router R1.

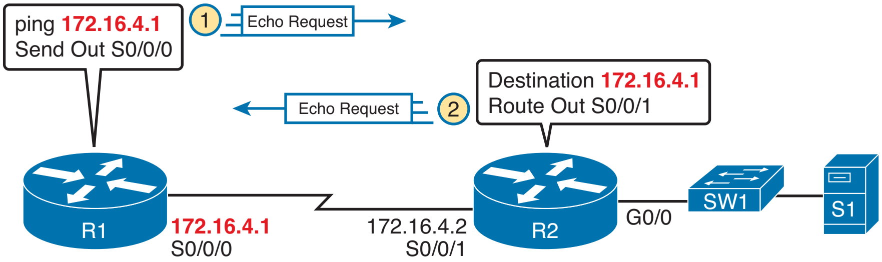

Figure B-11 *The First Steps in a Self Ping on R1, for R1's S0/0/0 IP Address*

Basically, R1 sends a packet destined to itself, and the other router (R2) treats the packet like any other IP packet, routing it to the correct destination (specifically, R2 does not issue the ICMP Echo Reply; it simply routes the packet back to R1).

To pull all these concepts together, now think about all four of those ACLs (refer to Figure B-9) while also thinking about what happens with the self ping in Figure B-11. R1's ping of its own serial IP address sends a packet that travels from R1 to R2 and back, so it could be filtered by ACLs B, C, and D. (R2 does not create a packet in this case—it forwards a packet created by R1—so R2 does indeed consider its outbound ACL C for this packet.) So, for R1's self ping of its serial interface IP address to work, these facts must be true:

- The link must work at Layers 1, 2, and 3. Specifically, both routers must have a working (up/up) serial interface, with correct IPv4 addresses configured.
- ACLs B, C, and D must permit the ICMP Echo Request and Reply packets.

So, when troubleshooting, if you choose to use self pings and they fail, do not forget to check to see whether the ACLs have filtered the ICMP traffic.

Self Ping of a Router Ethernet Interface IP Address

A self ping of a router's own Ethernet interface IP address works mostly like a self ping of a router's serial IP address, but with a couple of twists.

- The interface that uses that IP address must be working (in an up/up state); otherwise, the ping fails.
- The router does not forward the ICMP messages physically out the interface, so security features on neighboring switches (like port security) or routers (like ACLs) cannot possibly filter the messages used by the **ping** command.

- An incoming IP ACL on the local router can filter the router self ping of an Ethernet-based IP address.

Figure B-12 walks through an example. In this case, R2 issues a **ping 172.16.2.2** command to ping its own G0/0 IP address. Just like with a self ping on serial links, R2 creates the ICMP Echo Request. However, R2 basically processes the ping down its own TCP/IP stack and back up again, with the ICMP Echo never leaving the router's Ethernet interface.

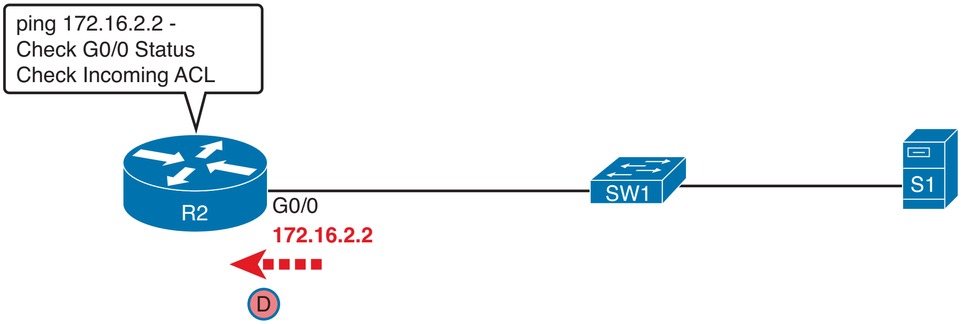

Figure B-12 *Self Ping of a Router's Ethernet Address*

ping of a Neighboring Device over Ethernet

Finally, as a brief review, routers use their IP ARP table to find the MAC address of another device that sits in the same LAN-based subnet. If the router has that other host's IP and MAC address in its ARP table, the router can encapsulate IP packets inside an Ethernet frame, putting that device's MAC address in the frame as the destination MAC address. If not, the router uses ARP messages to learn the neighboring device's MAC address.

Example B-11 shows R2's ARP table from Figure B-12. It shows only two entries, but you can identify the entry for R2's own interface IP address (172.16.2.2) based on the – in the output. The – means that the entry will not time out. (The other entry, 0200.3333.3333, happens to be for server S1, and in this case, the entry had not been used for 35 minutes.)

Example B-11 *Displaying a Router's IP ARP Table*

```
R2# show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.16.2.2              -   0200.2222.2222  ARPA   GigabitEthernet0/0
Internet  172.16.2.9             35   0200.3333.3333  ARPA   GigabitEthernet0/0
```

If the user sits at R2 and issues a **ping 172.16.2.9** command to ping server S1, R2 does the following:

1. Creates an IP packet, destination 172.16.2.9
2. Encapsulates the packet in an Ethernet frame, destination 0200.3333.3333 (S1's MAC address)
3. Sends the frame

First-Hop Redundancy Protocols

NOTE This material about FHRPs is best read along with Chapter 6, "Creating Redundant First-Hop Routers."

Influencing the HSRP Active Router Choice Using Tracking

Chapter 6 shows how to configure the HSRP priority so that one router will be preferred as the active router in an HSRP group. For example, Example 6-1 in Chapter 6 shows two routers in the same HSRP group, with R1 using a slightly better (numerically higher) priority of 110 and R2 using default priority 100, so that R1 becomes the active router when both routers are up and working. Figure B-13 shows a similar design, with Router R1 as active (priority 110) and R2 as standby (priority 100), using HSRP virtual IP address 10.1.1.1.

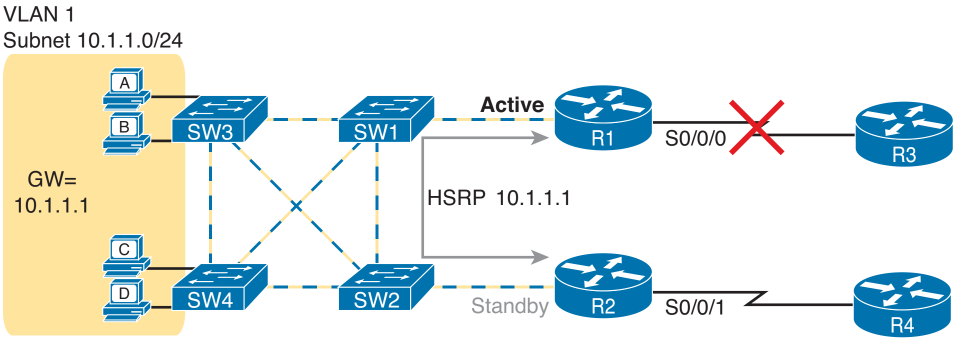

Figure B-13 *Design that Benefits from HSRP Tracking*

The network design benefits from another HSRP feature: interface tracking. IOS can track the state of an interface, with a variable for the interface as being either up or down. Then, you can change the HSRP priority value based on tracking variables, changing HSRP's choice of which router is primary based on other events and status inside the router.

For example, notice the big X over the upper WAN link in Figure B-13. What happens when R1's WAN link is down? Clearly, the WAN path through R2 and R4 should probably be used. However, all the hosts on the left still use R1 as their default gateway, when using R2 would clearly be more efficient. Instead, the HSRP configuration could be changed as follows:

- Set the priority values as noted earlier, so that under normal operation, R1 is active: R1 = 110, R2 = 100.
- R1 tracks its S0/0/0 interface, such that when S0/0/0 fails, R1 lowers its HSRP priority by 20.
- R2 tracks its S0/0/1 interface, such that when S0/0/1 fails, R2 lowers its HSRP priority by 20.
- When the standby router priority becomes better (higher) than the currently active router, take over the role of HSRP active (a feature called *preemption*).

Example B-12 completes the picture of HSRP interface tracking. Example B-12 shows R1's basic HSRP configuration, with tracking of the WAN interface as described here, and enables preemption. (Similar configuration would need to be added to R2 as well.) As a result, when all links work, R1 remains the active HSRP router, with priority 110. If R1's

WAN link then fails, R1's priority falls to 90 and R2's remains at 100, so R2 preempts R1's active role so that R2 takes over as the active HSRP router.

Example B-12 *HSRP Configuration on R1*

```
interface GigabitEthernet0/0
 ip address 10.1.1.9
 standby version 2
 standby ip 10.1.1.1
 standby 1 priority 110
 standby 1 track serial0/0/0 20
 standby 1 preempt
```

First Hop Redundancy Protocols (FHRP) all provide some level of tracking. For example, HSRP can track interfaces and use more complex object tracking that considers multiple factors to reach a decision.

Identifying FHRPs Based on Various Terms and Facts

The official exam topics for the ICND2 and CCNA exams mention that you must be ready to recognize FHRPs. This next short topic introduces a few miscellaneous facts about the FHRPs to provide some more comparison points, so you can more easily identify one FHRP versus another.

First, Table B-6 lists many facts introduced in Chapter 6, plus a few new facts, for the sake of comparison. For example, each FHRP uses slightly different terminology to refer to the router that is currently acting as the default gateway, and which router is waiting to take over. Each has a different maximum number of groups per LAN interface. Also, HSRP and GLBP use a virtual IP address that is not the interface IP address, whereas VRRP actually allows the use of one of the router's interface IP addresses as the virtual IP address. Table B-6 summarizes these points for easier comparison and study.

Table B-6 Comparison Points for FHRPs

Fact	HSRP	VRRP	GLBP
Main interface subcommand to configure this feature	standby	vrrp	glbp
Term for the router currently responding to ARP requests as the default gateway	Active	Master	Active
Term for the router waiting to take over for another	Standby	Backup	Standby or Listen[1]
The gateway IP address always differs from the router's interface IP address	Yes	No	Yes
Maximum number of concurrently configured groups per interface	16	255	4

[1] GLBP has two key roles: active virtual gateway (AVG) and forwarder. The AVG role is described with an active router and one or more standby routers; the forwarder role has one active router and one or more listening routers.

The virtual MAC address format also differs between FHRPs, so knowledge of the formats can be useful when scanning **show** command output. For example, if you scan back to the HSRP and GLBP **show** command examples in Chapter 6, you will see MAC addresses that begin with 0000.0C9F (HSRP) and 0000.B4 (GLBP). Knowing these prefixes can help you recognize which FHRP happens to be used. Also, the virtual MAC addresses contain information about the HSRP, VRRP, or GLBP group number hidden in the virtual MAC. Table B-7 summarizes the formats; note that the HSRP format differs depending on the version used.

Table B-7 FHRP Virtual MAC Address Formats

FHRP	Format	XX Value	YY Value
HSRP V1	`0000.0C07.ACXX`	HSRP group (2 hex digits)	N/A
HSRP V2	`0000.0C9F.FXXX`	HSRP group (3 hex digits)	N/A
VRRP	`0000.5E00.01XX`	VRRP group (2 hex digits)	N/A
GLBP	`0007.B40X.XXYY`	GLBP group (3 hex digits)	01, 02, 03, 04

OSPF Default Routes and Interface Configuration

NOTE The best time to read this section is when reading Chapter 8, "Implementing OSPF for IPv4." The default route material in this appendix is a copy of the *Cisco CCENT/CCNA ICND1 100-101 Official Cert Guide*'s Chapter 17 material on the same topic, and is repeated here as a reminder. This appendix also contains new coverage of OSPFv2, specifically of OSPFv2 interface configuration.

OSPF Default Routes

In some cases, routers benefit from using a default route. This section looks at a strategy for using default IP routes, one in which an OSPF router creates a default route and also advertises it with OSPF, so that other routers learn default routes dynamically.

The most classic case for using a routing protocol to advertise a default route has to do with an enterprise's connection to the Internet. As a strategy, the enterprise engineer uses these design goals:

- All routers learn specific routes for subnets inside the company; a default route is not needed when forwarding packets to these destinations.
- One router connects to the Internet, and it has a default route that points toward the Internet.
- All routers should dynamically learn a default route, used for all traffic going to the Internet, so that all packets destined to locations in the Internet go to the one router connected to the Internet.

Figure B-14 shows the idea of how OSPF advertises the default route, with the specific OSPF configuration. In this case, a company connects to an ISP with their Router R1. That router uses the OSPF **default-information originate** command (Step 1). As a result, the router advertises a default route using OSPF (Step 2) to the remote routers (B1, B2, and B3).

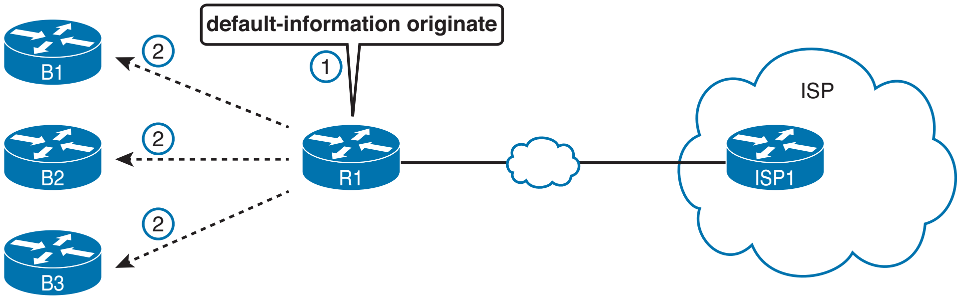

Figure B-14 *Using OSPF to Create and Flood a Default Route*

Figure B-15 shows the default routes that result from OSPF's advertisements in Figure B-14. On the far left, the three branch routers all have OSPF-learned default routes, pointing to R1. R1 itself also needs a default route, pointing to the ISP router, so that R1 can forward all Internet-bound traffic to the ISP.

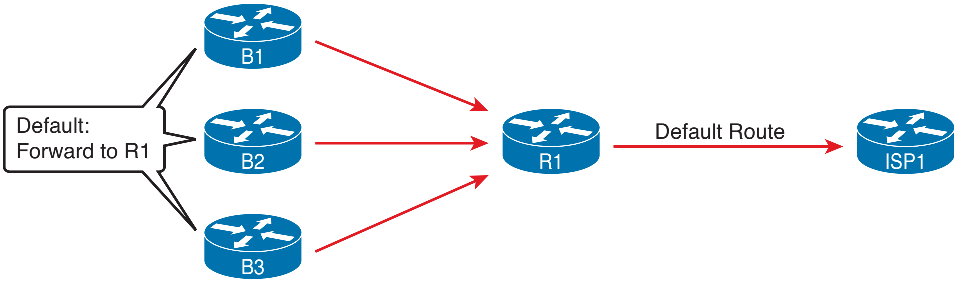

Figure B-15 *Default Routes Resulting from the* **default-information originate** *Command*

Finally, this feature gives the engineer control over when the router originates this default route. First, R1 needs a default route, either defined as a static default route or learned from the ISP. The **default-information originate** command then tells the R1 to advertise a default route when its own default route is working, and to advertise it as down when its own default route fails.

NOTE Interestingly, the **default-information originate always** router subcommand tells the router to always advertise the default route, no matter whether the router's default route is working or not.

OSPFv2 Interface Configuration

Cisco IOS supports two styles of configuration for OSPFv2:

- Traditional configuration using the **network** subcommand in router OSPF mode
- Newer configuration using the **ip ospf** subcommand in interface mode

The *Cisco CCENT/CCNA ICND1 100-101 Official Cert Guide* and Chapter 8 of this book show only the traditional configuration method for OSPFv2. This section introduces the newer configuration style.

NOTE OSPFv3 (that is, OSPF for IPv6) uses one configuration style, one that mirrors the interface configuration style shown here. See Chapter 17 for the OSPFv3 configuration.

Configuring the ip ospf Interface Subcommand

The newer style of configuration works just like the old, except in how the configuration enables OSPF directly on the interface. The traditional OSPFv2 configuration enables OSPFv2 on an interface, but indirectly, using the **network** command in OSPF configuration mode. IOS interpreted the parameters of the **network** command to decide on which interfaces to enable OSPF. With the new configuration, you simply add the **ip ospf** *process-id* **area** *area-id* interface subcommand to an interface to enable OSPFv2 on that interface.

Before looking at an example, refer to Chapter 8, Figures 8-16 and 8-17, along with Example 8-1 that follows them. Figure B-16 repeats Figure 8-17 for reference. Simply refamiliarize yourself with the traditional configuration in those examples, in preparation for comparing the older configuration with the new-style configuration.

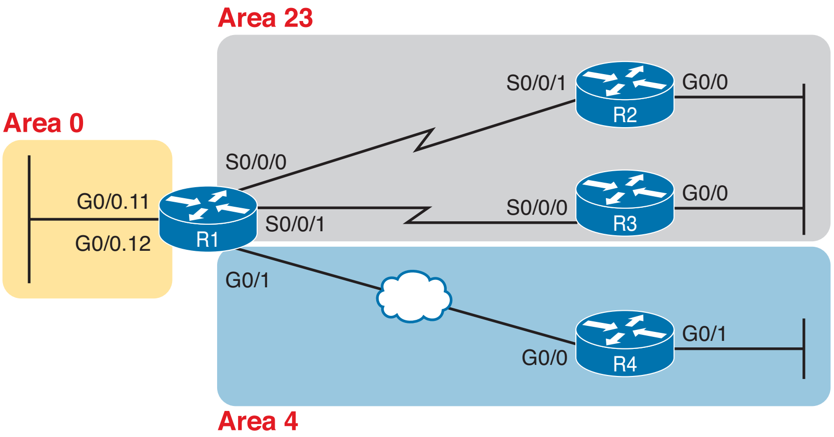

Figure B-16 *The Area Design Used in the Upcoming OSPF Example*

To convert from the old-style configuration in Examples 8-1, 8-2, and 8-3, simply do the following:

Step 1. Remove the **network** commands under the **router ospf** command.

Step 2. Add one **ip ospf** *process-id* **area** *area-id* command under each interface on which OSPF should operate, with the correct OSPF process (*process-id*) and the correct OSPF area number.

For example, Example 8-1 in Chapter 8 had a single **network** command that enabled OSPF on two interfaces, putting both in area 23. Example B-13 shows the replacement newer style of configuration.

Example B-13 *New-Style Configuration on Router R2*

```
interface GigabitEthernet0/0
  ip address 10.1.23.2 255.255.255.0
  ip ospf 1 area 23
!
interface serial 0/0/1
 ip address 10.1.12.2 255.255.255.0
 ip ospf 1 area 23
!
router ospf 1
 router-id 2.2.2.2
! Notice - no network commands here!
```

Verifying Newer OSPFv2 Configuration

OSPF operates the same way whether you use the new or old style configuration. The OSPF area design works the same, neighbor relationships form the same way, routers negotiate to become the DR and BDR the same way, and so on. However, you can see a few small differences in command output when using the newer OSPFv2 configuration if you look closely.

The **show ip protocols** command relists most of the routing protocol configuration, just in slightly different format, as shown in Example B-14. With the newer style configuration, the output lists the phrase "Interfaces Configured Explicitly," with the list of interfaces configured with the new **ip ospf** *process-id* **area** *area-id* commands, as highlighted in the example. With the old configuration, the output lists the contents of all the network commands, just leaving out the "network" word itself, as seen near the bottom of the example, showing R3's configuration.

Example B-14 *Differences in* **show ip protocols** *Output: Old- and New-Style OSPFv2 Configuration*

```
R2# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 22.2.2.2
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
  Routing on Interfaces Configured Explicitly (Area 23):
    Serial0/0/1
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    3.3.3.3              110      00:04:59
  Distance: (default is 110
```

```
! Showing only the part that differs on R3:
R3# show ip protocols
! … beginning lines omitted for brevity
  Routing for Networks:
    10.0.0.0 0.255.255.255 area 23
! … ending line omitted for brevity
```

The **show ip ospf interface** [*interface*] command lists details about OSPF settings for the interface(s) on which OSPF is enabled. The output also makes a subtle reference to whether that interface was enabled for OSPF with the old or new configuration style. As seen in Example B-15, R2's new-style configuration results in the highlighted text, "Attached via Interface Enable," whereas R3's old-style configuration lists "Attached via Network Statement."

Example B-15 *Differences in* **show ip protocols** *Output: Old- and New-Style OSPFv2 Configuration*

```
R2# show ip ospf interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 10.1.23.2/24, Area 23, Attached via Interface Enable
  Process ID 1, Router ID 22.2.2.2, Network Type BROADCAST, Cost: 1
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           1         no          no            Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 22.2.2.2, Interface address 10.1.23.2
  Backup Designated router (ID) 3.3.3.3, Interface address 10.1.23.3
```

```
! Showing only the part that differs on R3:
R3# show ip ospf interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 10.1.23.3/24, Area 23, Attached via Network Statement
! … ending line omitted for brevity
```

Finally, the sample output from R2 in Example B-15 shows a great sample with which to explain the Designated Router (DR) and Backup Designated Router (BDR) election criteria. When all the routers on a LAN come up at the same time, the router with the numerically highest RID wins. In this case, R2, with router ID 22.2.2.2, is the DR, and R3, with RID 3.3.3.3, is the BDR. (Note that if two or more routers have already elected a DR and BDR, when new routers join the subnet, they do not preempt the existing DR or BDR, even if they have better [higher] RID values.)

EIGRP Issues on Multipoint and Physical Interfaces

NOTE The best time to read this section is when reading Chapter 14, "Implementing Frame Relay," after the similar heading about OSPF issues over Frame Relay.

Chapter 14 discusses some problems with OSPF when operating over Frame Relay physical and point-to-multipoint interfaces. This topic discusses some different problems that EIGRP has on these same types of interfaces. The problems revolve around the fact that these types of interfaces connect the router to multiple other routers over the same subnet. As a result, the EIGRP Split Horizon feature has some effects on the operation of EIGRP.

To review, Split Horizon causes a router to advertise a subset of its known routes rather than all routes. Specifically, if router R1 learns a route to subnet X from a routing update received on interface Y, Split Horizon causes R1 to not advertise about subnet X in updates sent back out interface Y. By way of analogy: If you tell me that Fred just got a new blue bowling ball, I don't need to turn around and tell you the same thing. Split Horizon creates the same logic for routing protocols.

EIGRP Split Horizon works well with point-to-point subinterfaces because at most two routers exist in the same subnet. For example, in Figure B-17, the two routers use point-to-point subinterfaces. Mayberry advertises about subnet 199.1.11.0/24 to Mount Pilot. Mount Pilot does not need to then re-advertise a route for that same subnet back to Mayberry, so using EIGRP Split Horizon in this topology makes perfect sense.

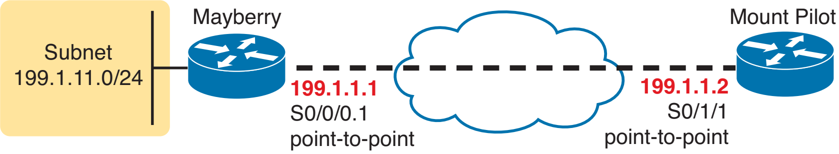

Figure B-17 *A Frame Relay Point-to-Point Topology: Split Horizon Works Perfectly*

However, with physical and point-to-multipoint interfaces, that one router can connect to more than one other router, resulting in more than two routers in that WAN subnet. As a result, in some cases, a router must learn routes from an EIGRP update received on that interface, and to then re-advertise that same route to another router off that same interface. Split Horizon would prevent that re-advertisement.

First, for perspective, consider the network in Figure B-18, which works well whether Split Horizon is enabled or disabled. In this case, three routers all use a physical serial interface for their Frame Relay configuration. The figure shows the PVCs in a full mesh, but the Frame Relay configuration sits on the one physical serial interface (S0/1/1 on each router). Because a PVC exists between each pair of routers, each router forms an EIGRP neighbor relationship with the other two routers. (Note that this design uses the same configuration shown in Chapter 14 for physical interface configuration [see Examples 14-1, 14-2, and 14-3].)

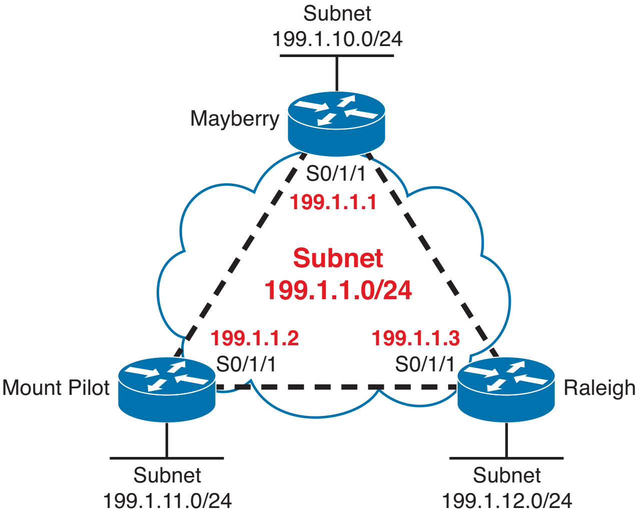

Figure B-18 *A Frame Relay Full Mesh: Split Horizon Works Perfectly*

Next, to see why Split Horizon can cause problems on physical interfaces, now imagine that the design in Figure B-18 does not use a full mesh, but instead uses a partial mesh, as shown in Figure B-19. One key to understanding what happens is that EIGRP forms neighbor relationships over Frame Relay only with routers connected directly by a PVC. In other words, in this example, Mount Pilot and Raleigh will not become EIGRP neighbors.

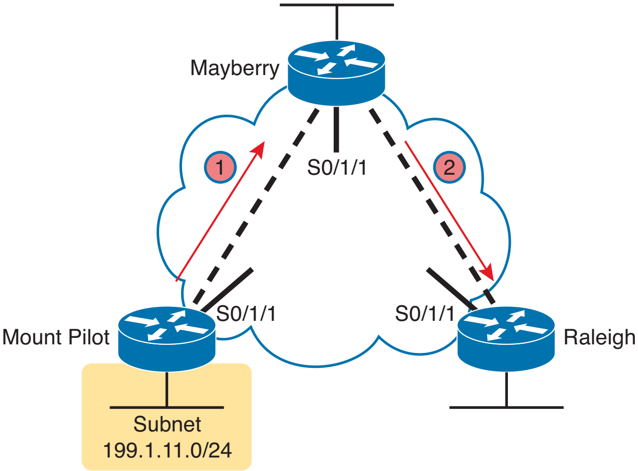

Figure B-19 *A Frame Relay Partial Mesh: Split Horizon Has Problems*

With the partial mesh of Figure B-19, some routers fail to learn all routes if using EIGRP Split Horizon. To allow all routers to learn all routes, Mayberry must learn routes from one router and advertise them to another. To do that, Mayberry must disable Split Horizon for the EIGRP protocol. Once disabled, the following will occur, using subnet 199.1.11.0/24 as an example:

1. At Step 1, Mount Pilot advertises a route for 199.1.11.0/24 to Mayberry.
2. At Step 2, Mayberry can advertise a route for 199.1.11.0/24 to Raleigh, out that same S0/1/1 interface, because Mayberry has disabled EIGRP Split Horizon on S0/1/1.

Step 2 in this example requires that Split Horizon be disabled. With this design, Mayberry must disable EIGRP Split Horizon, using the **no ip split-horizon eigrp** *asn* command. (The **ip split-horizon eigrp** *asn* command enables EIGRP Split Horizon again.)

As seen in the previous example, disabling Split Horizon on physical and point-to-multipoint subinterfaces makes sense if a partial mesh exists. And, even if the network has a full mesh design, PVCs can fail, creating a partial mesh. So, unless some other compelling reason exists to leave Split Horizon on, turn it off when using EIGRP physical and point-to-multipoint interfaces. Table B-8 summarizes the default settings for Split Horizon with EIGRP.

Table B-8 EIGRP Split Horizon Defaults for Frame Relay

(Sub)Interface Type	Default	Recommendation
Physical	On	Off
Point-to-multipoint	On	Off
Point-to-point	On	On